Документация Deckhouse Code

Дата генерации: 23.10.2026

Документация

О продукте ... **9**
Назначение и возможности ... 9
Редакции ... 11
Установка ... **18**
Пакет для ОС Linux ... 18
Требования ... 18
Быстрый старт ... 19
Настройка после установки ... 23
Диагностика ... 26
Omnibus Docker ... 28
Требования ... 28
Быстрый старт ... 30
Настройка после установки ... 34
Диагностика ... 37
Helm-чарт ... 39
Требования ... 39
Быстрый старт ... 43
Настройка ... 46
Расширенная настройка ... 57
Масштабирование ... 91
Диагностика ... 127
Модуль Deckhouse Kubernetes Platform ... 132
Администрирование ... **135**
Обзор ... 135
Миграция ... 136
С GitLab CE ... 136
Между типами установки ... 144
Лицензия ... 145
Пользователи и доступ ... 150
Аутентификация ... 152
Обзор ... 152
Настройка ... 167
OpenID Connect ... 171
SAML ... 178

LDAP ... 179
Настройки безопасности ... 196
Ключи доступа ... 203
Сервисные аккаунты ... 205
События аудита безопасности ... 206
Мониторинг и лимиты ... 226
Режим обслуживания ... 231
Расширенный поиск ... 236
Relay (KAS) ... 244
Резервное копирование и восстановление ... 248
Обновление ... 255
Раннеры ... 260
Ранний доступ ... 272
Использование ... 276
Обзор ... 276
Начало работы ... 276
Вход в систему ... 276
Работа с Git ... 278
Проекты и задачи ... 280
Группы и участники ... 280
Создание проекта ... 281
Импорт проекта ... 282
Управление задачами ... 284
Wiki группы ... 285
Аналитика задач ... 287
Аналитика вкладов ... 291
Аналитика потоков ценности ... 294
Дашборд потоков ценности ... 302
Репозитории ... 309
Управление репозиториями Git ... 309
Защищённые ветки ... 310
Правила отправки изменений ... 314
Pull-зеркалирование ... 316
CODEOWNERS ... 319
Групповая аналитика репозитория ... 328
Аналитика репозитория проекта ... 330
Аналитика участников ... 331

- Запросы на слияние ... 332
  - Ревью кода ... 332
  - Правила апрувов ... 334
  - Внешние проверки статуса ... 338
  - Цепочки слияния ... 347
  - Аналитика ревью кода ... 362
  - Аналитика запросов на слияние ... 363
  - Аналитика продуктивности ... 366
- CI/CD ... 372
  - Непрерывная сборка и поставка ПО ... 372
  - Защищённые окружения ... 373
  - Интеграция с внешним Vault ... 392
  - Доступ пользователей к Kubernetes ... 407
  - Аналитика CI/CD ... 411
  - DORA-метрики ... 415
- Безопасность ... 422
  - Настройки безопасности группы и проекта ... 422
  - Политики выполнения пайплайна ... 425
  - SAST-сканирование с помощью Opengrep ... 444
  - Обнаружение секретов с помощью Gitleaks ... 470
  - Сканирование с помощью Trivy ... 480
  - Список зависимостей и соответствие лицензий ... 527
  - Отчёт об уязвимостях ... 534
- Интеграции ... 541
  - Вебхуки ... 541
  - Интеграции со сканерами безопасности ... 546
  - Интеграция с DefectDojo ... 555
  - CodeScoring ... 561
  - Интеграция с Ягой ... 571
  - PT AI ... 575
- Расширенный поиск ... 590

Справка ... 601

- REST API ... 601
  - Обзор ... 601
  - Access requests ... 603
  - Access tokens ... 609
  - Admin opensearch ... 657
  - Agents ... 658

Ai agents ... 672
Alert management ... 674
Applications ... 679
Attestations ... 684
Audit events ... 686
Authn ... 687
Avatars ... 688
Award emoji ... 690
Badges ... 712
Batched background migrations ... 721
Batched background operations ... 725
Boards ... 730
Branches ... 747
Broadcast messages ... 756
CI catalog ... 762
CI jobs ... 763
CI lint ... 785
CI resource groups ... 790
CI runners ... 804
CI triggers ... 809
CI variables ... 823
Cluster agents ... 837
Clusters ... 845
Codeowners ... 874
Commit statuses ... 876
Commits ... 882
Container registry ... 910
Custom attributes ... 920
DORA metrics ... 928
Database dictionary ... 931
Dependency proxy ... 933
Deploy resources ... 933
Discussions ... 989
Draft notes ... 1024
Environments ... 1034
Error tracking ... 1062
Events ... 1067
External status checks ... 1077
Feature flags ... 1102

Features 1113
Files 1118
Freeze periods 1127
GLQL 1132
Geo 1135
GitLab Pages 1136
Group import and export 1144
Groups 1151
Hooks 1239
Imports 1271
Instance 1283
Integrations 1372
Internal operations 1656
Invitations 1656
Issues 1664
Jira Connect subscriptions 1702
Jira forge installation 1703
Jira forge subscriptions 1705
Jira forge user delegation 1707
Job artifacts 1708
Jobs 1733
Keys 1761
Labels 1782
MCP 1800
ML Model Registry 1802
MLOps 1807
Markdown 1836
Members 1837
Merge request approvals 1851
Merge requests 1856
Merge trains 1947
Metadata 1955
Metrics 1956
Migrations 1957
Milestones 1959
Mobile push subscriptions 1986
Namespaces 1988
Notes 1991
Notification settings 2016

Offline transfers ... 2023
Organizations ... 2034
Packages ... 2041
Packages: Cargo ... 2068
Packages: Composer ... 2073
Packages: Conan ... 2078
Packages: Debian ... 2155
Packages: Helm ... 2200
Packages: NPM ... 2203
Packages: NuGet ... 2217
Packages: PyPi ... 2242
Packages: RPM ... 2251
Packages: RubyGem ... 2253
Pipeline schedules ... 2258
Pipelines ... 2278
Plan limits ... 2326
Project import ... 2333
Project snapshots ... 2387
Project templates ... 2388
Project topics ... 2392
Projects ... 2396
Projects job token scope ... 2579
Protected branches ... 2590
Protected environments ... 2597
Protected tags ... 2612
Releases ... 2616
Remote mirrors ... 2639
Repositories ... 2646
Resource events ... 2674
Resource subscriptions ... 2687
Runners ... 2712
Search ... 2751
Secure files ... 2761
Service accounts ... 2765
Sidekiq ... 2793
Snippets ... 2813
Storage moves ... 2833
Submodules ... 2845
Suggestions ... 2849

Tags 2851
Templates 2859
Terraform 2866
To-dos 2881
Unleash 2888
Usage data 2891
Users 2895
VSCode 2947
Web commits 2953
Wikis 2954
Other 2965
История изменений 2985
Журнал изменений 2989

О продукте Deckhouse Code

Назначение и возможности

Deckhouse Code организует совместную работу команд разработки и эксплуатации: поддерживает хранение и версионирование кода, ревью кода, CI/CD-пайплайны и поставку в целевые среды.

Deckhouse Code закрывает следующие сценарии:

- хранение и версионирование Git-репозиториев, включая форки, защищённые ветки и pull-зеркалирование;
- рецензирование и слияние изменений через запросы на слияние с правилами апрувов и цепочками слияния;
- выполнение CI/CD-пайплайнов, описанных в `.gitlab-ci.yml`;
- управление проектами, группами и правами доступа;
- сканирование кода на уязвимости и отслеживание зависимостей — в Enterprise Edition.

Типы установки

Deckhouse Code поставляется в виде пакета для ОС Linux, образа Omnibus Docker, Helm-чарта или модуля Deckhouse Kubernetes Platform.

Тип установки	Что это	Когда применяется	Документация по установке
Пакет для ОС Linux	Deckhouse Code, установленный из пакета `.deb` или `.rpm` на одном хосте	Для хоста с поддерживаемым дистрибутивом Linux, без контейнера и кластера Kubernetes	Быстрый старт
Omnibus Docker	Deckhouse Code, запущенный как один контейнер, собранный из того же omnibus-пакета	Для хоста с Docker вместо прямой установки пакета	Быстрый старт
Helm-чарт	Deckhouse Code, развёрнутый в кластере Kubernetes с помощью Helm-чарта, без оператора Deckhouse Kubernetes Platform	Для кластера Kubernetes, управляемого независимо от Deckhouse Kubernetes Platform	Быстрый старт
Модуль Deckhouse Kubernetes Platform	Deckhouse Code, устанавливаемый и обслуживаемый оператором через ресурс CodeInstance	Для кластера Deckhouse Kubernetes Platform	Модуль Deckhouse Kubernetes Platform

Интеграции

Deckhouse Code поддерживает следующие интеграции:

- Jira — документация по интеграции с Jira.
- Jenkins — документация по интеграции с Jenkins.
- EvaProject — руководство по интеграции с EvaProject.

Дальнейшие шаги

Deckhouse Code доступен в редакциях Standard Edition и Enterprise Edition, различия описаны в разделе «Редакции».

Задачи администрирования инстанса описаны в разделе «Администрирование», начиная с его страницы «Обзор». Повседневная работа с проектами, репозиториями и запросами на слияние описана в разделе «Использование», начиная с его страницы «Обзор».

Редакции

Deckhouse Code поставляется в редакциях Standard Edition (SE) и Enterprise Edition (EE).

Доступность возможностей

Категория	Возможности	SE	EE
Типы установки	Пакет для ОС Linux, Omnibus Docker, Helm-чарт, модуль Deckhouse Kubernetes Platform	Да	Да
Проекты и группы	Группы и участники, уровни видимости проектов, импорт проектов, задачи с несколькими ответственными, метки, вехи, доски, wiki проекта и группы, отложенное удаление проектов и групп	Да	Да
Исходный код	Git-репозитории с форками и веб-редактором, защищённые ветки, правила отправки изменений, pull-зеркалирование, CODEOWNERS, доступ к Git по SSH и HTTPS	Да	Да
Ревью кода и запросы на слияние	Запросы на слияние с несколькими проверяющими и ответственными, правила апрувов, внешние проверки статуса, цепочки слияния	Да	Да
CI/CD	Пайплайны из `.gitlab-ci.yml`, защищённые окружения, секреты	Да	Да

Категория	Возможности	SE	EE
	из внешнего Vault или Deckhouse Stronghold, сервисные аккаунты для пайплайнов, доступ к Kubernetes от имени пользователя через агент для Kubernetes, образы раннера с плагином автомасштабирования для Deckhouse Virtualization Platform		
Сканирование и отчётность по безопасности	Политики сканирования безопасности, политики выполнения пайплайна, SAST-сканирование с помощью Opengrep, обнаружение секретов встроенным Gitleaks, сканирование зависимостей, лицензий, секретов, образов контейнеров и кластеров Kubernetes с помощью Trivy, интеграции сканеров CodeScoring и PT AI, отчёты сканирования зависимостей, контейнеров и DAST, список зависимостей из CycloneDX SBOM, соответствие	Нет	Да

Категория	Возможности	SE	EE
	лицензий, отчёт об уязвимостях, интеграция с DefectDojo		
Аналитические отчёты	Аналитика задач, аналитика CI/CD, групповая аналитика репозитория, аналитика ревью кода, аналитика запросов на слияние, аналитика продуктивности, аналитика вкладов, аналитика потоков ценности с пользовательскими потоками, дашборд потоков ценности, DORA-метрики	Нет	Да
Аналитика проекта	Языки программирования, покрытие тестами и статистика коммитов проекта, коммиты по авторам, встроенный поток ценности проекта	Да	Да
Настройки безопасности и аудит	Настройки безопасности группы и проекта, настройки безопасности инстанса, ключи доступа, события аудита	Да	Да
Идентификация и доступ	Вход с персональной двухфакторной аутентификацией,	Да	Да

Категория	Возможности	SE	EE
	провайдеры OmniAuth, синхронизация LDAP, ограничения входа, ограничения создания групп и проектов, ограничения протокола, обязательная двухфакторная аутентификация для инстанса		
Поиск	Расширенный поиск, индексирование поиска через OpenSearch	Да	Да
Интеграции	Вебхуки для событий группы, проекта и подгруппы; Relay (KAS) для подключения агентов для Kubernetes	Да	Да
Интеграции	Интеграция с системой управления задачами Яга	Да	Да
Эксплуатация	Лимиты ресурсов и настройки производительности, мониторинг, режим обслуживания, миграция с GitLab CE и между типами установки, резервное копирование и восстановление, обновление, диагностика,	Да	Да

Категория	Возможности	SE	EE
	лицензионный ключ инстанса		
REST API	Операции для проектов, групп, пользователей, запросов на слияние, CI/CD и остальных ресурсов; токены доступа и OAuth 2.0	Да	Да

Страницы, доступные только в Enterprise Edition, отмечены в боковом меню документации значком «EE».

Установка

Пакет для ОС Linux

Требования

Пакет для ОС Linux разворачивает Deckhouse Code на одном сервере: веб-интерфейс, Git, база данных и фоновые задачи работают на этой же машине.

Операционная система

Поддерживаемые дистрибутивы, архитектура — `x86_64`:

- РЕД ОС 8.x;
- AlmaLinux 9.x;
- Ubuntu 22.04 LTS;
- Ubuntu 24.04 LTS.

Системные зависимости устанавливаются из репозиториев ОС, поэтому доступ к ним нужен на время установки. Доступ в интернет не требуется.

Ресурсы

Значения рассчитаны на типовой инстанс:

Ресурс	Значение
Оперативная память	8 ГБ; минимум — 6 ГБ и раздел подкачки (swap)
Диск	от 20 ГБ под каталог данных `/var/opt/gitlab`

Для инстансов, обслуживающих сотни активных пользователей и больше, запросите у вендора рекомендации по подбору ресурсов.

Установочный пакет

Установочный пакет получите у вендора, имя файла зависит от ОС:

ОС	Файл пакета
РЕД ОС 8.x	`deckhouse-code-<VERSION>.el8.x86_64.rpm`
AlmaLinux 9.x	`deckhouse-code-<VERSION>.el9.x86_64.rpm`
Ubuntu 22.04 LTS	`deckhouse-code_<VERSION>_amd64.deb`
Ubuntu 24.04 LTS	`deckhouse-code_<VERSION>_amd64.deb`

Пакеты для Ubuntu 22.04 и Ubuntu 24.04 собираются отдельно и называются одинаково: запросите пакет для версии ОС на сервере.

`<VERSION>` — версия полученного пакета, например `1.0.0-0`.

Сетевые порты

Инстанс принимает соединения на портах, которые открывают на файрволе сервера:

Порт	Назначение
`22`	Git по SSH, обслуживается системным sshd
`80`	Перенаправление на HTTPS
`443`	Веб-интерфейс и Git по HTTPS

Быстрый старт

Сначала инстанс запускается по HTTP, затем на нём включается TLS. Перед началом убедитесь, что сервер отвечает требованиям.

В командах и файлах конфигурации замените `<HOSTNAME>` на FQDN инстанса, а `<VERSION>` — на версию полученного пакета, например `1.0.0-0`.

> CLI-утилиты (`gitlab-ctl`, `gitlab-backup`), файл конфигурации (`/etc/gitlab/gitlab.rb`) и служебные каталоги (`/opt/gitlab`, `/var/opt/gitlab`) сохраняют префикс `gitlab-`: Deckhouse Code работает на движке GitLab, и системные имена оставлены без изменений.

Подготовка

Имя инстанса `<HOSTNAME>` используется в адресе веб-интерфейса, в адресах репозиториев для Git, в параметре `external_url` и в полях CN и SAN сертификата — везде одно и то же имя.

1. Проверьте, что имя разрешается в IP-адрес машины:

```
getent hosts <HOSTNAME>
```

2. Если DNS-записи A или AAAA нет, добавьте имя в файл `/etc/hosts` на сервере:

   ```
   echo "<IP_ADDRESS> <HOSTNAME>" | sudo tee -a /etc/hosts
   ```

   `<IP_ADDRESS>` — IP-адрес машины инстанса. Ту же строку добавьте на каждой машине, с которой будут открывать веб-интерфейс и работать с Git, включая вашу рабочую станцию.

Установка пакета

Установка сама запускает первичную настройку (`gitlab-ctl reconfigure`), она занимает несколько минут. Адрес инстанса берётся из переменной `EXTERNAL_URL` при первой установке; TLS включается на следующем шаге.

РЕД ОС 8

Установите системные зависимости:

```
sudo dnf install -y libxcrypt-compat policycoreutils-python-utils
```

Пакет `libxcrypt-compat` даёт библиотеку `libcrypt.so.1` , без которой приложение не запускается: в минимальной поставке РЕД ОС этой библиотеки нет. Пакет `policycoreutils-python-utils` даёт утилиту `semanage` , которая выставляет контексты SELinux при первичной настройке.

Установите пакет Deckhouse Code:

```
sudo EXTERNAL_URL="http://<HOSTNAME>" \
  rpm -i ./deckhouse-code-<VERSION>.el8.x86_64.rpm
```

Режим SELinux `enforcing` поддерживается: модуль политики устанавливается при первичной настройке, ручные действия не нужны.

AlmaLinux 9

Отдельно устанавливать системные зависимости не нужно: они объявлены в пакете, и `dnf` возьмёт их из репозиториев ОС.

Установите пакет Deckhouse Code:

```
sudo EXTERNAL_URL="http://<HOSTNAME>" \
  dnf install -y ./deckhouse-code-<VERSION>.el9.x86_64.rpm
```

Режим SELinux `enforcing` поддерживается: модуль политики устанавливается при первичной настройке, ручные действия не нужны.

Ubuntu 22.04, 24.04

Отдельно устанавливать системные зависимости не нужно: они объявлены в пакете, и `apt` возьмёт их из репозиториев ОС.

Установите пакет Deckhouse Code:

```
sudo EXTERNAL_URL="http://<HOSTNAME>" \
  apt install -y ./deckhouse-code_<VERSION>_amd64.deb
```

Префикс `./` в пути к файлу обязателен: без него `apt` ищет пакет с таким именем в репозиториях.

Что получилось

На `http://<HOSTNAME>` работает инстанс: веб-интерфейс, Git поверх HTTP, база данных и фоновые задачи. Проверьте состояние — команды выполняются на самой машине инстанса:

```
# Все сервисы в состоянии run.
sudo gitlab-ctl status
# Ожидается код 200.
curl -s -o /dev/null -w "%{http_code}\n" http://localhost/-/health
# Первый пароль администратора.
sudo cat /etc/gitlab/initial_root_password
```

Откройте `http://<HOSTNAME>` в браузере и войдите под пользователем `root` с паролем из файла `/etc/gitlab/initial_root_password`.

⚠ Сохраните пароль: файл `initial_root_password` хранится 24 часа, и любой следующий запуск `gitlab-ctl reconfigure` удалит его, если с установки прошло больше суток. Само значение пароля при этом не меняется.

TLS

По HTTP пароли и содержимое репозиториев передаются в открытом виде, поэтому сразу после первого входа включите TLS.

Подготовьте каталог для сертификатов:

```
sudo mkdir -p /etc/gitlab/ssl && sudo chmod 755 /etc/gitlab/ssl
```

Разместите в нём сертификат и ключ под именами `<HOSTNAME>.crt` и `<HOSTNAME>.key`.

Свой сертификат

Выпустите в вашем центре сертификации сертификат на имя `<HOSTNAME>` и скопируйте файлы на сервер:

```
sudo install -m 600 <KEY_FILE> /etc/gitlab/ssl/<HOSTNAME>.key
# Сертификат вместе с цепочкой.
sudo install -m 644 <CERT_FILE> /etc/gitlab/ssl/<HOSTNAME>.crt
```

`<KEY_FILE>` и `<CERT_FILE>` — пути к полученным файлам ключа и сертификата.

Самоподписанный сертификат

На тестовом стенде выпустите сертификат на месте. Имя инстанса в поле SAN обязательно, иначе клиенты отклонят сертификат:

```
sudo openssl req -x509 -nodes -newkey rsa:2048 -days 825 \
  -keyout /etc/gitlab/ssl/<HOSTNAME>.key \
  -out /etc/gitlab/ssl/<HOSTNAME>.crt \
  -subj "/CN=<HOSTNAME>/O=Deckhouse Code/C=RU" \
  -addext "subjectAltName=DNS:<HOSTNAME>"
sudo chmod 600 /etc/gitlab/ssl/<HOSTNAME>.key
```

Включите HTTPS в файле `/etc/gitlab/gitlab.rb`:

```
external_url 'https://<HOSTNAME>'
letsencrypt['enable'] = false
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] = "/etc/gitlab/ssl/<HOSTNAME>.crt"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/<HOSTNAME>.key"
```

Без строки `letsencrypt['enable'] = false` настройка с адресом `https://` запускает получение сертификата через Let's Encrypt по HTTP-01 и в закрытом контуре завершается ошибкой.

Примените конфигурацию:

```
sudo gitlab-ctl reconfigure
```

Проверка

Проверьте состояние сервисов, ответ по HTTPS и перенаправление с HTTP:

```
# Все сервисы в состоянии run.
sudo gitlab-ctl status
# Параметр --resolve направляет запрос на loopback: адрес /-/health отвечает
# только с самой машины, проверка имени в сертификате при этом сохраняется.
# Ожидается health=200 tls=0.
curl --cacert /etc/gitlab/ssl/<HOSTNAME>.crt \
     --resolve '<HOSTNAME>:443:127.0.0.1' \
     -o /dev/null -w "health=%{http_code} tls=%{ssl_verify_result}\n" \
     https://<HOSTNAME>/-/health
# Перенаправление с HTTP на HTTPS, ожидается https://<HOSTNAME>/.
curl -sI http://<HOSTNAME>/ | grep -i location
```

Самоподписанный сертификат браузер и Git не примут, пока центр сертификации не добавлен в доверенные. Для Git укажите путь к файлу сертификата на машине клиента:

```
git config --global http.https://<HOSTNAME>.sslCAInfo <CERT_FILE>
```

Смените пароль пользователя `root` в веб-интерфейсе и удалите файл первоначального пароля:

```
sudo rm -f /etc/gitlab/initial_root_password
```

Настройка после установки

Эти настройки выполняются до передачи инстанса пользователям. Изменения в файле `/etc/gitlab/gitlab.rb` применяются командой `sudo gitlab-ctl reconfigure`.

Язык интерфейса

Язык задаётся для всего инстанса администратором и отдельно каждым пользователем для себя.

Русский язык для всего инстанса действует на страницу входа и на всех пользователей, которые не выбирали язык вручную. Войдите администратором, в правом верхнем углу выберите «Admin», в левой панели перейдите в «Settings» → «Preferences», прокрутите страницу до раздела «Localization», в поле «Default language» выберите «Russian - русский» и нажмите «Save changes».

Пользователь меняет язык для себя: аватар в правом верхнем углу → «Preferences» → «Localization» → «Language» → «Russian - русский» → «Save changes». Интерфейс переключится после обновления страницы. Личный выбор имеет приоритет над языком инстанса.

Почта (SMTP)

Без почты не работают сброс пароля, приглашения и уведомления. Задайте параметры подключения к серверу почты в файле `/etc/gitlab/gitlab.rb`:

```
gitlab_rails['smtp_enable'] = true
gitlab_rails['smtp_address'] = "<SMTP_HOST>"
gitlab_rails['smtp_port'] = 587
gitlab_rails['smtp_user_name'] = "<SMTP_USER>"
gitlab_rails['smtp_password'] = "<SMTP_PASSWORD>"
gitlab_rails['smtp_domain'] = "<DOMAIN>"
gitlab_rails['smtp_authentication'] = "login"
gitlab_rails['smtp_enable_starttls_auto'] = true
gitlab_rails['gitlab_email_from'] = "deckhouse-code@<DOMAIN>"
```

Примените конфигурацию и отправьте тестовое письмо на свой адрес:

```
sudo gitlab-ctl reconfigure
sudo gitlab-rails runner \
  "Notify.test_email('<ADMIN_EMAIL>', 'Тест', 'Работает').deliver_now"
```

Сетевой доступ (файрвол)

Инстансу нужны порты `22` (Git по SSH через системный sshd), `80` (перенаправление на HTTPS) и `443` (веб-интерфейс и Git по HTTPS). Откройте их средствами вашей ОС.

firewalld (РЕД ОС, AlmaLinux)

```
sudo systemctl enable --now firewalld
sudo firewall-cmd --permanent --add-service={ssh,http,https}
sudo firewall-cmd --reload
```

ufw (Ubuntu)

```
sudo ufw allow OpenSSH
sudo ufw allow 80,443/tcp
sudo ufw enable
```

Ресурсы

При памяти 6–8 ГБ закрепите число рабочих процессов, чтобы автоматическая настройка не заняла всю память. Задайте значения в файле `/etc/gitlab/gitlab.rb`:

```
puma['worker_processes'] = 2
sidekiq['concurrency'] = 10
```

Примените конфигурацию:

```
sudo gitlab-ctl reconfigure
```

Проверьте, что на сервере настроен раздел подкачки (swap): он покрывает пики потребления памяти при миграциях во время обновления и при тяжёлых операциях с репозиториями.

Порт SSH

Если служба SSH инстанса слушает порт, отличный от 22, укажите его в `/etc/gitlab/gitlab.rb`, чтобы URL для клонирования в веб-интерфейсе содержали этот порт:

```
gitlab_rails['gitlab_shell_ssh_port'] = 2222
```

Примените конфигурацию:

```
sudo gitlab-ctl reconfigure
```

Расширенный поиск (OpenSearch)

Расширенный поиск работает на внешнем кластере OpenSearch. Задайте подключение в `/etc/gitlab/gitlab.rb` и направьте задачи индексации в отдельную очередь Sidekiq:

```
gitlab_rails['fe_search'] = {
  'advanced_search_enabled' => true,
  'opensearch' => {
    'url' => 'https://<OPENSEARCH_HOST>:9200',
    'username' => '<OPENSEARCH_USER>',
    'password' => '<OPENSEARCH_PASSWORD>'
  }
}
sidekiq['routing_rules'] = [
  ['feature_category=fe_global_search', 'global-search-indexing'],
  ['*', 'default']
]
```

`<OPENSEARCH_HOST>` — адрес кластера OpenSearch, `<OPENSEARCH_USER>` и `<OPENSEARCH_PASSWORD>` — учётные данные аккаунта, которому принадлежат индексы. Примените конфигурацию:

```
sudo gitlab-ctl reconfigure
```

Затем переиндексируйте инстанс и включите расширенный поиск в настройках, как описано на странице «Расширенный поиск».

Диагностика

Ошибки ниже встречаются при установке пакета для ОС Linux и первичной настройке инстанса. Если нужного случая здесь нет, соберите вывод команд из раздела «Общая диагностика» и обратитесь к вендору.

Приложение не запускается: нет библиотеки libcrypt.so.1

Где проявляется. РЕД ОС, вывод установки пакета и первичной настройки:

```
libcrypt.so.1: cannot open shared object file
```

Причина. В минимальной поставке РЕД ОС нет библиотеки `libcrypt.so.1`.

Решение. Установите пакет с библиотекой и повторите настройку:

```
sudo dnf install -y libxcrypt-compat
sudo gitlab-ctl reconfigure
```

Настройка завершается ошибкой SELinux

Где проявляется. РЕД ОС, выполнение `gitlab-ctl reconfigure`, ресурс `bash[Set proper security context on ssh files for selinux]`:

```
ValueError: Type gitlab_shell_t is invalid
semanage: command not found
```

Причина. На сервере нет утилит SELinux из пакета `policycoreutils-python-utils` либо модуль политики не загрузился при первичной настройке.

Решение. Установите утилиты, повторите настройку и проверьте, что модуль политики загружен:

```
sudo dnf install -y policycoreutils-python-utils
sudo gitlab-ctl reconfigure
sudo semodule -l | grep -i gitlab
```

Имя инстанса недоступно: curl возвращает 000

Причина. Имя `<HOSTNAME>` не разрешается в IP-адрес на сервере или на машине, с которой открывают веб-интерфейс, либо порт `80` или `443` недоступен: закрыт файрволом или не запущен nginx.

Решение.

1. Проверьте имя командой `getent hosts <HOSTNAME>`. Если имени нет, создайте DNS-запись A или AAAA на IP-адрес машины, а для стенда без DNS добавьте строку в `/etc/hosts`. Имя должно совпадать с `external_url` и с полями CN и SAN сертификата.
2. Если имя разрешается в ожидаемый IP-адрес, проверьте на сервере состояние nginx и занятые порты:

   ```
   sudo gitlab-ctl status nginx
   sudo ss -tlnp | grep -E ':(80|443)\b'
   ```

3. Проверьте вход по HTTP с рабочей станции:

   ```
   curl -sS -o /dev/null -w "HTTP %{http_code}\n" \
     http://<HOSTNAME>/
   ```

   Любой HTTP-код (после включения TLS обычно 301 или 302) означает, что nginx доступен. Код 000 означает, что соединение не установлено. Если nginx запущен, а порт закрыт снаружи, откройте его по разделу «Сетевой доступ (файрвол)».

Веб-интерфейс отвечает 502 после настройки или перезапуска

Причина. Приложение ещё запускается: первые одну-две минуты после `reconfigure` или `restart` код 502 ожидаем.

Решение. Подождите и проверьте состояние сервисов командой `sudo gitlab-ctl status`. Если 502 держится дольше нескольких минут, читайте лог командой `sudo gitlab-ctl tail puma`: частая причина — нехватка памяти, порядок настройки описан в разделе «Ресурсы».

Настройка завершается ошибкой Let’s Encrypt

Где проявляется. Выполнение `gitlab-ctl reconfigure` после того, как в `external_url` задан адрес `https://`, а файлов сертификата на диске нет.

Причина. Без сертификата на диске включается получение сертификата через Let’s Encrypt, что в закрытом контуре невозможно.

Решение. Разместите сертификат и задайте `letsencrypt['enable'] = false`, как описано в разделе «TLS», затем выполните `sudo gitlab-ctl reconfigure`.

Чтение ошибки в выводе reconfigure

Где проявляется. Вывод команды `gitlab-ctl reconfigure`, завершившейся с ошибкой.

Решение. В конце вывода найдите блок с именем ресурса и причиной:

```
Error executing action ... on resource '...'
```

Полный лог запуска сохраняется в каталоге `/var/log/gitlab/reconfigure/`.

Общая диагностика

Команды применимы к любой из перечисленных ошибок:

```
# Состояние всех сервисов.
sudo gitlab-ctl status
# Живой лог сервиса, например puma или nginx.
sudo gitlab-ctl tail <SERVICE>
# Самопроверка инстанса.
sudo gitlab-rake gitlab:check SANITIZE=true
# Каталоги логов по сервисам.
ls /var/log/gitlab/
```

Omnibus Docker

Требования

Deckhouse Code работает в одном контейнере с тем же omnibus-пакетом, что и при установке пакетом для ОС Linux. Хост предоставляет Docker Engine, ресурсы одного инстанса, каталоги под тома и сетевые порты.

Хост

Контейнер запускается на хосте с ОС Linux, архитектурой x86_64 и Docker Engine.

Требования к памяти и диску совпадают с установкой пакетом для ОС Linux и перечислены в разделе «Требования»: 8 ГБ RAM (минимум 6 ГБ и swap) и не менее 20 ГБ свободного места под данные инстанса. Образ занимает на хосте ещё 2,5–3 ГБ.

Тома

Образ объявляет тома для конфигурации, данных и логов. Подключите их все: в них хранится состояние, которое переживает контейнер, — при обновлении контейнер пересоздаётся, а тома остаются.

Путь в контейнере	Содержимое
`/etc/gitlab`	Файл конфигурации `gitlab.rb`, ключи шифрования `gitlab-secrets.json`, TLS-сертификаты, ключи хоста SSH, файл с первоначальным паролем `root`.
`/var/opt/gitlab`	Репозитории, база данных, загруженные файлы, артефакты CI, реестр пакетов, архивы резервных копий в `/var/opt/gitlab/backups`.
`/var/log/gitlab`	Логи всех сервисов и лог каждого запуска настройки в `/var/log/gitlab/reconfigure`.

В примерах на этих страницах тома — это каталоги хоста: `/srv/code/config`, `/srv/code/data` и `/srv/code/logs`.

Сетевые порты

Опубликованные порты обслуживают трафик Git по SSH, HTTP и HTTPS.

Порт	Назначение
`22`	Git по SSH. В контейнере работает собственный `sshd`, который принимает пользователя `git` и аутентификацию по открытому ключу.
`80`	Веб-интерфейс и Git по HTTP, после настройки TLS — перенаправление на HTTPS.
`443`	Веб-интерфейс и Git по HTTPS.

Порты публикуются при создании контейнера. Порт хоста может отличаться от порта контейнера: тогда в адресе инстанса указывают порт хоста, который открывают пользователи, а порт SSH, отличный от 22, указывают в конфигурации, как описано на странице «Настройка после установки».

Разделяемая память

Создавайте контейнер с параметром `--shm-size 256m`. По умолчанию Docker выделяет под `/dev/shm` 64 МБ.

Быстрый старт

Для работающего инстанса достаточно одной команды `docker run` и первого запуска, который настраивает инстанс. Сначала инстанс отвечает по HTTP, включение TLS — последний шаг этой страницы.

Перед началом

Проверьте хост по разделу «Требования» и определитесь с именем `<HOSTNAME>`: по нему пользователи открывают веб-интерфейс и обращаются к репозиториям Git. Это же имя указывается в адресе инстанса и в сертификате.

Проверьте, что имя разрешается в IP-адрес хоста:

```
getent hosts <HOSTNAME>
```

Если DNS-записи нет, добавьте имя в `/etc/hosts` на хосте и на каждой машине, с которой открывают веб-интерфейс и работают с Git.

Получение образа

Образ публикуется в вариантах по базовому дистрибутиву внутри образа: название варианта подставляется вместо `<FLAVOR>` в ссылке на образ. На требования к хосту вариант не влияет — они приведены на странице «Требования».

Вариант	Базовый дистрибутив
`ubuntu_22.04`	Ubuntu 22.04
`ubuntu_24.04`	Ubuntu 24.04
`redos_8`	РЕД ОС 8
`almalinux_9`	AlmaLinux 9

Каждый тег называет версию, отдельного тега для последней версии нет, поэтому и при первом запуске, и при каждом обновлении версия указывается явно:

```
docker pull <REGISTRY>/<FLAVOR>:<VERSION>
```

Запуск контейнера

Создайте контейнер с томами, опубликованными портами и адресом инстанса:

```
docker run -d --name code \
  --shm-size 256m \
  -p 80:80 -p 443:443 -p 22:22 \
  -e GITLAB_OMNIBUS_CONFIG="external_url 'http://<HOSTNAME>'" \
  -v /srv/code/config:/etc/gitlab \
  -v /srv/code/logs:/var/log/gitlab \
  -v /srv/code/data:/var/opt/gitlab \
  <REGISTRY>/<FLAVOR>:<VERSION>
```

Имя `code` используется во всех командах ниже. Левый порт в каждом `-p` — это порт хоста, он может отличаться от порта контейнера; порт 22 на хосте обычно занят системным `sshd`, а настройка, которая указывает другой порт в URL для клонирования, описана в разделе «Настройка после установки».

Порты, тома и переменные окружения задаются при создании контейнера. Чтобы изменить любое из них, удалите контейнер и создайте новый с теми же томами.

Адрес инстанса

Без настройки адресом становится `http://<CONTAINER_HOSTNAME>`, где `<CONTAINER_HOSTNAME>` — имя хоста контейнера. Docker задаёт его равным идентификатору контейнера, если не указан

параметр `--hostname`. Этот идентификатор попадёт во все ссылки и URL для клонирования, поэтому задайте адрес при первом запуске.

Адрес — это настройка `external_url` в `gitlab.rb`, и образ берёт её из переменной окружения и из файла конфигурации:

- `GITLAB_OMNIBUS_CONFIG` содержит строки конфигурации и вычисляется при каждом запуске;
- `/etc/gitlab/gitlab.rb` на томе конфигурации читается после переменной, поэтому настройка из файла переопределяет ту же настройку из переменной.

Переменная `EXTERNAL_URL`, которую читает установщик пакета, в контейнере не действует.

Первый запуск

Стартовый скрипт готовит инстанс до того, как сервисы начнут принимать запросы:

- копирует `gitlab.rb` из шаблона, если на томе конфигурации файла нет;
- генерирует ключи хоста SSH в `/etc/gitlab`, если их нет;
- выполняет `gitlab-ctl reconfigure` — шаг, который при первом запуске занимает несколько минут;
- приводит базу данных к текущей версии PostgreSQL;
- выводит логи сервисов в вывод контейнера.

Следите за запуском по выводу контейнера:

```
docker logs -f code
```

В образе настроена проверка состояния, которая выполняется каждые 60 секунд. До первой успешной проверки состояние контейнера — `starting`, после пяти неуспешных проверок подряд — `unhealthy`; первый запуск доходит до этого состояния, пока настройка ещё выполняется:

```
docker inspect --format '{{.State.Health.Status}}' code
```

Запуск завершён, когда состояние контейнера — `healthy`, а все сервисы находятся в состоянии `run`:

```
docker exec code gitlab-ctl status
```

Первый вход

Первый пароль пользователя `root` записывается на том конфигурации. Прочитайте его из контейнера:

```
docker exec code cat /etc/gitlab/initial_root_password
```

Откройте `http://<HOSTNAME>` и войдите под пользователем `root` с этим паролем.

> ⚠ Файл с паролем удаляется при первой настройке, которая выполняется позже чем через 24 часа после записи файла, и шаг с TLS ниже — это такая настройка. Смените пароль `root` в веб-интерфейсе и удалите файл: `docker exec code rm -f /etc/gitlab/initial_root_password`.

По HTTP данные для входа и содержимое репозиториев передаются в открытом виде. Инстанс, с которым работают пользователи, работает по TLS.

TLS

Разместите сертификат и ключ на томе конфигурации, чтобы контейнер читал их из `/etc/gitlab/ssl`:

```
sudo mkdir -p /srv/code/config/ssl && sudo chmod 755 /srv/code/config/ssl
sudo install -m 644 <CERT_FILE> /srv/code/config/ssl/<HOSTNAME>.crt
sudo install -m 600 <KEY_FILE> /srv/code/config/ssl/<HOSTNAME>.key
```

`<CERT_FILE>` — файл сертификата в формате PEM, `<KEY_FILE>` — файл его закрытого ключа.

Выпуск сертификата, требование к `subjectAltName` и причина, по которой отключается Let's Encrypt, описаны в разделе «Быстрый старт».

Добавьте настройки в файл `/srv/code/config/gitlab.rb`, который контейнер читает как `/etc/gitlab/gitlab.rb`:

```
external_url 'https://<HOSTNAME>'
letsencrypt['enable'] = false
nginx['redirect_http_to_https'] = true
nginx['ssl_certificate'] = "/etc/gitlab/ssl/<HOSTNAME>.crt"
nginx['ssl_certificate_key'] = "/etc/gitlab/ssl/<HOSTNAME>.key"
```

Пути в настройках — это пути внутри контейнера. Примените изменение перезапуском контейнера: настройка выполняется при каждом запуске.

```
docker restart code
```

Проверка

Проверьте сервисы, состояние контейнера и перенаправление с HTTP:

```
docker exec code gitlab-ctl status
docker exec code gitlab-healthcheck --fail --max-time 10
docker inspect --format '{{.State.Health.Status}}' code
curl -sI http://<HOSTNAME>/ | grep -i location
```

Откройте `https://<HOSTNAME>` в браузере, войдите и создайте тестовый проект, чтобы проверить доступ к Git по HTTPS и по SSH.

Настройка после установки

Настройки инстанса в контейнере — это настройки установки пакетом для ОС Linux, к которым добавляется слой контейнера: где лежит файл конфигурации, какие переменные читает стартовый скрипт и какие порты публикует хост.

Применение изменений конфигурации

Файл конфигурации — `/etc/gitlab/gitlab.rb` на томе конфигурации, то есть `/srv/code/config/gitlab.rb` на хосте. Правьте его на хосте или в контейнере:

```
docker exec -it code editor /etc/gitlab/gitlab.rb
```

Примените изменение, не останавливая контейнер:

```
docker exec code gitlab-ctl reconfigure
```

Перезапуск даёт тот же результат: настройка выполняется при каждом запуске контейнера.

```
docker restart code
```

Язык интерфейса, почта, число рабочих процессов и срок хранения резервных копий — это те же настройки, что и при установке на хост; что делает каждая из них, описано в разделе «Настройка после установки».

Переменные запуска

Стартовый скрипт читает переменные при запуске контейнера, поэтому для изменения любой из них контейнер создают заново с теми же томами.

Переменная	Назначение
`GITLAB_OMNIBUS_CONFIG`	Содержит строки конфигурации, которые вычисляются до чтения `/etc/gitlab/gitlab.rb`. Настройка из файла переопределяет ту же настройку из переменной.
`GITLAB_SKIP_RECONFIGURE`	Со значением `true` на уже настроенном инстансе запуск пропускает `gitlab-ctl reconfigure`, и изменения конфигурации остаются неприменёнными до следующей настройки.
`GITLAB_SKIP_PG_UPGRADE`	Со значением `true` запуск не выполняет `gitlab-ctl pg-upgrade`, и база данных остаётся на текущей версии PostgreSQL.
`GITLAB_DISABLE_OPENSSH`	Со значением `true` запуск не готовит сервис `sshd`, и Git по SSH недоступен.
`GITLAB_SKIP_TAIL_LOGS`	Со значением `true` логи сервисов не выводятся в вывод контейнера и остаются на томе логов.
`GITLAB_PRE_RECONFIGURE_SCRIPT`	Если переменная задана, её значение выполняется как команда оболочки до запуска сервисов.
`GITLAB_POST_RECONFIGURE_SCRIPT`	Если переменная задана, её значение выполняется как команда оболочки после шага с PostgreSQL.
`GITLAB_ALLOW_SHA1_RSA`	Со значением `true` конфигурация `sshd` принимает алгоритм ключа хоста и тип открытого ключа `ssh-rsa`.
`OPENSSL_FORCE_FIPS_MODE`	Значение переменной записывается в окружение сервиса `sshd`.

Ключи хоста SSH

Ключи хоста RSA, ECDSA и Ed25519 генерируются при первом запуске в `/etc/gitlab` и связываются символическими ссылками с `/etc/ssh`, откуда их читает инстанс. Ключи лежат на томе конфигурации, поэтому контейнер, пересозданный при обновлении, сохраняет их, и клиенты

Git видят прежний ключ хоста. Новый том конфигурации означает новые ключи, и каждый клиент при следующем подключении сообщит о смене ключа хоста.

Git по SSH работает с ключами, которые пользователи добавляют в веб-интерфейсе: `sshd` в контейнере слушает порт 22, принимает только пользователя `git`, аутентификация по паролю отключена.

Ресурсы и рабочие процессы

Контейнеру доступна память хоста, если при создании она не ограничена параметром `--memory`, и ему нужен параметр `--shm-size 256m`, как описано в разделе «Требования».

На хосте с 6–8 ГБ RAM закрепите число рабочих процессов, чтобы его не подбирала автонастройка. Настройки задаются в `/etc/gitlab/gitlab.rb` или в `GITLAB_OMNIBUS_CONFIG`, их значения описаны в разделе «Настройка после установки»:

```
puma['worker_processes'] = 2
sidekiq['concurrency'] = 10
```

Порты и файрвол хоста

Порты 80, 443 и 22 публикуются при создании контейнера. Порт хоста может отличаться от порта контейнера, например `-p 8080:80 -p 8443:443 -p 2222:22`.

В адресе инстанса указывается порт хоста, который открывают пользователи; порт SSH, который веб-интерфейс показывает в URL для клонирования, задаёт параметр `gitlab_rails['gitlab_shell_ssh_port']`, описанный в разделе «Порт SSH».

Откройте опубликованные порты на хосте для сетей пользователей; команды для `firewalld` и `ufw` приведены в разделе «Настройка после установки». Docker публикует порты собственными правилами `iptables`, поэтому после каждого изменения правил файрвола проверяйте порты с рабочей станции:

```
curl -sI http://<HOSTNAME>/ | head -n 1
ssh -T -p 22 git@<HOSTNAME>
```

Расширенный поиск (OpenSearch)

Подключение к OpenSearch — параметр `gitlab_rails['fe_search']` и правило маршрутизации Sidekiq, описанные в разделе «Расширенный поиск (OpenSearch)». Добавьте их в `/etc/gitlab/gitlab.rb` в томе с конфигурацией или в `GITLAB_OMNIBUS_CONFIG`, затем перезапустите контейнер или выполните `docker exec code gitlab-ctl reconfigure`.

Диагностика

Большинство сбоев инстанса в контейнере проявляется при запуске, а причина видна в выводе контейнера или в логах на томе логов. В каждом разделе ниже приведены симптом, причина и порядок действий.

Веб-интерфейс отвечает 502 после запуска

Причина. При каждом запуске выполняется `gitlab-ctl reconfigure`, и приложение принимает запросы только после него. Первый запуск занимает несколько минут, последующие — одну-две минуты.

Решение. следите за запуском по выводу контейнера и дождитесь окончания настройки:

```
docker logs -f code
docker exec code gitlab-ctl status
```

Если 502 держится дольше, приложение не запускается: прочитайте его лог и проверьте свободную память на хосте.

```
docker exec code gitlab-ctl tail puma
```

Состояние контейнера остаётся unhealthy

Причина. Проверка состояния выполняется каждые 60 секунд и после пяти неуспешных проверок подряд переводит контейнер в состояние `unhealthy`; первый запуск доходит до этого состояния, пока настройка ещё выполняется. Если состояние `unhealthy` сохраняется и после неё, сервисы не запустились.

Решение. выполните ту же проверку вручную и посмотрите состояние сервисов и лог настройки:

```
docker exec code gitlab-healthcheck --fail --max-time 10
docker exec code gitlab-ctl status
docker exec code ls /var/log/gitlab/reconfigure
```

Сервисы не запускаются после копирования или восстановления томов

Причина. Владелец и права файлов на томах не соответствуют учётным записям внутри образа, которые созданы с фиксированными идентификаторами.

Решение. восстановите права командой из образа и перезапустите контейнер:

```
docker exec -it code update-permissions
docker restart code
```

Обновление базы данных завершилось с ошибкой, контейнер остановился

Где проявляется. в выводе контейнера есть строка `Upgrading the existing database failed and was reverted.`

Причина. При запуске выполняется `gitlab-ctl pg-upgrade`, обновление завершилось с ошибкой, и база данных возвращена к предыдущей версии.

Решение. создайте контейнер заново, добавив `-e GITLAB_SKIP_PG_UPGRADE=true` к команде из раздела «Быстрый старт». Инстанс запустится на текущей версии PostgreSQL, а обновление базы данных выполняется по разделу «Обновление».

Инстанс не отвечает по HTTPS

Причина. Образ объявляет порт 443, но порт доступен только тогда, когда контейнер его публикует, а порты задаются при создании контейнера.

Решение. удалите контейнер и создайте его заново командой из раздела «Быстрый старт» с параметром `-p 443:443` и теми же томами; данные остаются на томах:

```
docker stop code
docker rm code
```

В ссылках и URL для клонирования указан идентификатор контейнера

Причина. Адрес инстанса не задан, и образ строит его из имени хоста контейнера, которое Docker задаёт равным идентификатору контейнера.

Решение. задайте `external_url` в `/etc/gitlab/gitlab.rb` и перезапустите контейнер либо создайте контейнер заново с адресом в `GITLAB_OMNIBUS_CONFIG`, как описано в разделе «Быстрый старт».

Контейнер останавливается на проверке версии

Где проявляется. в выводе контейнера указаны найденная версия данных и версия, на которую нужно обновиться сначала.

Причина. При запуске версия на томе данных сравнивается с версией образа, и запуск прекращается, если перейти к ней за один шаг нельзя.

Решение. запустите сначала тег промежуточной версии, затем целевой, как описано в разделе «Обновление».

Изменение конфигурации не действует

Где проявляется. в выводе контейнера есть строка `Skipped reconfigure because GITLAB_SKIP_RECONFIGURE is set.`

Причина. Переменной `GITLAB_SKIP_RECONFIGURE` задано значение `true`, и запуск оставляет конфигурацию прежней. Тот же симптом без этой строки означает, что настройка задана в `GITLAB_OMNIBUS_CONFIG` и переопределена файлом `/etc/gitlab/gitlab.rb`.

Решение. выполните настройку вручную либо создайте контейнер без этой переменной:

```
docker exec code gitlab-ctl reconfigure
```

Git по SSH отклоняет подключение

Причина. Порт 22 контейнера не опубликован, либо переменной `GITLAB_DISABLE_OPENSSH` задано значение `true` и сервис `sshd` не подготовлен, либо клиент подключается к порту хоста, который не указан в URL для клонирования.

Решение. проверьте опубликованные порты и переменные контейнера, затем задайте порт SSH, как описано в разделе «Настройка после установки»:

```
docker port code
docker inspect --format '{{.Config.Env}}' code
```

Если клиент сообщает о смене ключа хоста после перезапуска, контейнер работает с новым томом конфигурации: ключи хоста сгенерированы заново.

Общая диагностика

Эти команды показывают состояние работающего контейнера:

```
docker logs -f code
docker exec code gitlab-ctl status
docker exec code gitlab-ctl tail <SERVICE>
docker exec code gitlab-rake gitlab:check SANITIZE=true
docker inspect --format '{{.State.Health.Status}}' code
```

Логи всех сервисов лежат на томе логов, на хосте это `/srv/code/logs`, а лог каждого запуска настройки — в `/var/log/gitlab/reconfigure`.

Helm-чарт

Требования

Helm-чарт разворачивает Deckhouse Code напрямую в кластере Kubernetes, без модуля Deckhouse Kubernetes Platform. Чарт разворачивает компоненты приложения и, если оно включено, хранилище образов контейнеров; кластер и сервисы ниже существуют до установки релиза.

Helm

Чарт публикуется как OCI-артефакт и устанавливается клиентом командной строки `helm`. Чарт выводит предупреждение, если клиент старше Helm 4.0, и просит проверить конфигурацию перед обновлением, если клиент старше Helm 3.17, потому что эта версия изменила слияние значений.

Ingress-контроллер

В кластере работает Ingress-контроллер. Чарт создаёт объекты Ingress для веб-интерфейса, хранилища образов контейнеров и, если включён, Pages; они адресованы классу Ingress из `global.ingress.class`, по умолчанию `nginx`. Балансировщик, который разворачивает чарт, передаёт HTTP и HTTPS в Service `ingress-nginx-controller.ingress-nginx.svc`, если `global.codeHelm.loadbalancer.ingressService` не указывает другой.

Service типа LoadBalancer

Чарт разворачивает балансировщик HAProxy с Service типа LoadBalancer, который принимает HTTP, HTTPS и Git по SSH на одном внешнем адресе. Кластер выдаёт внешние адреса таким Service через облачного провайдера или реализацию вроде MetalLB. Если в кластере её нет, выключите балансировщик значением `global.codeHelm.loadbalancer.enabled: false` и откройте SSH через `gitlab.gitlab-shell.service.type`; см. раздел Балансировщик.

cert-manager

В кластере работает cert-manager с API `cert-manager.io/v1`. Чарт использует его в двух местах:

- для сертификатов Ingress, которые подписывает ClusterIssuer из `global.codeHelm.ingress.tls.clusterIssuer` (по умолчанию `letsencrypt`); ClusterIssuer существует до установки, либо `global.codeHelm.ingress.tls.mode` установлен в `manual`, а сертификаты переданы как Secret;
- для внутренних сертификатов между компонентами, которые подписывает самоподписанный Issuer, создаваемый чартом.

StorageClass для Git-репозиториев

Каждый под Gitaly монтирует PersistentVolumeClaim для Git-репозиториев из StorageClass по умолчанию или из класса, указанного в `gitlab.gitaly.persistence.storageClass`. Чарт рекомендует `50Gi` в `gitlab.gitaly.persistence.size` и принимает `1Gi` как минимальное значение. Том растёт вместе с репозиториями, поэтому задайте нужный им размер.

PostgreSQL

Внешний сервер PostgreSQL версии 17 или новее, доступный из кластера, с заранее созданными базами данных и ролями из таблицы ниже. Чарт не разворачивает PostgreSQL.

База данных	Значения	Кто использует
База данных приложения	`global.psql.host`, `global.psql.database`, `global.psql.username`	Веб-сервис, Sidekiq, задача миграций и toolbox.
База данных Praefect	`global.praefect.psql.host`, `global.praefect.psql.dbName`, `global.praefect.psql.user`	Praefect, который направляет Git-трафик в Gitaly. Praefect включён по умолчанию, поэтому эта база данных обязательна.
База метаданных хранилища образов	`registry.database.host`, `registry.database.dbname`, `registry.database.user`	Хранилище образов контейнеров, только при `registry.database.enabled: true`.

Redis

Внешний сервер Redis, доступный из кластера и указанный в `global.redis.host` и `global.redis.port`, хранит очереди фоновых задач, кеш и сессии. Чарт не разворачивает Redis.

Объектное хранилище

S3-совместимое объектное хранилище с отдельным бакетом для каждого хранилища данных. Объектное хранилище включено по умолчанию, и заменить его локальными томами нельзя; чарт не разворачивает хранилище. Бакеты и их значения:

Бакет	Значение	Имя по умолчанию
Объекты Git LFS	`global.appConfig.lfs.bucket`	`git-lfs`
Артефакты CI/CD	`global.appConfig.artifacts.bucket`	`gitlab-artifacts`
Загруженные файлы	`global.appConfig.uploads.bucket`	`gitlab-uploads`
Пакеты	`global.appConfig.packages.bucket`	`gitlab-packages`
Диффы запросов на слияние	`global.appConfig.externalDiffs.bucket`	`gitlab-mr-diffs`
Состояние Terraform	`global.appConfig.terraformState.bucket`	`gitlab-terraform-state`
Защищённые файлы CI/CD	`global.appConfig.ciSecureFiles.bucket`	`gitlab-ci-secure-files`
Содержимое планов агента	`global.appConfig.agentPlanContent.bucket`	`gitlab-agent-plan-content`
Архивы каталога CI/CD	`global.appConfig.ciCatalogBundles.bucket`	`gitlab-ci-catalog-bundles`
Прокси зависимостей	`global.appConfig.dependencyProxy.bucket`	`gitlab-dependency-proxy`
Архивы резервных копий	`global.appConfig.backups.bucket`	`gitlab-backups`
Временные файлы резервного копирования	`global.appConfig.backups.tmpBucket`	`tmp`
Хранилище образов контейнеров	`global.codeHelm.externalSecrets.registry.storage.s3.bucket` , если хранилище образов включено	—
Содержимое Pages	`global.pages.objectStore.bucket` , если Pages включён	`gitlab-pages`

Точка доступа хранилища и ключи доступа записываются в `global.codeHelm.externalSecrets.s3`, `global.codeHelm.externalSecrets.backup` и `global.codeHelm.externalSecrets.registry.storage.s3`; см. раздел Secret из значений.

Доступ к хранилищу образов

Кластер загружает образы компонентов из хранилища образов Deckhouse Code. Учётные данные записываются в `global.codeHelm.externalSecrets.registryDockerConfig` как `.dockerconfigjson` в кодировке Base64; чарт превращает их в Secret для загрузки образов каждого пода.

DNS-имена

Веб-интерфейс отвечает по имени `global.hosts.gitlab.name`, включённое хранилище образов по `global.hosts.registry.name`, включённый Pages по `global.hosts.pages.name`. Незаданное имя выводится из `global.hosts.domain` как `gitlab.<домен>`, `registry.<домен>` и `pages.<домен>`. Имена разрешаются во внешний адрес Service балансировщика или в адрес Ingress-контроллера, если балансировщик выключен; адрес известен после установки, см. раздел Быстрый старт.

Быстрый старт

Перед началом выполните проверки из раздела Требования. Шаги ниже устанавливают Deckhouse Code как релиз `code` в неймспейсе `code` с веб-интерфейсом по адресу `code.example.com`. Хранилище образов контейнеров по умолчанию выключено; как его включить, описано в разделе Хранилище образов контейнеров.

Вход в реестр чартов

Чарт публикуется как OCI-артефакт. `<CHART_REGISTRY>` — адрес реестра, `<CHART_VERSION>` — версия чарта; оба значения передаются вместе с продуктом и учётными данными реестра. Войдите в реестр:

```
helm registry login <CHART_REGISTRY>
```

Подготовка файла значений

Создайте файл `values.yaml` с именами хостов, классом Ingress, подключениями к PostgreSQL, Redis и объектному хранилищу, именами бакетов и StorageClass для Git-репозиториев. Чарт создаёт Secret с паролями и ключами из блока `global.codeHelm.externalSecrets`, поэтому вручную Secret не создаются:

```
global:
  hosts:
    domain: example.com
    gitlab:
      name: code.example.com
  ingress:
    class: nginx
  psql:
    host: <POSTGRESQL_HOST>
    port: 5432
    database: <POSTGRESQL_DATABASE>
    username: <POSTGRESQL_USERNAME>
  praefect:
    psql:
      host: <POSTGRESQL_HOST>
      user: <PRAEFECT_USERNAME>
      dbName: <PRAEFECT_DATABASE>
  redis:
    host: <REDIS_HOST>
    port: 6379
  appConfig:
    lfs:
      bucket: <LFS_BUCKET>
    artifacts:
      bucket: <ARTIFACTS_BUCKET>
    uploads:
      bucket: <UPLOADS_BUCKET>
    packages:
      bucket: <PACKAGES_BUCKET>
    externalDiffs:
      bucket: <MR_DIFFS_BUCKET>
    terraformState:
      bucket: <TERRAFORM_STATE_BUCKET>
    dependencyProxy:
      bucket: <DEPENDENCY_PROXY_BUCKET>
    backups:
      bucket: <BACKUPS_BUCKET>
      tmpBucket: <BACKUP_TMP_BUCKET>
  codeHelm:
    ingress:
      tls:
        clusterIssuer: <CLUSTER_ISSUER_NAME>
    externalSecrets:
      enabled: true
      psqlPassword: <POSTGRESQL_PASSWORD>
      praefectPassword: <PRAEFECT_PASSWORD>
      redisPassword: <REDIS_PASSWORD>
      s3:
```

```
        provider: AWS
        region: <S3_REGION>
        endpoint: <S3_ENDPOINT_URL>
        aws_access_key_id: <S3_ACCESS_KEY_ID>
        aws_secret_access_key: <S3_SECRET_ACCESS_KEY>
      backup:
        aws_access_key_id: <S3_ACCESS_KEY_ID>
        aws_secret_access_key: <S3_SECRET_ACCESS_KEY>
      registryDockerConfig: <BASE64_DOCKERCONFIGJSON>
    backup:
      region: <S3_REGION>
      endpoint: <S3_ENDPOINT_URL>
gitlab:
  gitaly:
    persistence:
      size: 50Gi
      storageClass: <STORAGE_CLASS_NAME>
```

Замените каждый плейсхолдер в угловых скобках. `<S3_ENDPOINT_URL>` — URL хранилища со схемой, например `https://storage.example.com`. `<BASE64_DOCKERCONFIGJSON>` — файл учётных данных хранилища образов, закодированный командой `base64 -w0`. Бакеты существуют в хранилище до установки. Раздел Настройка объясняет каждое показанное здесь значение, раздел Расширенная настройка перечисляет остальные.

Установка релиза

```
helm upgrade --install code oci://<CHART_REGISTRY>/code \
  --version <CHART_VERSION> \
  -n code --create-namespace \
  -f values.yaml
```

Команда создаёт неймспейс `code`, если его нет, и устанавливает релиз `code`. Чарт проверяет значения до создания объектов; если проверка не пройдена, команда выводит причину и завершается без установки. Сообщения перечислены в разделе Диагностика.

Настройка DNS на балансировщик

Релиз публикует HTTP, HTTPS и Git по SSH через один Service типа LoadBalancer. Получите его внешний адрес:

```
d8 k -n code get service code-code-loadbalancer
```

Создайте DNS-запись для `code.example.com`, указывающую на адрес из колонки `EXTERNAL-IP`. Git-клиенты подключаются по SSH к порту 22 того же адреса.

Ожидание готовности релиза

Задача миграций выполняется один раз и подготавливает схему базы данных; поды веб-сервиса и Sidekiq ждут её завершения. Убедитесь, что задача завершена, а все поды в состоянии `Running`:

```
d8 k -n code get jobs,pods
```

Сертификаты Ingress выпускает cert-manager после установки. Убедитесь, что у каждого Certificate в колонке `READY` значение `True`:

```
d8 k -n code get certificate
```

Первый вход

Чарт генерирует начальный пароль учётной записи `root` в Secret `code-initial-root-password`. Прочитайте его:

```
d8 k -n code get secret code-initial-root-password -o jsonpath='{.data.password}' |
base64 -d
```

Откройте `https://code.example.com` и войдите как `root` с этим паролем. После первого входа смените пароль в настройках учётной записи.

Проверка установки

Создайте проект в веб-интерфейсе, затем клонируйте его по HTTPS и по SSH, чтобы убедиться, что оба пути через балансировщик работают:

```
git clone https://code.example.com/root/<PROJECT>.git
git clone git@code.example.com:root/<PROJECT>.git
```

Для клонирования по SSH добавьте открытый ключ своей учётной записи в настройках пользователя.

Настройка

Значения ниже настраивают Helm-релиз Deckhouse Code сверх минимальной установки из раздела Быстрый старт. Имена значений совпадают с именами в `values.yaml` чарта; раздел Расширенная настройка перечисляет значения, которые чарт добавляет к upstream-чарту GitLab, и изменённые значения upstream-чарта по умолчанию. Изменение применяется той же командой, что и установка:

```
helm upgrade --install code oci://<CHART_REGISTRY>/code \
  --version <CHART_VERSION> -n code -f values.yaml
```

Имена хостов и Ingress

`global.hosts.domain` — базовый домен. Веб-интерфейс и, если они включены, хранилище образов и Pages отвечают по именам `gitlab.<домен>`, `registry.<домен>` и `pages.<домен>`, если `global.hosts.gitlab.name`, `global.hosts.registry.name` и `global.hosts.pages.name` не задают другие. `global.ingress.class` выбирает Ingress-контроллер; `global.shell.port` — порт SSH, который Git-клиенты видят в URL для клонирования.

```
global:
  hosts:
    domain: example.com
    gitlab:
      name: code.example.com
    registry:
      name: registry.example.com
  ingress:
    class: nginx
  shell:
    port: 22
```

TLS

По умолчанию сертификаты Ingress выпускает cert-manager из ClusterIssuer, указанного в `global.codeHelm.ingress.tls.clusterIssuer`, в Secret `code-tls`, `code-registry-tls` и `code-pages-tls`, по одному для каждого включённого адреса. Чтобы использовать собственный сертификат, переключитесь в ручной режим и укажите Secret типа `kubernetes.io/tls`, который покрывает все имена хостов:

```
global:
  codeHelm:
    ingress:
      tls:
        mode: manual
  ingress:
    tls:
      secretName: <TLS_SECRET_NAME>
```

Создайте Secret в неймспейсе релиза до обновления; чарт его не создаёт и не продлевает. Отдельный Secret для каждого включённого компонента указывается в `gitlab.webservice.ingress.tls.secretName`, `registry.ingress.tls.secretName`, `gitlab.gitlab-pages.ingress.tls.secretName` и `gitlab.kas.ingress.tls.secretName`. Оба режима описаны в разделе TLS для Ingress.

Балансировщик

Релиз публикует HTTP, HTTPS и Git по SSH через Service `code-code-loadbalancer` типа LoadBalancer, который обслуживает Deployment HAProxy. Внешний адрес Service назначает кластер; `global.codeHelm.loadbalancer.annotations` передаёт аннотации, которые читает реализация балансировщика, например фиксированный адрес для MetalLB:

```
global:
  codeHelm:
    loadbalancer:
      enabled: true
      annotations:
        metallb.universe.tf/loadBalancerIPs: <EXTERNAL_IP>
      replicas: 2
      resources:
        requests:
          cpu: 100m
          memory: 128Mi
        limits:
          cpu: 500m
          memory: 256Mi
```

Если в кластере работает несколько реализаций балансировщика, `global.codeHelm.loadbalancer.loadBalancerClass` выбирает одну из них. `global.codeHelm.loadbalancer.hpa` масштабирует поды HAProxy по использованию CPU и памяти вместо `replicas`, а `global.codeHelm.loadbalancer.pdb.enabled` сохраняет один под доступным, пока узлы освобождаются от подов. HTTP и HTTPS направляются в Service Ingress-контроллера `ingress-nginx-controller.ingress-nginx.svc`; для другого контроллера задайте `global.codeHelm.loadbalancer.ingressService`. Все значения перечислены в разделе Балансировщик.

Сохранение IP-адреса клиента

По умолчанию Ingress-контроллер и Git-оболочка видят IP-адрес пода HAProxy: журналы доступа, ограничение частоты запросов и блокировка IP-адресов в GitLab работают с одним адресом для всех клиентов.

Service `code-code-loadbalancer` создаётся с `externalTrafficPolicy: Local`: кластер передаёт поду HAProxy IP-адрес клиента без подмены, и трафик приходит только на узлы с готовыми подами HAProxy. При одной реплике весь внешний трафик обслуживает один узел.

Значение `global.codeHelm.loadbalancer.proxyProtocol: true` включает PROXY protocol v2 в соединениях с бэкендами. HAProxy отправляет перед началом трафика заголовок с IP-адресом и портом клиента, и бэкенды восстанавливают адрес клиента из этого заголовка.

Включайте `proxyProtocol` в таком порядке:

1. Включите во внешнем Ingress-контроллере `use-proxy-protocol: "true"`. Значение `proxy-real-ip-cidr` дополнительно ограничивает источники, адрес клиента из заголовка которых контроллер доверяет, например CIDR диапазоном подов кластера.
2. Обновите релиз со значением `proxyProtocol: true`:

```
global:
  codeHelm:
    loadbalancer:
      enabled: true
      proxyProtocol: true
```

> Ingress-контроллер должен принимать заголовок PROXY protocol до обновления релиза. Пока HAProxy отправляет заголовок, который контроллер не принимает, каждый web-запрос завершается ошибкой 400 или ошибкой TLS-соединения.

Сторона Git-оболочки включена значениями по умолчанию: `gitlab.gitlab-shell.sshDaemon: gitlab-sshd` и `gitlab.gitlab-shell.config.proxyProtocol: true`. Демон gitlab-sshd принимает и соединения без заголовка PROXY protocol, поэтому Git по SSH работает и при выключенном `proxyProtocol`. Установка завершается ошибкой, если `proxyProtocol` включён при `gitlab.gitlab-shell.sshDaemon: openssh` или `gitlab.gitlab-shell.config.proxyProtocol: false`: OpenSSH не декодирует заголовок, и Git по SSH перестал бы работать.

Балансировщик за прокси с PROXY protocol

Перед балансировщиком может стоять прокси, который принимает клиентские соединения и отправляет PROXY protocol: облачный NLB с включённым `proxy_protocol_v2`, HAProxy или Envoy перед кластером. Для такой цепочки включите `global.codeHelm.loadbalancer.frontendProxyProtocol: true` вместе с `proxyProtocol`:

```
global:
  codeHelm:
    loadbalancer:
      enabled: true
      frontendProxyProtocol: true
      proxyProtocol: true
```

HAProxy восстанавливает IP-адрес клиента из входящего заголовка и передаёт его бэкендам. Без `proxyProtocol` IP-адрес клиента остаётся только в журналах HAProxy, а бэкенды получают IP-адрес пода. Если прокси впереди отправляет заголовок, а `frontendProxyProtocol` выключен, HAProxy передаёт заголовок в поток данных, и соединения прекращаются.

> Включайте `frontendProxyProtocol` только при прокси с PROXY protocol впереди. С этим значением HAProxy принимает на портах 80, 443 и 22 только соединения с заголовком; адрес от реализации без PROXY protocol, например MetalLB или Cilium LB IPAM, прекращает весь трафик — web и Git по SSH.

Сервисная сеть Istio

`global.codeHelm.istio.enabled` включает поды релиза в сервисную сеть Istio, которая шифрует трафик между компонентами. Выключите внутренний TLS чарта и TLS каждого компонента в тех же значениях; пока один из них включён, установка завершается ошибкой:

```
global:
  codeHelm:
    istio:
      enabled: true
      strict: false
    tls:
      internal:
        enabled: false
  workhorse:
    tls:
      enabled: false
  gitaly:
    tls:
      enabled: false
  praefect:
    tls:
      enabled: false
gitlab:
  webservice:
    tls:
      enabled: false
```

Если хранилище образов включено, `registry.tls.enabled` тоже остаётся `false`. Чарт помечает каждый под для внедрения sidecar-контейнера и создаёт объект PeerAuthentication для подов релиза; если в кластере нет API `security.istio.io/v1beta1`, объект не создаётся. Задачи миграций, резервного копирования и переиндексации базы данных запускают прокси как init-контейнер с `restartPolicy: Always`, для этого нужен Kubernetes 1.29 или новее.

`global.codeHelm.istio.strict: true` переключает режим mTLS с `PERMISSIVE` на `STRICT`. Порты, к которым подключаются клиенты вне сети, остаются в режиме `PERMISSIVE`: Workhorse, который принимает запросы Ingress-контроллера, SSH и, если включён kube-rbac-proxy, порты метрик. Чарт добавляет в свои объекты Ingress аннотации `nginx.ingress.kubernetes.io`, которые направляют запросы Ingress-контроллера в сеть. Значения перечислены в разделе Istio.

PostgreSQL

`global.psql.host`, `global.psql.port`, `global.psql.database` и `global.psql.username` указывают на базу данных приложения. Пароль берётся из `global.codeHelm.externalSecrets.psqlPassword` или из существующего Secret, указанного в `global.psql.password.secret`, по ключу из `global.psql.password.key`. `global.psql.sslMode` задаёт режим SSL libpq, по умолчанию `prefer`.

```
global:
  psql:
    host: <POSTGRESQL_HOST>
    port: 5432
    database: <POSTGRESQL_DATABASE>
    username: <POSTGRESQL_USERNAME>
    sslMode: require
    password:
      secret: <POSTGRESQL_PASSWORD_SECRET>
      key: password
```

Praefect хранит метаданные маршрутизации в собственной базе данных, указанной в `global.praefect.psql.host`, `global.praefect.psql.port`, `global.praefect.psql.dbName` и `global.praefect.psql.user`. Её пароль берётся из `global.codeHelm.externalSecrets.praefectPassword` или из Secret в `global.praefect.dbSecret.secret` по ключу из `global.praefect.dbSecret.key`.

```
global:
  praefect:
    psql:
      host: <POSTGRESQL_HOST>
      port: 5432
      dbName: <PRAEFECT_DATABASE>
      user: <PRAEFECT_USERNAME>
      sslMode: require
    dbSecret:
      secret: <PRAEFECT_PASSWORD_SECRET>
      key: password
```

Redis

`global.redis.host` и `global.redis.port` указывают на сервер Redis. Пароль берётся из `global.codeHelm.externalSecrets.redisPassword` или из Secret в `global.redis.auth.secret` по ключу из `global.redis.auth.key`. Для сервера без пароля установите `global.redis.auth.enabled` в `false`.

```
global:
  redis:
    host: <REDIS_HOST>
    port: 6379
    auth:
      enabled: true
      secret: <REDIS_PASSWORD_SECRET>
      key: password
```

Для Redis за Sentinel `global.redis.host` называет набор master, а `global.redis.sentinels` перечисляет серверы Sentinel, каждый с `host` и `port`. Если серверам Sentinel нужен пароль, установите `global.redis.sentinelAuth.enabled` в `true` и передайте его через `global.codeHelm.externalSecrets.redisSentinelPassword` или через Secret, указанный в `global.redis.sentinelAuth.secret`, по ключу из `global.redis.sentinelAuth.key`.

Объектное хранилище

Приложение хранит загруженные файлы, артефакты, объекты LFS, пакеты, диффы запросов на слияние, состояние Terraform, кеш прокси зависимостей и архивы резервных копий в S3-совместимом хранилище. Подключение берётся из `global.codeHelm.externalSecrets.s3` или из существующего Secret, указанного в `global.appConfig.object_store.connection.secret`, ключ `connection` которого содержит подключение в формате библиотеки хранилища:

```
provider: AWS
region: <S3_REGION>
endpoint: <S3_ENDPOINT_URL>
aws_access_key_id: <S3_ACCESS_KEY_ID>
aws_secret_access_key: <S3_SECRET_ACCESS_KEY>
path_style: true
```

```
global:
  appConfig:
    object_store:
      connection:
        secret: <OBJECT_STORAGE_SECRET>
        key: connection
    lfs:
      bucket: <LFS_BUCKET>
    artifacts:
      bucket: <ARTIFACTS_BUCKET>
    uploads:
      bucket: <UPLOADS_BUCKET>
    packages:
      bucket: <PACKAGES_BUCKET>
    externalDiffs:
      bucket: <MR_DIFFS_BUCKET>
    terraformState:
      bucket: <TERRAFORM_STATE_BUCKET>
    dependencyProxy:
      bucket: <DEPENDENCY_PROXY_BUCKET>
    backups:
      bucket: <BACKUPS_BUCKET>
      tmpBucket: <BACKUP_TMP_BUCKET>
```

`path_style: true` нужен хранилищам, которые адресуют бакеты в пути URL вместо имени хоста, например MinIO. У каждого хранилища данных свой бакет в `global.appConfig.<хранилище>.bucket`; раздел Требования перечисляет их с именами по умолчанию.

Хранилище Git

Каждый под Gitaly хранит репозитории на PersistentVolumeClaim размером из `gitlab.gitaly.persistence.size` в StorageClass из `gitlab.gitaly.persistence.storageClass`. `global.praefect.virtualStorages[0].gitalyReplicas` задаёт число подов Gitaly за Praefect; каждый под хранит полную копию каждого репозитория.

```
global:
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 3
        maxUnavailable: 1
gitlab:
  gitaly:
    persistence:
      size: 200Gi
      storageClass: <STORAGE_CLASS_NAME>
```

Размер существующего PersistentVolumeClaim меняется через расширение тома в StorageClass; значение действует для PersistentVolumeClaim, созданных после изменения.

Хранилище образов контейнеров

Хранилище образов по умолчанию выключено. `registry.enabled` разворачивает его по имени `global.hosts.registry.name`, а `global.registry.enabled` включает хранилище образов в приложении: приложение выдаёт токены доступа к нему и показывает образы каждого проекта. Установите оба значения в `true`. Хранилище образов хранит слои образов в бакете из `global.codeHelm.externalSecrets.registry.storage.s3` или читает конфигурацию хранилища из существующего Secret, указанного в `registry.storage.secret`; без одного из них установка завершается ошибкой.

```
global:
  hosts:
    registry:
      name: registry.example.com
  registry:
    enabled: true
  codeHelm:
    externalSecrets:
      registry:
        storage:
          s3:
            bucket: <REGISTRY_BUCKET>
            region: <S3_REGION>
            regionendpoint: <S3_ENDPOINT_URL>
            accesskey: <S3_ACCESS_KEY_ID>
            secretkey: <S3_SECRET_ACCESS_KEY>
registry:
  enabled: true
```

Создайте DNS-запись для имени хранилища образов, указывающую на тот же адрес, что и веб-интерфейс. Входите в хранилище образов с именем пользователя и персональным токеном доступа с областями `read_registry` и `write_registry`:

```
docker login registry.example.com
```

Чтобы запустить хранилище образов с базой метаданных, укажите в `registry.database` базу данных PostgreSQL и передайте её пароль через `global.codeHelm.externalSecrets.registry.storage.database.password`:

```
registry:
  enabled: true
  database:
    enabled: true
    host: <POSTGRESQL_HOST>
    port: 5432
    dbname: <REGISTRY_DATABASE>
    user: <REGISTRY_USERNAME>
    sslmode: require
    migrations:
      enabled: true
global:
  codeHelm:
    externalSecrets:
      registry:
        storage:
          database:
            password: <REGISTRY_PASSWORD>
```

`registry.database.migrations.enabled` запускает миграции схемы хранилища образов задачей при каждом обновлении.

Pages

`global.pages.enabled` разворачивает Pages, который раздаёт статические сайты из бакета `global.pages.objectStore.bucket` по имени `global.hosts.pages.name` и его поддоменам. `global.pages.accessControl` ограничивает сайт участниками его проекта.

```
global:
  pages:
    enabled: true
    accessControl: false
    objectStore:
      bucket: <PAGES_BUCKET>
  hosts:
    pages:
      name: pages.example.com
```

Имя хоста Pages и маска `*.pages.example.com` разрешаются в тот же адрес, что и веб-интерфейс, а сертификат `code-pages-tls` выпускается для `pages.example.com`. Сайтам на поддоменах нужен сертификат с маской, который передаётся через `gitlab.gitlab-pages.ingress.tls.secretName` в ручном режиме TLS.

SMTP

`global.smtp.enabled` включает отправку писем. `global.smtp.address` и `global.smtp.port` указывают на сервер; `global.smtp.user_name` и Secret из `global.smtp.password.secret` хранят учётные данные; `global.smtp.authentication` выбирает `plain`, `login` или `cram_md5`; `global.smtp.starttls_auto` переводит соединение на STARTTLS. `global.email.from` и `global.email.display_name` задают отправителя.

```
global:
  smtp:
    enabled: true
    address: <SMTP_HOST>
    port: 587
    user_name: <SMTP_USERNAME>
    password:
      secret: <SMTP_PASSWORD_SECRET>
      key: password
    authentication: plain
    starttls_auto: true
    domain: <SMTP_HELO_DOMAIN>
  email:
    from: <SENDER_EMAIL>
    display_name: Deckhouse Code
    reply_to: <REPLY_TO_EMAIL>
```

Чарт не создаёт Secret с паролем SMTP. Создайте его до обновления:

```
d8 k -n code create secret generic <SMTP_PASSWORD_SECRET> --from-literal=password='<SM
TP_PASSWORD>'
```

Расширенная настройка

Чарт Deckhouse Code построен на Helm-чарте GitLab. Значения, которых нет на этой странице, сохраняют имена и смысл upstream-чарта. В таблицах ниже перечислены значения, которые чарт добавляет в `global.codeHelm`, дополнения в `global.psql` и `global.appConfig` и изменённые значения upstream-чарта по умолчанию. В колонке «По умолчанию» указано значение, с которым поставляется чарт; `—` означает, что значение пустое, пока вы его не задали.

Компоненты, которые предоставляет кластер

Чарт разворачивает приложение и, если `registry.enabled` равен `true`, хранилище образов контейнеров. Компоненты ниже кластер предоставляет до установки релиза. Чарт их не разворачивает:

- Ingress-контроллер, на который указывают `global.ingress.class` и, при включённом балансировщике, `global.codeHelm.loadbalancer.ingressService`;
- cert-manager с ClusterIssuer, который используют `global.codeHelm.ingress.tls` и `global.codeHelm.tls.internal`;
- PostgreSQL, Redis и S3-совместимое объектное хранилище, на которые указывают `global.psql`, `global.redis` и `global.appConfig.object_store`;
- Prometheus Operator и Grafana, если используются объекты из раздела Мониторинг;
- GitLab Runner.

Upstream-сабчарты `gitlab-runner`, `prometheus`, `certmanager`, `nginx-ingress`, `nginx-ingress-geo`, `haproxy`, `traefik`, `kubernetes-ingress`, `gitlab-zoekt` и `redis-ha` не входят в чарт. Если установить их ключ `install` или `enabled` в `true` либо `installCertmanager` в `true`, установка завершается ошибкой.

Secret из значений

`global.codeHelm.externalSecrets` создаёт Secret с учётными данными внешних сервисов из файла значений вместо Secret, создаваемых вручную перед установкой. Имя каждого Secret начинается с имени релиза, поэтому несколько релизов могут находиться в одном неймспейсе.

Если `registry.enabled` равен `true`, хранилищу образов контейнеров нужен Secret с настройками хранилища: либо из блока `registry.storage.s3` ниже, либо из `registry.storage.secret`; если отсутствуют оба, установка завершается ошибкой.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.externalSecrets.enabled`	булево	`false`	Создаёт ли чарт Secret, описанные в этой таблице. При `true` обязателен `registryDockerConfig` и заданы либо учётные данные `backup`, либо `gitlab.toolbox.backups.objectStorage.config.secret`; иначе установка завершается ошибкой.
`global.codeHelm.externalSecrets.psqlPassword`	строка	—	Пароль роли PostgreSQL из `global.psql.username`. Создаёт Secret `<release>-postgres` с ключом из `global.psql.password.key`; релиз использует этот Secret, если `global.psql.password.secret` не задан.
`global.codeHelm.externalSecrets.redisPassword`	строка	—	Пароль сервера Redis. Создаёт Secret с именем из `global.redis.auth.secret` (по умолчанию `<release>-redis-secret`) и ключом из `global.redis.auth.key`.
`global.codeHelm.externalSecrets.redisSentinelPassword`	строка	—	Пароль серверов Sentinel из `global.redis.sentin`

Ключ	Тип	По умолчанию	Описание
			`els`. Требует `global.redis.sentinelAuth.enabled: true`; без него установка завершается ошибкой. Создаёт Secret `<release>-redis-sentinel-secret` с ключом `secret` — это имя релиз читает, если `global.redis.sentinelAuth.secret` не задан; если заданы оба, установка завершается ошибкой.
`global.codeHelm.externalSecrets.praefectPassword`	строка	—	Пароль пользователя PostgreSQL для Praefect из `global.praefect.psql.user`. Требует `global.praefect.psql.host`; без него установка завершается ошибкой. Создаёт Secret с именем из `global.praefect.psql.dbSecret.secret` (по умолчанию `<release>-praefect-dbsecret`).
`global.codeHelm.externalSecrets.s3.provider`	строка	`AWS`	Имя серверной части хранилища в строке подключения к объектному хранилищу. Остаётся

Ключ	Тип	По умолчанию	Описание
			`AWS` для любого S3-совместимого хранилища.
`global.codeHelm.externalSecrets.s3.region`	строка	`us-east-1`	Регион объектного хранилища. Обязателен, если используется блок `s3`.
`global.codeHelm.externalSecrets.s3.aws_access_key_id`	строка	—	Идентификатор ключа доступа к объектному хранилищу. Его или секретный ключ достаточно задать, чтобы блок `s3` заработал: чарт создаёт Secret `<release>-s3-connection` с ключом `connection`, и `global.appConfig.object_store.connection` указывает на этот Secret, если вы не задали его сами.
`global.codeHelm.externalSecrets.s3.aws_secret_access_key`	строка	—	Секретный ключ доступа к объектному хранилищу. Обязателен, если используется блок `s3`.
`global.codeHelm.externalSecrets.s3.endpoint`	строка	—	URL точки доступа S3 со схемой, например `https://storage.example.com`. Обязателен, если

Ключ	Тип	По умолчанию	Описание
			используется блок `s3`.
`global.codeHelm.externalSecrets.s3.host`	строка	—	Хост точки доступа S3 без схемы, для хранилищ с адресацией через путь.
`global.codeHelm.externalSecrets.s3.path_style`	булево	`false`	Передаётся ли имя бакета в пути URL вместо имени хоста.
`global.codeHelm.externalSecrets.s3.aws_signature_version`	целое число	`4`	Версия подписи запросов, которую принимает хранилище.
`global.codeHelm.externalSecrets.backup.aws_access_key_id`	строка	—	Идентификатор ключа доступа к хранилищу, в которое загружаются архивы резервных копий. Вместе с секретным ключом создаёт Secret `<release>-s3cfg` с ключом `config`, который читает утилита резервного копирования. Хранилище может отличаться от указанного в блоке `s3`.
`global.codeHelm.externalSecrets.backup.aws_secret_access_key`	строка	—	Секретный ключ доступа к хранилищу резервных копий.
`global.codeHelm.externalSecrets.regist`	строка	—	Бакет, в котором хранятся слои

Ключ	Тип	По умолчанию	Описание
`ry.storage.s3.bucket`			образов контейнеров. Его или `accesskey` достаточно задать, чтобы блок заработал: чарт создаёт Secret `<release>-registry-storage-config`, хранилище образов читает его без дополнительных настроек, а `registry.storage.secret` не учитывается.
`global.codeHelm.externalSecrets.registry.storage.s3.accesskey`	строка	—	Ключ доступа к объектному хранилищу образов. Обязателен, если используется блок.
`global.codeHelm.externalSecrets.registry.storage.s3.secretkey`	строка	—	Секретный ключ объектного хранилища образов. Обязателен, если используется блок.
`global.codeHelm.externalSecrets.registry.storage.s3.region`	строка	`us-east-1`	Регион объектного хранилища образов. Обязателен, если используется блок.
`global.codeHelm.externalSecrets.registry.storage.s3.regionendpoint`	строка	—	URL точки доступа объектного хранилища образов, например `https://storage.example.com`. Обязателен, если используется блок.
	булево	`true`	

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.externalSecrets.registry.storage.s3.v4auth`			Подписывает ли хранилище образов запросы по AWS Signature Version 4.
`global.codeHelm.externalSecrets.registry.storage.s3.secure`	булево	`true`	Подключается ли хранилище образов к объектному хранилищу по HTTPS.
`global.codeHelm.externalSecrets.registry.storage.s3.rootdirectory`	строка	`/`	Префикс внутри бакета, под которым хранилище образов хранит данные.
`global.codeHelm.externalSecrets.registry.storage.s3.checksumDisabled`	булево	`false`	Опускает ли хранилище образов контрольные суммы в запросах копирования частей. Установите `true` для хранилищ, которые отклоняют запросы `UploadPartCopy` с контрольной суммой.
`global.codeHelm.externalSecrets.registry.storage.database.password`	строка	—	Пароль пользователя PostgreSQL базы метаданных хранилища образов. Создаёт Secret `<release>-registry-database-password` с ключом `password`.
`global.codeHelm.externalSecrets.registryDockerConfig`	строка	—	Содержимое `.dockerconfigjso` в кодировке Base64 с учётными данными хранилища образов, из которого чарт

Ключ	Тип	По умолчанию	Описание
			загружает образы. Создаёт Secret `<release>-image-pull-secret` типа `kubernetes.io/dockerconfigjson`. Обязателен при `enabled: true`.

Хранилище резервных копий

`global.codeHelm.backup` хранит адрес хранилища, в которое загружаются архивы резервных копий. Вместе с учётными данными из `global.codeHelm.externalSecrets.backup` он формирует Secret `<release>-s3cfg`.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.backup.region`	строка	`us-east-1`	Регион хранилища резервных копий.
`global.codeHelm.backup.endpoint`	строка	`http://minio:9000`	URL точки доступа хранилища резервных копий. Схема выбирает транспорт: `http://` подключается без TLS, `https://` с TLS.

TLS для Ingress

`global.codeHelm.ingress.tls` задаёт, откуда берутся сертификаты объектов Ingress.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.ingress.tls.mode`	строка	`certManager`	Источник сертификатов Ingress. `certManager`: чарт создаёт Certificate cert-manager для каждого включённого адреса и хранит их в Secret `<release>-tls` (веб-интерфейс), `<release>-registry-tls`, `<release>-pages-tls` и `<release>-kas-tls`; установка завершается ошибкой, если cert-manager отсутствует или переопределение `ingress.tls.secretName` указывает на другой Secret. `manual`: чарт не выпускает сертификаты для Ingress. Ingress каждого компонента использует Secret, указанный в его `ingress.tls.secretName` (`gitlab.webservice`, `registry`, `gitlab.gitlab-pages`, `gitlab.kas`); компонент без этого значения использует Secret из `global.ingress.tls.`

Ключ	Тип	По умолчанию	Описание
			`secretName`, и его сертификат покрывает все включённые хосты. Установка завершается ошибкой, если не задано ни `global.ingress.tls.secretName`, ни `gitlab.webservice.ingress.tls.secretName`.
`global.codeHelm.ingress.tls.clusterIssuer`	строка	`letsencrypt`	Имя ClusterIssuer cert-manager, который подписывает сертификаты Ingress в режиме `certManager`.

В режиме `manual` создайте Secret сами или добавьте `cert-manager.io/cluster-issuer: <ISSUER>` в `global.ingress.annotations`, где `<ISSUER>` — имя ClusterIssuer cert-manager. Эти аннотации есть у каждого Ingress чарта, и cert-manager выпускает сертификат в Secret, указанный в Ingress.

Балансировщик

`global.codeHelm.loadbalancer` разворачивает Deployment с HAProxy и Service типа LoadBalancer с именем `<release>-code-loadbalancer`. Service принимает порты 80, 443 и 22: HTTP и HTTPS передаются Ingress-контроллеру, SSH передаётся Service Git-оболочки внутри кластера. Инстанс получает один внешний адрес для Git по HTTPS и SSH.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.loadbalancer.enabled`	булево	`true`	Разворачивает ли чарт балансировщик. При `true` значение `gitlab.gitlab-shell.service.type` остаётся `ClusterIP`; иначе установка завершается ошибкой. При `false` откройте SSH сами, задав `gitlab.gitlab-shell.service.type` равным `LoadBalancer` или `NodePort`.
`global.codeHelm.loadbalancer.proxyProtocol`	булево	`false`	Заставляет ли HAProxy отправлять бэкендам заголовок PROXY protocol v2 с IP-адресом клиента. Требует Ingress-контроллер с `use-proxy-protocol: "true"` и Git-оболочку с `gitlab.gitlab-shell.sshDaemon: gitlab-sshd` и `gitlab.gitlab-shell.config.proxyProtocol: true`; сторона Git-оболочки проверяется при установке, без контроллера с приёмом заголовка весь web-трафик прекращается.

Ключ	Тип	По умолчанию	Описание
			Порядок включения описан в разделе Сохранение IP-адреса клиента.
`global.codeHelm.loadbalancer.frontendProxyProtocol`	булево	`false`	Заставляет ли HAProxy требовать заголовок PROXY protocol на входящих соединениях портов 80, 443 и 22. Включайте только при прокси с PROXY protocol перед балансировщиком и вместе со значением `proxyProtocol`.
`global.codeHelm.loadbalancer.ingressService`	строка	`ingress-nginx-controller.ingress-nginx.svc`	DNS-имя Service Ingress-контроллера, который получает трафик HTTP и HTTPS, когда `httpBackends` и `ingressClass` пусты.
`global.codeHelm.loadbalancer.ingressClass`	строка	—	Имя класса Ingress, контроллер которого получает трафик HTTP и HTTPS. Адресом становится `<ingressClass>-load-balancer`; используется, когда `httpBackends` пуст.
`global.codeHelm.loadbalancer.httpBackends`	массив строк	`[]`	Имена адресатов трафика HTTP и HTTPS. Каждое имя `N` превращается в адресат `N-load-balancer` с портами

Ключ	Тип	По умолчанию	Описание
			80 и 443. Имеет приоритет над `ingressClass` и `ingressService`.
`global.codeHelm.loadbalancer.replicas`	целое число	`1`	Число подов HAProxy при выключенном автомасштабировании.
`global.codeHelm.loadbalancer.resources`	объект	запросы `100m` CPU и `128Mi` памяти, лимиты `500m` CPU и `256Mi` памяти	Запросы и лимиты ресурсов контейнера HAProxy.
`global.codeHelm.loadbalancer.hpa.enabled`	булево	`false`	Масштабирует ли HorizontalPodAutoscaler Deployment с HAProxy.
`global.codeHelm.loadbalancer.hpa.minReplicas`	целое число	`1`	Нижняя граница автомасштабирования.
`global.codeHelm.loadbalancer.hpa.maxReplicas`	целое число	`2`	Верхняя граница автомасштабирования.
`global.codeHelm.loadbalancer.hpa.cpuUtilization`	целое число	`80`	Целевая средняя загрузка CPU в процентах от запроса.
`global.codeHelm.loadbalancer.hpa.memoryUtilization`	целое число	`80`	Целевая средняя загрузка памяти в процентах от запроса.
`global.codeHelm.loadbalancer.pdb.enabled`	булево	`false`	Сохраняет ли PodDisruptionBudget хотя бы один под HAProxy доступным

Ключ	Тип	По умолчанию	Описание
			при плановых прерываниях.
`global.codeHelm.loadbalancer.annotations`	объект	`{}`	Аннотации Service типа LoadBalancer, например аннотация адреса облачного балансировщика или MetalLB.
`global.codeHelm.loadbalancer.loadBalancerClass`	строка	—	Значение аннотации `service.kubernetes.io/load-balancer-class` этого Service, когда в кластере работает несколько реализаций балансировщика.

Внутренний TLS

`global.codeHelm.tls.internal` шифрует трафик между компонентами релиза сертификатами cert-manager: чарт создаёт самоподписанный Issuer, внутренний удостоверяющий центр и Certificate для веб-сервиса, Workhorse, Praefect и каждого виртуального хранилища Gitaly. Secret называются `<release>-webservice-tls`, `<release>-workhorse-tls`, `<release>-praefect-tls` и Secret Gitaly каждого виртуального хранилища.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.tls.internal.enabled`	булево	`true`	Создаёт ли чарт внутренний удостоверяющий центр и сертификаты компонентов. Требует cert-manager в кластере и upstream-флаги `gitlab.webservice.tls.enabled`, `global.workhorse.tls.enabled`, `global.gitaly.tls.enabled` и `global.praefect.tls.enabled` со значением `true` (значение чарта по умолчанию); если один из них `false`, установка завершается ошибкой. Не действует при `global.codeHelm.istio.enabled: true`.

Istio

`global.codeHelm.istio` включает релиз в сервисную сеть Istio: поды получают метку внедрения sidecar-контейнера, а объект PeerAuthentication задаёт режим mTLS неймспейса. Сеть заменяет внутренний TLS, поэтому `global.codeHelm.tls.internal.enabled`, `global.gitaly.tls.enabled`, `global.praefect.tls.enabled`, `global.workhorse.tls.enabled`, `gitlab.webservice.tls.enabled` и, если хранилище образов включено, `registry.tls.enabled` должны быть `false`; иначе установка завершается ошибкой.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.istio.enabled`	булево	`false`	Входят ли поды в сервисную сеть. Требует Istio с API `security.istio.io/v1beta1` и включённое внедрение sidecar-контейнеров для неймспейса. Задачи миграций, резервного копирования и переиндексации базы данных запускают прокси как native sidecar, для этого нужен Kubernetes 1.29 или новее; классический sidecar-контейнер не даёт завершившемуся поду задачи остановиться.
`global.codeHelm.istio.strict`	булево	`false`	Режим mTLS объекта PeerAuthentication: `false` соответствует `PERMISSIVE`, `true` соответствует `STRICT`. В строгом режиме порты Workhorse и SSH остаются в режиме `PERMISSIVE`, потому что Ingress-контроллер и Git-клиенты находятся вне сети.

Метрики через kube-rbac-proxy

`global.codeHelm.kubeRbacProxy` добавляет sidecar-контейнер kube-rbac-proxy в поды веб-сервиса, Sidekiq, Gitaly, Git-оболочки, Pages, Praefect, экспортера, хранилища образов и балансировщика. Контейнер отдаёт метрики своего пода на порту `https-metrics` и пропускает запрос только после проверки его токена через TokenReview и SubjectAccessReview. Чарт создаёт Service `<release>-<компонент>-metrics` и ServiceMonitor `<release>-<компонент>` для каждого компонента, Secret `<release>-prometheus-token`, из которого ServiceMonitor берут токен, и объекты RBAC, нужные контейнерам для проверок.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.kubeRbacProxy.enabled`	булево	`false`	Создаются ли sidecar-контейнеры, Service, ServiceMonitor и объекты RBAC. При `true` выключите upstream-объекты ServiceMonitor, чтобы Prometheus собирал метрики через прокси: `gitlab.webservice.serviceMonitor.enabled`, `gitlab.sidekiq.serviceMonitor.enabled`, `gitlab.gitaly.serviceMonitor.enabled`, `gitlab.gitlab-shell.serviceMonitor.enabled`, `gitlab.gitlab-pages.serviceMonitor.enabled`, `gitlab.praefect.serviceMonitor.enabled` и `gitlab.gitlab-exporter.serviceMonitor.enabled`.
`global.codeHelm.kubeRbacProxy.serviceAccountToken`	строка	—	Токен, который Prometheus предъявляет sidecar-контейнерам. Обязателен при `enabled: true`; иначе установка завершается ошибкой. ServiceAccount,

Ключ	Тип	По умолчанию	Описание
			которому принадлежит токен, нужен глагол `get` на подресурсы `deployments/prometheus-metrics` и `statefulsets/prometheus-metrics` в неймспейсе релиза.
`global.codeHelm.kubeRbacProxy.caSecret`	строка	—	Имя существующего Secret с сертификатом удостоверяющего центра в ключе `ca.crt`, по которому sidecar-контейнеры проверяют клиентские сертификаты. Если пусто, сертификат выпускает `clusterIssuer`.
`global.codeHelm.kubeRbacProxy.clusterIssuer`	строка	`selfsigned`	Имя ClusterIssuer cert-manager, который подписывает Certificate удостоверяющего центра `<release>-kube-rbac-proxy-ca`, когда `caSecret` пуст. Если пусты оба значения при `enabled: true`, установка завершается ошибкой.
	объект		

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.kubeRbacProxy.resources`		запросы `10m` CPU, `25Mi` памяти и `50Mi` временного хранилища	Запросы и лимиты ресурсов sidecar-контейнера.
`global.codeHelm.kubeRbacProxy.securityContext`	объект	без повышения привилегий, корневая файловая система только для чтения, все capabilities сброшены, профиль seccomp `RuntimeDefault`	Контекст безопасности sidecar-контейнера.

События аудита

`global.codeHelm.audit` добавляет sidecar-контейнер `audit` в поды веб-сервиса и Sidekiq. Контейнер читает журнал аудита своего пода и выводит каждое событие в стандартный вывод в формате CEF, поэтому стек логирования кластера собирает события из логов контейнера. Чтобы прочитать их напрямую:

```
d8 k -n code logs <POD_NAME> -c audit
```

Сами события описаны в разделе События аудита.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.audit.enabled`	булево	`true`	Добавляется ли sidecar-контейнер в поды веб-сервиса и Sidekiq.
`global.codeHelm.audit.resources`	объект	запросы и лимиты `100m` CPU и `128Mi` памяти	Запросы и лимиты ресурсов sidecar-контейнера.
`global.codeHelm.audit.securityContext`	объект	без повышения привилегий, все capabilities сброшены, профиль seccomp `RuntimeDefault`	Контекст безопасности sidecar-контейнера.

Мониторинг

`global.codeHelm.monitoring` создаёт объекты, которые подхватывают Prometheus Operator и Grafana.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.monitoring.prometheusRules`	булево	`false`	Создаёт ли чарт PrometheusRule `<release>-code-helm` с меткой `prometheus: d8-monitoring`. Правила предупреждают о недоступности веб-сервиса и Sidekiq, о сбоях и устаревании синхронизации LDAP и о сбоях pull-зеркал. Создаётся только при наличии в кластере API `monitoring.coreos.com/v1`.
`global.codeHelm.monitoring.dashboards`	булево	`true`	Создаёт ли чарт ConfigMap `<release>-grafana-dashboard` с меткой `grafana_dashboard: "1"` и дашбордом Grafana для инстанса. Sidecar-контейнер дашбордов Grafana подхватывает его из неймспейса релиза; для другого неймспейса расширьте список неймспейсов, которые он просматривает.

Расширенный поиск

`global.codeHelm.search` подключает инстанс к кластеру OpenSearch для поиска по коду и содержимому. Индексирование и сам поиск описаны в разделе Расширенный поиск.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.search.enabled`	булево	`false`	Включён ли расширенный поиск в конфигурации приложения.
`global.codeHelm.search.opensearch.url`	строка	`http://opensearch:9200/`	URL точки доступа OpenSearch.
`global.codeHelm.search.opensearch.credentials.username`	строка	—	Имя пользователя учётной записи OpenSearch. Вместе с паролем создаёт Secret `<release>-code-helm-opensearch-creds`, который монтируют поды веб-сервиса и Sidekiq; без обоих значений подключение анонимное.
`global.codeHelm.search.opensearch.credentials.password`	строка	—	Пароль учётной записи OpenSearch.
`global.codeHelm.search.opensearch.maxBulkSizeBytes`	целое число	`10485760`	Верхний размер одного пакетного запроса индексирования в байтах.
`global.codeHelm.search.opensearch.maxBulkConcurrency`	целое число	`10`	Число одновременно выполняемых пакетных запросов индексирования.
`global.codeHelm.search.opensearch.repositoryIndexerMaxConcurrency`	целое число	`30`	Число одновременно работающих процессов индексирования репозиториев.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.search.opensearch.indexedFileSizeLimitKb`	целое число	`1024`	Верхний размер файла, содержимое которого индексируется, в килобайтах. Более крупные файлы индексируются только по имени.
`global.codeHelm.search.opensearch.readTimeout`	целое число	`10`	Предельное время одного поискового запроса, в секундах. Запрос, превысивший его, не повторяется, и поиск возвращает ошибку.
`global.codeHelm.search.opensearch.adminRequestTimeout`	целое число	`60`	Предельное время запросов состояния индексов и пересоздания индексов, в секундах. Эти запросы не повторяются.
`global.codeHelm.search.opensearch.indexingRequestTimeout`	целое число	`1800`	Предельное время одного запроса индексирования, в секундах. Неудачный запрос индексирования повторяется, поэтому задача индексирования может ждать дольше этого значения.
`global.codeHelm.search.sidekiq.routingRules`	массив	`[["feature_category=fe_global_search", "global-search-`	Правила маршрутизации Sidekiq: каждый элемент — пара из селектора задач и

Ключ	Тип	По умолчанию	Описание
		`indexing"], ["*", "default"]]`	очереди, в которую они попадают. По умолчанию задачи индексирования идут в очередь `global-search-indexing`, остальные в `default`.

Чтобы задачи индексирования выполнялись в отдельных подах, добавьте под Sidekiq для этой очереди:

```
gitlab:
  sidekiq:
    pods:
      - name: all-in-1
      - name: global-search
        queues: global-search-indexing
```

Образы контейнеров

Образ каждой рабочей нагрузки закреплён в чарте при сборке. `global.codeHelm.images` заменяет закреплённый образ компонента, например когда образы скопированы в хранилище образов внутри сети.

Ключ	Тип	По умолчанию	Описание
`global.codeHelm.images.<компонент>`	строка	—	Полная ссылка на образ компонента, `registry/repository:tag` или `registry/repository@sha256:...`; она полностью заменяет закреплённую ссылку. Имена компонентов: `webservice-default`, `workhorse`, `sidekiq-all-in-1-v2`, `sidekiq-global-search-v2`, `gitaly-default`, `gitaly-init-cgroups`, `praefect`, `registry`, `shell`, `toolbox`, `rails`, `exporter`, `mailroom`, `pages`, `base`, `certificates`, `kube-rbac-proxy`, `haproxy`, `vector`, `kubectl`, `cfssl-self-sign`, `kas`. Под Sidekiq из `gitlab.sidekiq.pods` берёт образ из ключа `sidekiq-<имя пода>-v2`, а при его отсутствии из `sidekiq-all-in-1-v2`.
`global.codeHelm.nameOverride`	строка	`<release>-code-helm`	Префикс имён объектов, которые

Ключ	Тип	По умолчанию	Описание
			чарт добавляет к upstream-чарту: ConfigMap аудита, Secret OpenSearch, PrometheusRule, а также Issuer и удостоверяющий центр внутреннего TLS.

Шифрование подключения к PostgreSQL

`global.psql.sslMode` и `global.psql.ssl` задают, как приложение подключается к PostgreSQL. `global.praefect.psql.sslMode` делает то же для базы данных Praefect и по умолчанию тоже равен `prefer`.

Ключ	Тип	По умолчанию	Описание
`global.psql.sslMode`	строка	`prefer`	Режим SSL libpq для подключений приложения: `disable`, `allow`, `prefer`, `require`, `verify-ca` или `verify-full`. Если значение пустое, а `global.psql.ssl` задан, действует режим `verify-ca`.
`global.psql.ssl.secret`	строка	—	Имя Secret с сертификатами для взаимной проверки TLS. Если используется блок `ssl`, обязателен каждый ключ блока.
`global.psql.ssl.serverCA`	строка	—	Ключ внутри этого Secret с сертификатом удостоверяющего центра сервера.
`global.psql.ssl.clientCertificate`	строка	—	Ключ внутри этого Secret с клиентским сертификатом.
`global.psql.ssl.clientKey`	строка	—	Ключ внутри этого Secret с закрытым ключом клиента.

Провайдеры аутентификации

Чарт принимает upstream-поля элемента `global.appConfig.omniauth.providers` (`name`, `label`, `icon`, `secret`, `key`, `args`) и поля групп ниже. Если элемент указывает `secret`, провайдер читается из этого Secret, и поля групп записываются в его YAML. Полные примеры элементов приведены в разделе SAML.

Ключ	Тип	По умолчанию	Описание
`global.appConfig.omniauth.providers[].allowed_groups`	массив строк	—	Группы провайдера идентификации, участники которых могут войти. Если не задано, войти может любой аутентифицированный пользователь.
`global.appConfig.omniauth.providers[].admin_groups`	массив строк	`[]`	Группы, участники которых получают права администратора инстанса.
`global.appConfig.omniauth.providers[].auditor_groups`	массив строк	`[]`	Группы, участники которых получают права аудитора инстанса.
`global.appConfig.omniauth.providers[].groups_attribute`	строка	`Groups`	Имя атрибута SAML, который несёт членство в группах.
`global.appConfig.omniauth.providers[].external_groups`	массив строк	—	Группы, участники которых отмечаются как внешние пользователи.

Сервер LDAP в `global.appConfig.ldap.servers.<имя>` принимает upstream-поля (`host`, `port`, `uid`, `base`, `bind_dn`, `password`, `active_directory` и остальные) и блок `group_sync`, который сопоставляет группы LDAP с группами инстанса. Синхронизация описана в разделе LDAP.

Ключ	Тип	По умолчанию	Описание
`global.appConfig.ldap.servers.<имя>.group_sync.create_groups`	булево	`false`	Создаются ли при синхронизации группы, которые есть в LDAP и отсутствуют на инстансе.
`global.appConfig.ldap.servers.<имя>.group_sync.base`	строка	—	Базовый DN поиска групп в LDAP.
`global.appConfig.ldap.servers.<имя>.group_sync.filter`	строка	—	Фильтр LDAP, который отбирает группы для синхронизации.
`global.appConfig.ldap.servers.<имя>.group_sync.scope`	строка или целое число	—	Область поиска групп в LDAP: `0` — базовый объект, `1` — один уровень под ним, `2` — всё поддерево.
`global.appConfig.ldap.servers.<имя>.group_sync.top_level_group`	строка	—	Путь группы на инстансе, внутри которой создаются синхронизированные группы.
`global.appConfig.ldap.servers.<имя>.group_sync.name_mask`	строка	—	Регулярное выражение, которое применяется к `cn` группы LDAP и извлекает имя группы на инстансе.
`global.appConfig.ldap.servers.<имя>.group_sync.owner`	строка	`root`	Имя пользователя, который становится владельцем групп, созданных синхронизацией.
`global.appConfig.ldap.servers.<имя>.gr`	строка	—	Регулярное выражение, которое

Ключ	Тип	По умолчанию	Описание
`oup_sync.role_mapping[].by_name`			сопоставляется с именем группы LDAP.
`global.appConfig.ldap.servers.<имя>.group_sync.role_mapping[].gitlab_role`	строка	—	Роль, которую получают участники совпавшей группы: `guest`, `reporter`, `developer`, `maintainer` или `owner`.
`global.appConfig.ldap.servers.<имя>.group_sync.prefix.attribute`	строка	—	Атрибут LDAP, значение которого добавляется перед именем группы.
`global.appConfig.ldap.servers.<имя>.group_sync.prefix.default`	строка	—	Префикс, который используется при отсутствии атрибута.
`global.appConfig.ldap.servers.<имя>.group_sync.prefix.name_mask`	строка	—	Регулярное выражение, которое извлекает префикс из значения атрибута.

Изменённые значения upstream-чарта по умолчанию

Чарт поставляет перечисленные ниже значения upstream-чарта с другими значениями по умолчанию. Каждая строка называет, чего изменение требует от кластера или файла значений.

Ключ	В upstream-чарте	В чарте	Следствие
`global.registry.enabled`, `registry.enabled`	`true`	`false`	Хранилище образов контейнеров не разворачивается, и приложение не показывает хранилище образов; чтобы включить его, установите оба значения в `true` и настройте хранилище, см. раздел Хранилище образов контейнеров.
`global.ingress.class`	не задан	`nginx`	Объекты Ingress адресованы классу `nginx`; задайте другой класс, если в кластере другой контроллер.
`global.appConfig.object_store.enabled`	`false`	`true`	Загрузки, артефакты, объекты LFS, пакеты и остальные хранилища находятся в объектном хранилище; требуются подключение через `global.codeHelm.externalSecrets.s3` или `global.appConfig.object_store.connection` и бакет для каждого хранилища.
`global.appConfig.externalDiffs.enabled`, `global.appConfig.te`	`false`	`true`	Диффам запросов на слияние, состоянию Terraform и прокси

Ключ	В upstream-чарте	В чарте	Следствие
`rraformState.enabled`, `global.appConfig.dependencyProxy.enabled`			зависимостей нужен бакет в `global.appConfig.<хранилище>.bucket`.
`global.praefect.enabled`	`false`	`true`	Gitaly работает за Praefect с одним виртуальным хранилищем `default`, одной репликой Gitaly (`global.praefect.virtualStora` в upstream-чарте `3`) и одним подом Praefect (`gitlab.praefect.replicas`, в upstream-чарте `2`). Praefect нужна собственная база данных PostgreSQL в `global.praefect.psql.host`, `user` и `dbName`.
`global.praefect.psql.sslMode`	`disable`	`prefer`	Подключение Praefect использует TLS, если сервер его предлагает.
`global.gitaly.tls.enabled`, `global.praefect.tls.enabled`, `global.workhorse.tls.enabled`, `gitlab.webservice.tls.enabled`	`false`	`true`	Компоненты ожидают сертификаты из раздела Внутренний TLS; оставьте `true`, пока `global.codeHelm.tls.internal.enabled` равен `true`, и установите `false` для Istio.

Ключ	В upstream-чарте	В чарте	Следствие
`global.kas.enabled`	`true`	`false`	Сервер агентов Kubernetes не разворачивается, сертификат `<release>-kas-tls` не создаётся.
`global.gatewayApi.enabled`, `global.gatewayApi.installEnvoy`	`true`	`false`	Чарт публикует инстанс через объекты Ingress; объекты Gateway API и Envoy Gateway не создаются.
`gitlab.gitlab-shell.service.type`	`LoadBalancer`	`ClusterIP`	SSH публикуется через Балансировщик; меняйте тип только при выключенном балансировщике.
`gitlab.gitlab-shell.sshDaemon`	`openssh`	`gitlab-sshd`	SSH обслуживает демон gitlab-sshd; он декодирует заголовок PROXY protocol от балансировщика и принимает соединения без заголовка.
`gitlab.gitlab-shell.config.proxyProtocol`	`false`	`true`	gitlab-sshd ожидает заголовок PROXY protocol от балансировщика; вместе со `sshDaemon` составляет сторону Git-оболочки для сохранения IP-адреса клиента. Установка завершается ошибкой, если при

Ключ	В upstream-чарте	В чарте	Следствие
			включённом `global.codeHelm.loadbalancer.proxyProtocol` значения переопределены на upstream-варианты.
`gitlab.gitaly.cgroups.enabled`	`false`	`true`	Gitaly запускает процессы Git в cgroups: `2 GiB` и `150000` единиц квоты CPU для родительской группы, `100` групп репозиториев по `1 GiB` и `75000` единиц квоты CPU каждая. Под Gitaly запускает init-контейнер, которому нужен доступ на запись к файловой системе cgroup узла.
`gitlab.toolbox.backups.cron.extraArgs`	не задан	`--repositories-server-side`	Gitaly загружает резервные копии репозиториев в бакет резервных копий напрямую, поэтому бакет должен быть доступен из подов Gitaly.

Масштабирование

Значения ниже подбирают ресурсы Helm-релиза Deckhouse Code под ожидаемое число пользователей. У каждого размера есть одиночный вариант и отказоустойчивый вариант. В одиночном варианте каждый компонент работает в одной реплике. В отказоустойчивом варианте каждый компонент работает минимум в двух репликах, Praefect — в трёх; объекты HorizontalPodAutoscaler добавляют реплики при росте нагрузки на CPU, объекты

PodDisruptionBudget сохраняют поды доступными, пока узлы освобождаются от подов, а каждый из двух подов Gitaly хранит полную копию всех репозиториев (см. раздел Хранилище Git).

Значения `gitlab.gitlab-pages` и `registry` действуют после включения Pages и хранилища образов контейнеров; по умолчанию оба компонента выключены (см. разделы Хранилище образов контейнеров и Pages).

Применение значений

Сохраните выбранный вариант в файл, например `scaling.yaml`, и передайте его команде установки после файла значений установки. Helm объединяет файлы в порядке флагов `-f`; ключ, заданный в нескольких файлах, получает значение из последнего:

```
helm upgrade --install code oci://<CHART_REGISTRY>/code \
  --version <CHART_VERSION> \
  -n code --create-namespace \
  -f values.yaml \
  -f scaling.yaml
```

`values.yaml` — файл значений установки из раздела Быстрый старт. Та же команда меняет размер работающего релиза: поды компонентов, значения которых изменились, пересоздаются.

300 пользователей

Значения для 300 пользователей:

Одиночный

```
global:
  codeHelm:
    loadbalancer:
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi

gitlab:
  webservice:
    workerProcesses: 4
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 7G
      requests:
        cpu: 4
        memory: 7G
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600M
      requests:
        cpu: 10m
        memory: 24M
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
  gitaly:
    resources:
      limits:
        cpu: 1500m
        memory: 2Gi
      requests:
        cpu: 1500m
```

```
        memory: 2Gi
    cgroups:
      memoryBytes: 2147483648
      cpuQuotaUs: 150000
      repositories:
        memoryBytes: 1073741824
        cpuQuotaUs: 75000
  praefect:
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi
```

Отказоустойчивый

```
global:
  codeHelm:
    loadbalancer:
      replicas: 2
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi
      pdb:
        enabled: true
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 2
        maxUnavailable: 1

gitlab:
  webservice:
    workerProcesses: 3
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 5500Mi
      requests:
        cpu: 3
        memory: 5500Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1
    minReplicas: 2
  sidekiq:
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 350m
    minReplicas: 2
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
```

```
        memory: 600M
      requests:
        cpu: 10m
        memory: 24M
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 100m
    minReplicas: 2
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
    hpa:
      enabled: true
      minReplicas: 2
      cpu:
        targetAverageValue: 100m
  gitaly:
    resources:
      limits:
        cpu: 1500m
        memory: 2Gi
      requests:
        cpu: 1500m
        memory: 2Gi
    cgroups:
      memoryBytes: 2147483648
      cpuQuotaUs: 150000
      repositories:
        memoryBytes: 1073741824
        cpuQuotaUs: 75000
  praefect:
    replicas: 3
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi

registry:
  hpa:
    enabled: true
```

```
    minReplicas: 2
    cpu:
      targetAverageUtilization: 75
```

500 пользователей

Значения для 500 пользователей:

Одиночный

```
global:
  codeHelm:
    loadbalancer:
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi

gitlab:
  webservice:
    workerProcesses: 4
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 7G
      requests:
        cpu: 4
        memory: 7G
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600M
      requests:
        cpu: 10m
        memory: 24M
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
  gitaly:
    resources:
      limits:
        cpu: 1500m
        memory: 2Gi
      requests:
        cpu: 1500m
```

```
        memory: 2Gi
    cgroups:
      memoryBytes: 2147483648
      cpuQuotaUs: 150000
      repositories:
        memoryBytes: 1073741824
        cpuQuotaUs: 75000
  praefect:
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi
```

Отказоустойчивый

```
global:
  codeHelm:
    loadbalancer:
      replicas: 2
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi
      pdb:
        enabled: true
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 2
        maxUnavailable: 1

gitlab:
  webservice:
    workerProcesses: 4
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 7G
      requests:
        cpu: 4
        memory: 7G
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1
    minReplicas: 2
  sidekiq:
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 350m
    minReplicas: 2
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
```

```
        memory: 600M
      requests:
        cpu: 10m
        memory: 24M
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 100m
    minReplicas: 2
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
    hpa:
      enabled: true
      minReplicas: 2
      cpu:
        targetAverageValue: 100m
  gitaly:
    resources:
      limits:
        cpu: 1500m
        memory: 2Gi
      requests:
        cpu: 1500m
        memory: 2Gi
    cgroups:
      memoryBytes: 2147483648
      cpuQuotaUs: 150000
      repositories:
        memoryBytes: 1073741824
        cpuQuotaUs: 75000
  praefect:
    replicas: 3
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi

registry:
  hpa:
    enabled: true
```

```
    minReplicas: 2
    cpu:
      targetAverageUtilization: 75
```

1000 пользователей

Значения для 1000 пользователей:

Одиночный

```
global:
  codeHelm:
    loadbalancer:
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi

gitlab:
  webservice:
    workerProcesses: 6
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 10G
      requests:
        cpu: 6
        memory: 10G
  sidekiq:
    resources:
      limits:
        memory: 3G
      requests:
        cpu: 1
        memory: 1500Mi
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600Mi
      requests:
        cpu: 32m
        memory: 50Mi
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
```

```
  gitaly:
    resources:
      limits:
        cpu: 2000m
        memory: 4000Mi
      requests:
        cpu: 2000m
        memory: 4000Mi
    cgroups:
      memoryBytes: 4194304000
      cpuQuotaUs: 200000
      repositories:
        memoryBytes: 2097152000
        cpuQuotaUs: 100000
  praefect:
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi
```

Отказоустойчивый

```
global:
  codeHelm:
    loadbalancer:
      replicas: 2
      resources:
        limits:
          cpu: 500m
          memory: 256Mi
        requests:
          cpu: 250m
          memory: 128Mi
      pdb:
        enabled: true
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 2
        maxUnavailable: 1

gitlab:
  webservice:
    workerProcesses: 5
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 8500Mi
      requests:
        cpu: 5
        memory: 8500Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1200m
    minReplicas: 2
    maxReplicas: 4
    maxUnavailable: 2
  sidekiq:
    resources:
      limits:
        memory: 2G
      requests:
        cpu: 1
        memory: 1500Mi
    hpa:
```

```
      enabled: true
      cpu:
        targetAverageValue: 600m
    minReplicas: 2
    maxReplicas: 4
    maxUnavailable: 1
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600Mi
      requests:
        cpu: 32m
        memory: 50Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 100m
    minReplicas: 2
    maxReplicas: 4
    maxUnavailable: 1
  gitlab-pages:
    resources:
      limits:
        cpu: 1500m
        memory: 2G
      requests:
        cpu: 900m
        memory: 1G
    hpa:
      enabled: true
      minReplicas: 2
      maxReplicas: 3
      cpu:
        targetAverageValue: 100m
    maxUnavailable: 1
  gitaly:
    resources:
      limits:
        cpu: 2000m
        memory: 4000Mi
      requests:
        cpu: 2000m
        memory: 4000Mi
    cgroups:
      memoryBytes: 4194304000
      cpuQuotaUs: 200000
      repositories:
        memoryBytes: 2097152000
```

```
        cpuQuotaUs: 100000
  praefect:
    replicas: 3
    resources:
      limits:
        cpu: 300m
        memory: 600Mi
      requests:
        cpu: 100m
        memory: 128Mi

registry:
  hpa:
    enabled: true
    minReplicas: 2
    maxReplicas: 3
    cpu:
      targetAverageUtilization: 75
  maxUnavailable: 2
```

3000 пользователей

Значения для 3000 пользователей:

Одиночный

```
global:
  codeHelm:
    loadbalancer:
      resources:
        limits:
          cpu: 1000m
          memory: 512Mi
        requests:
          cpu: 500m
          memory: 256Mi

gitlab:
  webservice:
    workerProcesses: 8
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 13G
      requests:
        cpu: 8
        memory: 13G
  sidekiq:
    resources:
      limits:
        memory: 3G
      requests:
        cpu: 1
        memory: 1500Mi
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600Mi
      requests:
        cpu: 32m
        memory: 50Mi
  gitlab-pages:
    resources:
      limits:
        memory: 3G
      requests:
        cpu: 900m
        memory: 1G
  gitaly:
```

```
    resources:
      limits:
        cpu: 6000m
        memory: 16000Mi
      requests:
        cpu: 6000m
        memory: 16000Mi
    cgroups:
      memoryBytes: 16777216000
      cpuQuotaUs: 600000
      repositories:
        memoryBytes: 8388608000
        cpuQuotaUs: 300000
  praefect:
    resources:
      limits:
        cpu: 600m
        memory: 1200Mi
```

Отказоустойчивый

```
global:
  codeHelm:
    loadbalancer:
      replicas: 2
      resources:
        limits:
          cpu: 1000m
          memory: 512Mi
        requests:
          cpu: 500m
          memory: 256Mi
      hpa:
        enabled: true
        minReplicas: 2
        maxReplicas: 3
        cpuUtilization: 80
        memoryUtilization: 80
      pdb:
        enabled: true
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 2
        maxUnavailable: 1

gitlab:
  webservice:
    workerProcesses: 6
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 10G
      requests:
        cpu: 6
        memory: 10G
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1600m
    minReplicas: 2
    maxReplicas: 4
    maxUnavailable: 2
  sidekiq:
    resources:
```

```
      limits:
        memory: 3G
      requests:
        cpu: 1
        memory: 1500Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 700m
    minReplicas: 4
    maxReplicas: 8
    maxUnavailable: 1
  gitlab-shell:
    resources:
      limits:
        cpu: 500m
        memory: 600Mi
      requests:
        cpu: 32m
        memory: 50Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1600m
    minReplicas: 3
    maxReplicas: 6
    maxUnavailable: 1
  gitlab-pages:
    resources:
      limits:
        memory: 3G
      requests:
        cpu: 900m
        memory: 1G
    hpa:
      enabled: true
      minReplicas: 2
      maxReplicas: 6
      cpu:
        targetAverageValue: 100m
    maxUnavailable: 1
  gitaly:
    resources:
      limits:
        cpu: 6000m
        memory: 16000Mi
      requests:
        cpu: 6000m
        memory: 16000Mi
```

```
    cgroups:
      memoryBytes: 16777216000
      cpuQuotaUs: 600000
      repositories:
        memoryBytes: 8388608000
        cpuQuotaUs: 300000
  praefect:
    replicas: 3
    resources:
      limits:
        cpu: 600m
        memory: 1200Mi

registry:
  hpa:
    enabled: true
    minReplicas: 3
    maxReplicas: 5
    cpu:
      targetAverageUtilization: 75
  maxUnavailable: 3
```

5000 пользователей

Значения для 5000 пользователей:

Одиночный

```
global:
  codeHelm:
    loadbalancer:
      resources:
        limits:
          cpu: 1500m
          memory: 1Gi
        requests:
          cpu: 750m
          memory: 512Mi

gitlab:
  webservice:
    workerProcesses: 10
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 16G
      requests:
        cpu: 10
        memory: 16G
  sidekiq:
    resources:
      limits:
        memory: 3G
      requests:
        cpu: 1
        memory: 1500Mi
  gitlab-shell:
    resources:
      limits:
        cpu: 384m
        memory: 250Mi
      requests:
        cpu: 32m
        memory: 50Mi
  gitlab-pages:
    resources:
      limits:
        memory: 4G
      requests:
        cpu: 900m
        memory: 2G
  gitaly:
```

```
    resources:
      limits:
        cpu: 6000m
        memory: 16000Mi
      requests:
        cpu: 6000m
        memory: 16000Mi
    cgroups:
      memoryBytes: 16777216000
      cpuQuotaUs: 600000
      repositories:
        memoryBytes: 8388608000
        cpuQuotaUs: 300000
  praefect:
    resources:
      limits:
        cpu: 600m
        memory: 1200Mi
```

Отказоустойчивый

```
global:
  codeHelm:
    loadbalancer:
      replicas: 3
      resources:
        limits:
          cpu: 1500m
          memory: 1Gi
        requests:
          cpu: 750m
          memory: 512Mi
      hpa:
        enabled: true
        minReplicas: 3
        maxReplicas: 4
        cpuUtilization: 80
        memoryUtilization: 80
      pdb:
        enabled: true
  praefect:
    virtualStorages:
      - name: default
        gitalyReplicas: 2
        maxUnavailable: 1

gitlab:
  webservice:
    workerProcesses: 8
    puma:
      threads:
        min: 8
        max: 8
    resources:
      limits:
        memory: 13G
      requests:
        cpu: 8
        memory: 13G
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 1600m
    minReplicas: 2
    maxReplicas: 5
    maxUnavailable: 1
  sidekiq:
    resources:
```

```
      limits:
        memory: 3G
      requests:
        cpu: 1
        memory: 1500Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 700m
    minReplicas: 6
    maxReplicas: 12
    maxUnavailable: 1
  gitlab-shell:
    resources:
      limits:
        cpu: 384m
        memory: 250Mi
      requests:
        cpu: 32m
        memory: 50Mi
    hpa:
      enabled: true
      cpu:
        targetAverageValue: 100m
    minReplicas: 6
    maxReplicas: 12
    maxUnavailable: 2
  gitlab-pages:
    resources:
      limits:
        memory: 4G
      requests:
        cpu: 900m
        memory: 2G
    hpa:
      enabled: true
      minReplicas: 2
      maxReplicas: 6
      cpu:
        targetAverageValue: 100m
    maxUnavailable: 1
  gitaly:
    resources:
      limits:
        cpu: 6000m
        memory: 16000Mi
      requests:
        cpu: 6000m
        memory: 16000Mi
```

```
    cgroups:
      memoryBytes: 16777216000
      cpuQuotaUs: 600000
      repositories:
        memoryBytes: 8388608000
        cpuQuotaUs: 300000
  praefect:
    replicas: 3
    resources:
      limits:
        cpu: 600m
        memory: 1200Mi

registry:
  hpa:
    enabled: true
    minReplicas: 4
    maxReplicas: 8
    cpu:
      targetAverageUtilization: 75
  maxUnavailable: 4
```

Диагностика

Каждый раздел ниже описывает один симптом, его причину и действие, которое его устраняет. Команды рассчитаны на релиз `code` в неймспейсе `code`.

Установка завершается ошибкой проверки значений

`helm upgrade` завершается до создания объектов, а вывод заканчивается сообщением проверки значений чарта.

Причина. Значения противоречат друг другу или пропущено обязательное поле. Чарт проверяет их первыми и выводит все замечания сразу.

Решение. Измените значения, названные в сообщении, и повторите команду. Сообщения и их исправления:

Сообщение	Исправление
`is not bundled with Deckhouse Code`	Удалите `install: true` или `enabled: true` у сабчарта, который называет сообщение, либо `installCertmanager: true`, и предоставьте компонент в кластере; см. раздел Компоненты, которые предоставляет кластер.
`Registry storage secret is required`	Заполните `global.codeHelm.externalSecrets.registry.storage.s3`, укажите существующий Secret в `registry.storage.secret` или установите `registry.enabled: false`.
`global.codeHelm.externalSecrets.registryDockerConfig is required`	Запишите `.dockerconfigjson` хранилища образов в кодировке Base64 в `global.codeHelm.externalSecrets.registryDockerConfig`.
`Backup object storage is not configured`	Заполните `global.codeHelm.externalSecrets.backup` ключами доступа к хранилищу резервных копий или укажите существующий Secret с конфигурацией s3cmd в `gitlab.toolbox.backups.objectStorage.config.secret`.
`praefectPassword is set but global.praefect.psql.host is empty`	Укажите в `global.praefect.psql.host`, `user` и `dbName` базу данных Praefect.
`redisSentinelPassword is set but global.redis.sentinelAuth.enabled is false`	Установите `global.redis.sentinelAuth.enabled: true` и перечислите серверы Sentinel в `global.redis.sentinels` или удалите `global.codeHelm.externalSecrets.redisSentinelPassword`.
`Two sources for the Redis Sentinel password`	Удалите `global.redis.sentinelAuth.secret` и `key`, чтобы Secret создал чарт, или удалите `global.codeHelm.externalSecrets.redisSentinelPassword` и оставьте существующий Secret.

Сообщение	Исправление
`global.codeHelm.externalSecrets.s3.<поле> is required`	Заполните каждое поле блока `s3`: `provider`, `region`, `endpoint`, `aws_access_key_id` и `aws_secret_access_key`.
`codeHelm.ingress.tls.mode=certManager requires cert-manager`	Установите cert-manager или переключите `global.codeHelm.ingress.tls.mode` в `manual` и передайте сертификаты как Secret.
`codeHelm.ingress.tls.mode=manual: no TLS secret name configured`	Задайте `global.ingress.tls.secretName` или `gitlab.webservice.ingress.tls.secretName` вместе с `ingress.tls.secretName` остальных включённых компонентов.
`generates Certificate resources with fixed secret names, but the following overrides were found`	Удалите переопределения `ingress.tls.secretName` или переключитесь в ручной режим TLS.
`Only the default internal ClusterIP shell Service is compatible with the loadbalancer`	Верните `gitlab.gitlab-shell.service.type` в `ClusterIP` или выключите балансировщик значением `global.codeHelm.loadbalancer.enabled: false`.
`Istio service mesh and internal TLS are mutually exclusive`	С Istio установите `global.codeHelm.tls.internal.enabled`, `global.gitaly.tls.enabled`, `global.praefect.tls.enabled`, `global.workhorse.tls.enabled`, `gitlab.webservice.tls.enabled` и, если хранилище образов включено, `registry.tls.enabled` в `false`.
`the following upstream TLS flags must also be enabled`	Верните перечисленные в сообщении флаги в `true` или выключите внутренний TLS значением `global.codeHelm.tls.internal.enabled: false`.
`kubeRbacProxy.enabled=true requires caSecret or clusterIssuer`	Укажите в `global.codeHelm.kubeRbacProxy.clusterIssuer` существующий ClusterIssuer или в `caSecret` Secret с ключом `ca.crt`.

Сообщение	Исправление
`kubeRbacProxy.enabled=true requires kubeRbacProxy.serviceAccountToken`	Запишите токен ServiceAccount Prometheus в `global.codeHelm.kubeRbacProxy.serviceAccountToken`.

У Service балансировщика нет внешнего адреса

`d8 k -n code get service code-code-loadbalancer` показывает `<pending>` в колонке `EXTERNAL-IP`.

Причина. В кластере нет реализации, которая выдаёт адреса Service типа LoadBalancer, либо реализации нужна аннотация для выбора адреса.

Решение. Прочитайте события Service:

```
d8 k -n code describe service code-code-loadbalancer
```

Добавьте аннотацию, которую ожидает реализация, в `global.codeHelm.loadbalancer.annotations`, например `metallb.universe.tf/loadBalancerIPs` для MetalLB. Если такой реализации в кластере нет, выключите балансировщик значением `global.codeHelm.loadbalancer.enabled: false`, направьте DNS на Ingress-контроллер и откройте SSH значением `gitlab.gitlab-shell.service.type: NodePort`.

Поды остаются в состоянии ImagePullBackOff

`d8 k -n code get pods` показывает поды в состоянии `ImagePullBackOff` или `ErrImagePull`.

Причина. Secret для загрузки образов `code-image-pull-secret`, собранный из `global.codeHelm.externalSecrets.registryDockerConfig`, содержит неверные или истёкшие учётные данные, либо узлы не достигают хранилища образов.

Решение. Прочитайте ошибку загрузки в событиях пода:

```
d8 k -n code describe pod <POD_NAME>
```

Ответ `unauthorized` указывает на учётные данные: заново закодируйте `.dockerconfigjson` командой `base64 -w0` и обновите релиз. Ошибка соединения указывает на сетевой путь от узлов к хранилищу образов.

Поды остаются в состоянии Pending

`d8 k -n code get pods` показывает поды Gitaly или Praefect в состоянии `Pending`.

Причина. Ни один StorageClass не подходит под `gitlab.gitaly.persistence.storageClass`, в кластере нет StorageClass по умолчанию, либо ни у одного узла не хватает CPU и памяти для запросов пода.

Решение. Прочитайте события пода:

```
d8 k -n code describe pod <POD_NAME>
```

Событие `FailedScheduling` с именем PersistentVolumeClaim указывает на StorageClass; выведите классы командой `d8 k get storageclass`. Событие с упоминанием CPU или памяти указывает на ресурсы узлов.

Сертификат не выпускается

`d8 k -n code get certificate` показывает `False` в колонке `READY` для `code-tls`, `code-registry-tls` или `code-pages-tls`, а браузер сообщает о недействительном сертификате.

Причина. ClusterIssuer из `global.codeHelm.ingress.tls.clusterIssuer` не существует, либо издатель не может пройти проверку, потому что DNS-имя ещё не разрешается в адрес балансировщика.

Решение. Прочитайте состояние Certificate и издателя:

```
d8 k -n code describe certificate code-tls
d8 k get clusterissuer
```

Создайте ClusterIssuer или укажите существующий и убедитесь, что DNS-записи указывают на адрес балансировщика. cert-manager повторяет выпуск сам, когда причина устранена.

Задача миграций завершается ошибкой

`d8 k -n code get jobs` показывает задачу `code-migrations-<суффикс>` с `0/1` завершений, а поды веб-сервиса и Sidekiq ждут её.

Причина. PostgreSQL недоступен по адресу `global.psql.host`, пароль в `global.codeHelm.externalSecrets.psqlPassword` или в Secret из `global.psql.password.secret` неверен, у роли нет прав на базу данных, либо релиз пропустил версию Deckhouse Code, от которой зависит более поздняя миграция.

Решение. Прочитайте лог задачи:

```
d8 k -n code logs job/code-migrations-<суффикс>
```

Сначала устраните проблему подключения или учётных данных. Если лог называет отсутствующую более раннюю миграцию, следуйте порядку версий из раздела Обновление.

Поды Praefect перезапускаются

Поды `code-praefect` перезапускаются, а в их логах сообщение о неудачном подключении к PostgreSQL.

Причина. База данных Praefect из `global.praefect.psql.host`, `dbName` и `user` не существует, либо пароль из `global.codeHelm.externalSecrets.praefectPassword` неверен. У Praefect собственная база данных, базу данных приложения он не использует.

Решение. Прочитайте лог пода:

```
d8 k -n code logs statefulset/code-praefect
```

Создайте базу данных и роль на сервере PostgreSQL, задайте пароль и обновите релиз.

Объектное хранилище отклоняет запросы

Загрузка файлов и артефактов CI/CD завершается ошибкой, а в логах веб-сервиса или Sidekiq виден ответ хранилища `AccessDenied` или `NoSuchBucket`.

Причина. Подключение в `global.codeHelm.externalSecrets.s3` или в Secret из `global.appConfig.object_store.connection.secret` содержит неверную точку доступа, регион или ключ, либо бакет из `global.appConfig.<хранилище>.bucket` не существует.

Решение. Сверьте подключение с консолью хранилища и убедитесь, что каждый бакет из раздела Требования существует, а у ключей есть право записи в него. Для хранилища, которое адресует бакеты в пути URL, добавьте в подключение `path_style: true`.

Helm upgrade завершается по тайм-ауту

`helm upgrade` завершается с сообщением `Error: UPGRADE FAILED: timed out waiting for the condition`.

Причина. Под или задача миграций не стали готовы за время ожидания Helm по одной из причин выше.

Решение. Проверьте поды и задачи релиза:

```
d8 k -n code get pods,jobs
```

Устраните причину и повторите `helm upgrade`; добавьте `--timeout`, если релизу нужно больше времени, чем по умолчанию.

Модуль Deckhouse Kubernetes Platform

Модуль Deckhouse Kubernetes Platform поставляет Deckhouse Code через компонент `code-operator`, который устанавливает и обслуживает инстанс Deckhouse Code. Каждая настройка инстанса — это поле ресурса CodeInstance; для пустого поля оператор берёт глобальную настройку платформы: шаблон публичных доменов, IngressClass или режим HTTPS. В собственном ModuleConfig модуля задаётся только параметр `logLevel`.

Веб-интерфейс модульной поставки доступен на поддомене `code` шаблона публичных доменов платформы, например `https://code.example.com`. Если в CodeInstance задан `spec.network.web.hostname`, используется указанный в нём адрес.

Требования

Для работы модуля необходимы:

- Deckhouse Kubernetes Platform версии 1.68 или новее;
- Kubernetes версии 1.29 или новее;
- включённый модуль `cert-manager`;
- PostgreSQL версии 17 или новее;
- Redis версии 7.0 или новее;
- S3-хранилище с провайдером `YCloud` или `Generic`;
- StorageClass для Git-репозиториев, указанный в поле `gitData.storageClass`;
- IngressClass для трафика к веб-интерфейсу.

Ресурсы кластера для размера инстанса приведены в разделе «Минимальные требования» документации модуля, а настройка PostgreSQL, Redis и S3-хранилища описана в разделе «Начало работы».

Установка

Включение модуля устанавливает `code-operator`; когда PostgreSQL, Redis и S3-хранилище готовы, применение ресурса CodeInstance запускает установку Deckhouse Code с заданными параметрами. Шаги установки — от включения модуля до первого входа — приведены в разделе «Начало работы» документации модуля.

> Отключение модуля удаляет конфигурацию Code, данные Git и все Secret и ConfigMap, принадлежащие оператору, включая ключи шифрования данных в PostgreSQL. Перед отключением модуля создайте резервную копию.

Настройка

Каждая настройка модульной поставки — это поле ресурса CodeInstance. Полный список полей приведён в разделе Custom Resources документации модуля.

Эксплуатация

Резервное копирование, масштабирование и настройка сети модульной поставки — это поля CodeInstance, которые описаны в документации модуля: «Обслуживание», «Масштабирование» и «Сеть». Обновление модуля выполняет платформа; резервная копия перед обновлением описана на странице «Обновление». Перенос инстанса между пакетом для ОС Linux или Omnibus Docker и модулем описан в разделе «Миграция».

Администрирование

Администратор модульной поставки использует раздел «Обзор» и остальные страницы раздела «Администрирование» для задач уровня инстанса: пользователи и доступ, аутентификация, безопасность, мониторинг, резервное копирование и обновление.

Администрирование

Обзор

Раздел «Администрирование» описывает настройки уровня инстанса Deckhouse Code: контроль доступа, аутентификацию, безопасность, мониторинг, режим обслуживания, резервное копирование и обновление.

Страницы, которые отличаются по типу установки

Deckhouse Code поставляется в виде пакета для ОС Linux, Omnibus Docker, Helm-чарта или модуля Deckhouse Kubernetes Platform. В разделе «Мониторинг и лимиты» встроенный мониторинг каждого типа установки вынесен на отдельную вкладку. «Резервное копирование и восстановление» и «Обновление» — единые страницы: сначала общая для всех типов установки часть, затем каждый шаг с различающимися командами во вкладках по типу установки. «Relay (KAS)» включается, получает адрес и проверяется по-разному в пакете для ОС Linux, в Omnibus Docker и в Helm-чарте, и каждый из этих шагов оформлен вкладками по типу установки.

Остальные страницы

- Лицензия — просмотр, установка и удаление лицензионного ключа инстанса.
- Пользователи и доступ — ограничение создания групп и проектов и управление доступом на уровне инстанса.
- Аутентификация — настройка входа через провайдеров OmniAuth и LDAP.
- Настройки безопасности — настройки безопасности инстанса, включая ограничения регистрации и двухфакторную аутентификацию администраторов.
- Ключи доступа — просмотр и отзыв токенов, ключей SSH и GPG, выданных в рамках инстанса.
- Сервисные аккаунты — создание сервисных аккаунтов для CI/CD-пайплайнов и интеграций.
- События аудита безопасности — просмотр событий аудита, зафиксированных для инстанса.
- Режим обслуживания — снижение количества операций записи на время регламентных работ.
- Расширенный поиск — управление полнотекстовой индексацией после подключения OpenSearch.
- Раннеры — образы раннера и примеры развёртывания с автомасштабированием на Deckhouse Virtualization Platform, Docker executor и Kubernetes executor.

Операции уровня кластера для модуля Deckhouse Kubernetes Platform — расчёт размера, настройка сети, миграция с пакета для ОС Linux или Omnibus Docker в модуль и обратно — описаны в документации модуля.

Миграция

Миграция с GitLab CE

Deckhouse Code работает на GitLab и принимает данные инстанса GitLab Community Edition (CE) той же версии GitLab: репозитории, пользователей, группы, задачи, запросы на слияние, настройки и историю CI/CD. Инстанс переносится целиком. Отдельные проекты между инстансами переносятся по инструкции «Импорт проекта».

Порядок миграции зависит от того, где работает Deckhouse Code:

- на том же сервере или с теми же томами Docker, что и GitLab CE: пакет или образ Deckhouse Code заменяет GitLab CE, данные остаются на месте, как описано в разделе «Миграция на месте»;
- на другом сервере, в установке из Helm-чарта или в модуле Deckhouse Kubernetes Platform: данные переносятся в архиве резервной копии, как описано в разделе «Миграция через архив резервной копии».

Перед началом

Проверьте условия на инстансе GitLab CE:

- Версия GitLab на инстансе совпадает с версией GitLab устанавливаемого выпуска Deckhouse Code. Архив резервной копии восстанавливается только в ту версию, в которой он создан. Если на инстансе более старая версия, сначала обновите его по пути обновления GitLab.
- Все фоновые миграции завершены: откройте «Admin» → «Мониторинг» → «Фоновые миграции» и проверьте, что вкладки «В очереди», «Завершаются» и «Неудачные» пусты.
- На диске достаточно свободного места для архива резервной копии.

Проверьте версию GitLab на инстансе:

Пакет для ОС Linux

```
sudo cat /opt/gitlab/embedded/service/gitlab-rails/VERSION
```

Omnibus Docker

В командах на этой странице контейнер GitLab CE называется `gitlab`; замените имя на своё.

```
docker exec gitlab cat /opt/gitlab/embedded/service/gitlab-rails/VERSION
```

Миграция на месте

Пакет Deckhouse Code заменяет пакет `gitlab-ce`, а образ Deckhouse Code запускается на томах контейнера GitLab CE. Конфигурация в `/etc/gitlab/gitlab.rb`, секреты в `/etc/gitlab/gitlab-secrets.json` и данные в `/var/opt/gitlab` сохраняются. Запуск настройки применяет миграции базы данных, которые создают таблицы Deckhouse Code.

Во время установки сервисы перезапускаются, и инстанс недоступен до окончания настройки и миграций.

Пакет для ОС Linux

1. Создайте резервную копию и скопируйте её за пределы сервера вместе с файлами `/etc/gitlab/gitlab.rb`, `/etc/gitlab/gitlab-secrets.json` и директорией `/etc/gitlab/ssl/`:

   ```
   sudo gitlab-backup create
   ```

2. Установите пакет Deckhouse Code. Менеджер пакетов удалит пакет `gitlab-ce`:

   ```
   # РЕД ОС.
   sudo dnf install -y ./deckhouse-code-<VERSION>.el8.x86_64.rpm
   # AlmaLinux.
   sudo dnf install -y ./deckhouse-code-<VERSION>.el9.x86_64.rpm
   # Ubuntu.
   sudo apt install -y ./deckhouse-code_<VERSION>_amd64.deb
   ```

3. Запустите настройку:

   ```
   sudo gitlab-ctl reconfigure
   ```

4. Проверьте, что не осталось неприменённых миграций, что зашифрованные данные в базе читаются секретами инстанса и что инстанс проходит самопроверку:

   ```
   sudo gitlab-rake db:migrate:status | grep -w down
   sudo gitlab-rake gitlab:fe:secret:verify
   sudo gitlab-rake gitlab:check SANITIZE=true
   ```

   Первая команда ничего не выводит, когда все миграции применены. Если она выводит миграции, примените их и повторите проверки:

   ```
   sudo gitlab-rake db:migrate
   ```

Omnibus Docker

В командах ниже `<GITLAB_HOME>` — директория на хосте с томами `config`, `logs` и `data` контейнера GitLab CE.

1. Создайте резервную копию и скопируйте её за пределы хоста вместе с директорией `<GITLAB_HOME>/config`:

   ```
   docker exec -t gitlab gitlab-backup create
   ```

2. Остановите и удалите контейнер GitLab CE. Тома — это директории на хосте, они остаются на месте:

   ```
   docker stop gitlab
   docker rm gitlab
   ```

3. Создайте контейнер Deckhouse Code на тех же томах, с портами и переменными окружения контейнера GitLab CE:

   ```
   docker run -d --name code \
     --shm-size 256m \
     -p 80:80 -p 443:443 -p 22:22 \
     -e GITLAB_OMNIBUS_CONFIG="external_url 'https://<HOSTNAME>'" \
     -v <GITLAB_HOME>/config:/etc/gitlab \
     -v <GITLAB_HOME>/logs:/var/log/gitlab \
     -v <GITLAB_HOME>/data:/var/opt/gitlab \
     <REGISTRY>/<FLAVOR>:<VERSION>
   ```

4. Следите за запуском. Миграции базы данных выполняются во время настройки, и до её окончания инстанс отвечает кодом 502:

   ```
   docker logs -f code
   ```

5. Проверьте, что не осталось неприменённых миграций, что зашифрованные данные в базе читаются секретами инстанса и что инстанс проходит самопроверку:

   ```
   docker exec code gitlab-rake db:migrate:status | grep -w down
   docker exec code gitlab-rake gitlab:fe:secret:verify
   docker exec code gitlab-rake gitlab:check SANITIZE=true
   ```

   Первая команда ничего не выводит, когда все миграции применены. Если она выводит миграции, примените их и повторите проверки:

```
docker exec code gitlab-rake db:migrate
```

Миграция через архив резервной копии

Данные переносятся в архиве резервной копии, а файлы, которых нет в архиве, копируются отдельно. Deckhouse Code устанавливается заранее, с версией GitLab исходного инстанса и тем же внешним адресом.

Сбор данных GitLab CE

Соберите файлы с инстанса GitLab CE и храните их с той же степенью защиты, что и пароли:

- архив резервной копии;
- `gitlab-secrets.json` — ключи шифрования базы данных; без них теряются настройки двухфакторной аутентификации, токены и зашифрованные переменные CI/CD;
- `gitlab.rb` — конфигурация инстанса;
- сертификаты и ключи из директории `/etc/gitlab/ssl/`;
- SSH-ключи хоста: с ними Git-клиенты подключаются к Deckhouse Code по SSH без предупреждения о смене ключа хоста.

Пакет для ОС Linux

1. Создайте резервную копию:

   ```
   sudo gitlab-backup create
   ```

   Архив `<BACKUP_NAME>_gitlab_backup.tar` записывается в `/var/opt/gitlab/backups/`.

2. Скопируйте за пределы сервера архив, файлы `/etc/gitlab/gitlab.rb` и `/etc/gitlab/gitlab-secrets.json`, директорию `/etc/gitlab/ssl/` и файлы `/etc/ssh/ssh_host_*`.

Omnibus Docker

1. Создайте резервную копию:

   ```
   docker exec -t gitlab gitlab-backup create
   ```

   Архив `<BACKUP_NAME>_gitlab_backup.tar` записывается в `<GITLAB_HOME>/data/backups/` на хосте.

2. Скопируйте за пределы хоста архив, файлы `gitlab.rb` и `gitlab-secrets.json`, директорию `ssl/` и файлы `ssh_host_*` из `<GITLAB_HOME>/config`.

Если GitLab CE хранит загруженные файлы, артефакты CI/CD, объекты LFS или пакеты в объектном хранилище, архив их не содержит. Подключите Deckhouse Code к тем же бакетам или скопируйте бакеты в хранилище Deckhouse Code, например командой `rclone sync`.

`<BACKUP_NAME>` — имя файла архива без суффикса `_gitlab_backup.tar`: для архива `1784800000_2026_07_23_19.4.1_gitlab_backup.tar` это `1784800000_2026_07_23_19.4.1`.

Восстановление в Deckhouse Code

После восстановления в базе данных схема GitLab CE. Команда `gitlab-rake db:migrate` добавляет в неё таблицы Deckhouse Code.

Пакет для ОС Linux

Установите Deckhouse Code, как описано в разделе «Быстрый старт».

1. Замените `/etc/gitlab/gitlab.rb` и `/etc/gitlab/gitlab-secrets.json` файлами GitLab CE, скопируйте сертификаты и ключи в `/etc/gitlab/ssl/` и запустите настройку:

   ```
   sudo gitlab-ctl reconfigure
   ```

2. Скопируйте SSH-ключи хоста в `/etc/ssh/` и перезапустите SSH-сервер:

   ```
   sudo cp ssh_host_* /etc/ssh/
   # РЕД ОС, AlmaLinux.
   sudo systemctl restart sshd
   # Ubuntu.
   sudo systemctl restart ssh
   ```

3. Скопируйте архив в директорию резервных копий и передайте файл пользователю `git`:

   ```
   sudo cp <BACKUP_NAME>_gitlab_backup.tar /var/opt/gitlab/backups/
   sudo chown git:git /var/opt/gitlab/backups/<BACKUP_NAME>_gitlab_backup.tar
   ```

4. Остановите сервисы, которые пишут в базу данных:

   ```
   sudo gitlab-ctl stop puma
   sudo gitlab-ctl stop sidekiq
   ```

5. Восстановите данные. Команда дважды запрашивает подтверждение: перед пересозданием файла `authorized_keys` и перед удалением текущих таблиц базы данных; оба раза ответьте `yes`:

   ```
   sudo gitlab-backup restore BACKUP=<BACKUP_NAME>
   ```

6. Примените миграции базы данных Deckhouse Code:

   ```
   sudo gitlab-rake db:migrate
   ```

7. Перезапустите инстанс и проверьте его:

   ```
   sudo gitlab-ctl restart
   sudo gitlab-rake gitlab:fe:secret:verify
   sudo gitlab-rake gitlab:check SANITIZE=true
   ```

Omnibus Docker

Контейнер читает конфигурацию, секреты и SSH-ключи хоста с тома конфигурации и создаёт ключи хоста, только когда их на томе нет, поэтому файлы помещаются на том до первого запуска.

1. Поместите файлы GitLab CE в директорию конфигурации на хосте: `gitlab.rb`, `gitlab-secrets.json` и файлы `ssh_host_*` — в `/srv/code/config/`, сертификаты и ключи — в `/srv/code/config/ssl/`.
2. Создайте контейнер, как описано в разделе «Быстрый старт», и дождитесь окончания первого запуска:

   ```
   docker logs -f code
   ```

3. Скопируйте архив в директорию резервных копий на томе данных и назначьте ему владельца, от имени которого работает инстанс:

   ```
   sudo cp <BACKUP_NAME>_gitlab_backup.tar /srv/code/data/backups/
   docker exec code chown git:git /var/opt/gitlab/backups/
   <BACKUP_NAME>_gitlab_backup.tar
   ```

4. Остановите сервисы, которые пишут в базу данных:

   ```
   docker exec code gitlab-ctl stop puma
   docker exec code gitlab-ctl stop sidekiq
   ```

5. Восстановите данные. Команда дважды запрашивает подтверждение: перед пересозданием файла `authorized_keys` и перед удалением текущих таблиц базы данных; оба раза ответьте `yes`:

   ```
   docker exec -it code gitlab-backup restore BACKUP=<BACKUP_NAME>
   ```

6. Примените миграции базы данных Deckhouse Code:

   ```
   docker exec code gitlab-rake db:migrate
   ```

7. Перезапустите контейнер и проверьте инстанс:

   ```
   docker restart code
   docker exec code gitlab-rake gitlab:fe:secret:verify
   docker exec code gitlab-rake gitlab:check SANITIZE=true
   ```

Helm-чарт

Установите Deckhouse Code, как описано в разделе «Быстрый старт»: релиз `code` в неймспейсе `code` с настроенным объектным хранилищем для резервных копий.

1. Преобразуйте раздел `gitlab_rails` файла `gitlab-secrets.json` в файл секретов Rails и замените им объект Secret релиза:

   ```
   yq -p json -o yaml '{"production": .gitlab_rails}' gitlab-secrets.json >
   secrets.yml
   d8 k -n code create secret generic code-rails-secret --from-file=secrets.yml
   \
     --dry-run=client -o yaml | d8 k apply -f -
   ```

2. Замените объект Secret с SSH-ключами хоста:

   ```
   d8 k -n code create secret generic code-gitlab-shell-host-keys --from-file=.
   \
     --dry-run=client -o yaml | d8 k apply -f -
   ```

   Выполните команду в директории, где лежат только файлы `ssh_host_*`.

3. Перезапустите поды, чтобы они прочитали новые объекты Secret:

   ```
   d8 k -n code delete pods -l app=webservice
   d8 k -n code delete pods -l app=sidekiq
   d8 k -n code delete pods -l app=gitlab-shell
   ```

4. Загрузите архив в бакет резервных копий — значение `global.appConfig.backups.bucket`.

5. Восстановите данные в поде toolbox:

   ```
   d8 k -n code exec -it deploy/code-toolbox -- backup-utility --restore -t
   <BACKUP_NAME>
   ```

6. Примените миграции базы данных Deckhouse Code:

   ```
   d8 k -n code exec -it deploy/code-toolbox -- gitlab-rake db:migrate
   ```

7. Перезапустите поды и проверьте инстанс:

```
d8 k -n code delete pods -l app=webservice
d8 k -n code delete pods -l app=sidekiq
d8 k -n code exec -it deploy/code-toolbox -- gitlab-rake
gitlab:fe:secret:verify
d8 k -n code exec -it deploy/code-toolbox -- gitlab-rake gitlab:check SANITIZ
E=true
```

Модуль Deckhouse Kubernetes Platform

Модуль берёт архив резервной копии из бакета резервных копий ресурса CodeInstance, а секреты и SSH-ключи хоста — из объектов Secret в неймспейсе `d8-code`. Порядок действий описан в разделе «Миграция» документации модуля; установка GitLab из пакета для ОС Linux, о которой там идёт речь, — это инстанс GitLab CE.

После миграции

Если Deckhouse Code работает на другом сервере, направьте DNS-имя инстанса на него, затем проверьте инстанс и завершите перенос:

1. Войдите в веб-интерфейс под учётной записью GitLab CE и клонируйте репозиторий по HTTPS и по SSH.
2. Установите лицензионный ключ, как описано в разделе «Лицензия». Без ключа инстанс работает в пробном режиме.
3. Если внешний адрес изменился, обновите параметр `url` каждого GitLab Runner в `/etc/gitlab-runner/config.toml` и перезапустите раннер командой `gitlab-runner restart`.

Конвейер сохраняет `CI_SERVER_URL`, `CI_PROJECT_URL` и другие переменные `CI_*` на момент своего создания. Если адрес изменился, повторно запущенное задание конвейера, созданного до миграции, использует старый адрес; запустите новый конвейер.

Переход между типами установки

Инстанс переходит на другой тип установки вместе с данными в архиве резервной копии. На целевой установке работает та версия Deckhouse Code, в которой создан архив.

Между пакетом для ОС Linux и Omnibus Docker

Пакет для ОС Linux и Omnibus Docker создают и восстанавливают одинаковый архив резервной копии командой `gitlab-backup`, и файлы конфигурации у них одинаковые.

1. Создайте резервную копию на исходном инстансе, как описано в разделе «Создание резервной копии».
2. Скопируйте с исходного инстанса архив, файлы `gitlab.rb` и `gitlab-secrets.json`, сертификаты и ключи из директории `ssl/` и SSH-ключи хоста. В пакете для ОС Linux SSH-

ключи хоста — это файлы `/etc/ssh/ssh_host_*`, в Omnibus Docker — файлы `ssh_host_*` на томе конфигурации.

3. Установите Deckhouse Code той же версии на целевой стороне: «Быстрый старт» пакета для ОС Linux или «Быстрый старт» Omnibus Docker. Для Omnibus Docker поместите файлы конфигурации и SSH-ключи хоста на том конфигурации до первого запуска, чтобы контейнер не создал новые ключи хоста.
4. Для пакета для ОС Linux поместите файлы конфигурации в `/etc/gitlab/`, а SSH-ключи хоста — в `/etc/ssh/`, затем выполните `sudo gitlab-ctl reconfigure`.
5. Восстановите архив, как описано в разделе «Восстановление».

Между пакетом для ОС Linux или Omnibus Docker и модулем

Перенос инстанса с пакета для ОС Linux или Omnibus Docker в модуль Deckhouse Kubernetes Platform и обратно описан в разделе «Миграция» документации модуля.

Лицензия

Лицензионный ключ, установленный на инстансе, задаёт его редакцию, Standard Edition (SE) или Enterprise Edition (EE), и вместе с ней набор возможностей. Администратор управляет ключом на странице «Admin» → «Лицензия».

Что определяет лицензия

Редакция, указанная в ключе, определяет доступность сканирования и отчётности по безопасности и аналитических отчётов: в EE они работают, в SE недоступны. Остальные возможности работают в обеих редакциях. Таблица по возможностям приведена на странице «Редакции».

Инстанс берёт редакцию из ключа так:

- ключ не установлен: инстанс работает в пробном режиме, доступны все возможности EE;
- в ключе указана редакция `ee`: инстанс работает как EE;
- в ключе указана редакция `se`, редакция не указана или указана редакция, неизвестная этой сборке продукта: инстанс работает как SE;
- ключ не проходит проверку или отозван: инстанс работает как SE, администраторы видят баннер.

Ключ выдаётся клиенту и подписывается поставщиком. Инстанс проверяет подпись ключами, встроенными в продукт; проверка выполняется на инстансе без обращения к сети. Ключ содержит имя и идентификатор клиента, номер лицензии, дату выдачи, тип лицензии и секцию Deckhouse Code с редакцией, количеством лицензированных мест, датой истечения, датой прекращения действия и датой, до которой доступны обновления продукта.

Установка и удаление ключа записываются в «События аудита безопасности» как `license_set` и `license_removed`.

Просмотр лицензии

Перейдите в «Admin» → «Лицензия». Страница открывается и из баннера лицензии по ссылке «Управление лицензией».

Без ключа страница показывает «На этом инстансе не установлена лицензия.». С ключом она показывает таблицу «Информация о лицензии»:

Строка	Содержимое
«Клиент», «Идентификатор клиента»	Клиент, которому выдан ключ
«Номер лицензии»	Идентификатор ключа; на него ссылаются события аудита и список отозванных лицензий
«Выдана»	Дата выдачи ключа
«Тип лицензии»	Тип лицензии, как он указан при выдаче
«Редакция»	Редакция из ключа, `se` или `ee`; «—», если редакция в ключе не указана
«Статус»	«Действует», «Истекла», «Отозвана» или «Недействительна»
«Действует до»	Дата истечения или «Бессрочная» для ключа без неё. За 30 дней до даты рядом с ней появляется пометка «Истекает», после даты — «Истекла»
«Обновления доступны до»	Дата, до которой ключ покрывает обновления продукта, или «Не ограничено». За 30 дней до даты рядом с ней появляется пометка «Обновления скоро станут недоступны», после даты — «Обновления недоступны». Дата не влияет на набор доступных возможностей
«Занято мест»	Число активных учётных записей и лицензированных мест, «N из M», при превышении лимита — пометка «Превышено»; для ключа без лимита мест — «N мест используется»

Даты показаны как календарные даты в UTC; при наведении на пометку «Истекает», «Истекла», «Обновления скоро станут недоступны» или «Обновления недоступны» показывается точное время.

Место занимает активная учётная запись пользователя: заблокированные пользователи, боты и сервисные аккаунты места не занимают. Число обновляется раз в 10 минут. При превышении лимита появляются пометка «Превышено» и баннер; набор возможностей от числа мест не зависит.

Статус «Отозвана» означает, что номер лицензии входит в список отозванных лицензий, встроенный в продукт; «Недействительна» — что поле секции Deckhouse Code в ключе имеет неожиданный тип. В обоих состояниях инстанс работает как SE. Список отозванных лицензий попадает на инстанс с обновлением продукта, поэтому ключ может стать отозванным после обновления.

Баннер лицензии

Администраторы видят баннер в верхней части каждой страницы, пока лицензия требует внимания; остальные пользователи его не видят. Баннер содержит ссылку «Управление лицензией» на страницу лицензии. Одновременно показывается один баннер, самый серьёзный:

Баннер	Состояние	Можно закрыть
«На этом инстансе не установлена лицензия.»	Ключ не установлен	Нет
«Установленный лицензионный ключ недействителен для этого инстанса.»	Сохранённый ключ не проходит проверку	Нет
«Лицензия отозвана. Обратитесь к поставщику.»	Номер лицензии входит в список отозванных лицензий	Нет
«Не удалось разобрать секцию Deckhouse Code в лицензии.»	Статус «Недействительна»	Нет
«Лицензия инстанса истекла (%{date}). Продлите её, чтобы сохранить лицензионное покрытие.»	Дата истечения прошла	Нет
«Активных пользователей (%{count}) больше, чем лицензированных мест (%{limit}).»	Активных учётных записей больше, чем лицензированных мест	Да
«Лицензия инстанса истекает (%{date}).»	За 30 дней до даты истечения	Да

Закрытый баннер скрыт 7 дней; он возвращается раньше, если ключ или его состояние изменились.

Установка ключа

1. Перейдите в «Admin» → «Лицензия».
2. Вставьте ключ в поле «Лицензионный ключ» под заголовком «Установить лицензию». Подсказка поля гласит «Вставьте лицензионный ключ, полученный от поставщика».
3. Нажмите «Установить лицензию».

Сообщение «Лицензионный ключ установлен.» подтверждает установку. Новый ключ заменяет сохранённый ранее и вступает в силу сразу. Ключ, который не прошёл проверку, не сохраняется, прежний ключ остаётся на месте.

Ключ — текст объёмом около 1 КБ из заглавных латинских букв A–Z и цифр 2–7. Переносы строк, пробелы, регистр букв и знаки `=` в конце, попавшие в текст при копировании, не учитываются.

Страница показывает причину, по которой ключ отклонён:

Сообщение	Причина
«Не удалось разобрать лицензионный ключ. Убедитесь, что он скопирован полностью.»	Текст не является лицензионным ключом, скопирован не полностью или изменён
«Подпись лицензионного ключа недействительна для этой инсталляции.»	Подпись не соответствует ключам поставщика, встроенным в эту сборку продукта
«Лицензия ещё не начала действовать — срок её действия начинается в будущем.»	Срок действия ключа начинается позже текущего времени инстанса
«Лицензия отозвана. Обратитесь к поставщику.»	Номер лицензии входит в список отозванных лицензий, встроенный в продукт
«Лицензионный ключ не содержит лицензии для Deckhouse Code.»	Ключ выдан для другого продукта Deckhouse
«Лицензионный ключ слишком длинный для сохранения.»	Вставленный текст длиннее 4096 байт

Удаление ключа

1. Перейдите в «Admin» → «Лицензия».
2. Нажмите «Удалить лицензию» и подтвердите вопрос «Вы уверены, что хотите удалить лицензионный ключ?».

Сообщение «Лицензионный ключ удалён.» подтверждает удаление. Инстанс возвращается в пробный режим со всеми возможностями, а администраторы видят баннер «На этом инстансе не установлена лицензия.», пока ключ не установлен.

Истечение срока действия

Ключ содержит дату истечения, показанную в строке «Действует до», и следующую за ней дату прекращения действия, которую страница не показывает. Какая из дат наступила, определяют часы инстанса.

- За 30 дней до даты истечения рядом с датой появляется пометка «Истекает», а администраторы видят баннер «Лицензия инстанса истекает (%{date}).»; баннер можно закрыть на 7 дней.
- С даты истечения статус меняется на «Истекла», а администраторы видят баннер «Лицензия инстанса истекла (%{date}). Продлите её, чтобы сохранить лицензионное покрытие.», который нельзя закрыть. Между датой истечения и датой прекращения действия лицензия находится в льготном периоде: набор возможностей не меняется.

- С даты прекращения действия инстанс с ключом EE работает как SE.

Ключ без даты прекращения действия сохраняет набор возможностей и после даты истечения; ключ без даты истечения («Бессрочная») не истекает. Истечение ключа SE меняет только статус и баннер.

Когда ключ EE достигает даты прекращения действия:

- политики сканирования перестают добавлять задачи сканеров в пайплайны, сканирования по расписанию останавливаются;
- страницы политик сканирования, списка зависимостей и соответствия лицензий отвечают 404, их пункты меню исчезают, а интеграции сканеров исчезают из списков интеграций и настроек;
- интеграция с DefectDojo перестаёт обрабатывать отчёты;
- страницы аналитических отчётов отвечают 404, их пункты меню исчезают, а поля GraphQL, на которых они построены, возвращают null.

Собранные данные остаются в хранилище: отчёты SBOM показываются снова, как только установлен действующий ключ EE, а страницы аналитики строятся по задачам, запросам на слияние и пайплайнам, которые остаются на инстансе. Продлённый ключ устанавливается как любой другой, и возможности возвращаются сразу.

Получение ключа

Ключи выдаёт поставщик. На странице «Лицензия» приведена строка для связи: «Нужна лицензия или есть вопрос? Свяжитесь с нами на сайте Deckhouse Code или напишите на contact@deckhouse.ru.»; ссылка ведёт на https://deckhouse.ru/products/code/.

Пользователи и доступ

Ограничение создания групп и проектов

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Видимость и контроль доступа» и в поле «Минимальная роль по умолчанию, необходимая для создания проектов» укажите минимальную роль, которой разрешено создавать проекты в инстансе.
3. Разверните раздел «Аккаунт и ограничения». В группе «Ограничения пользователя» снимите флажки «Разрешить новым пользователям создавать top level группы» и «Разрешить пользователям с ролью гостя создавать группы и личные проекты».
4. Нажмите «Сохранить изменения» в каждом изменённом разделе.

Флажок «Разрешить новым пользователям создавать top level группы» задаёт значение по умолчанию для учётных записей, созданных после изменения. Для уже существующей учётной записи разрешение задаётся в разделе «Admin» → «Обзор» → «Пользователи»: откройте

пользователя, нажмите «Редактировать» и задайте значение «Может создавать группы верхнего уровня».

Включённые протоколы доступа к Git

Инстанс принимает операции Git по SSH и по HTTP(S). Чтобы оставить один протокол:

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Видимость и контроль доступа».
3. В списке «Включенные протоколы доступа к Git» выберите «Only SSH» или «Only HTTP(S)».
4. Нажмите «Сохранить изменения».

Операции Git по оставшемуся за списком протоколу отклоняются.

Аутентификация по паролю для Git через HTTPS

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Ограничения входа».
3. Снимите флажок «Разрешить аутентификацию по паролю для Git через HTTP(S)».
4. Нажмите «Сохранить изменения».

После этого Git-клиент проходит аутентификацию по HTTPS с персональным токеном доступа вместо пароля учётной записи. При настроенном LDAP принимается также пароль LDAP.

Двухфакторная аутентификация для всего инстанса

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Ограничения входа».
3. Установите флажок «Требовать включения двухфакторной аутентификации для всех пользователей», чтобы второй фактор требовался от каждого пользователя инстанса, или флажок «Требовать двухфакторную аутентификацию для администраторов», чтобы он требовался только от администраторов.
4. В поле «Период отсрочки двухфакторной аутентификации» укажите, сколько часов пользователь может откладывать настройку. Значение `0` требует настройки при следующем входе.
5. Нажмите «Сохранить изменения».

Пользователю, который не настроил второй фактор, настройка предлагается при следующем входе. Доступные пользователю методы описаны на странице «Вход в систему».

Правила отправки изменений на уровне инстанса

Правила, заданные в разделе «Admin» → «Настройки» → «Репозиторий» → «Push-правила», действуют на все проекты инстанса: группа и проект получают их как значения по умолчанию и меняют их, только пока правило разрешает переопределение. Сами правила и порядок переопределения описаны на странице «Правила отправки изменений».

Аутентификация

Обзор

Deckhouse Code выполняет вход пользователей через внешних провайдеров аутентификации (OmniAuth) и берёт членство в группах и права доступа из каталога LDAP. Поведение провайдеров, правила синхронизации и решения о предоставлении и отзыве доступа одинаковы для всех типов установки; различается место, где записываются параметры:

- пакет для ОС Linux — файл `/etc/gitlab/gitlab.rb`;
- Omnibus Docker — переменная `GITLAB_OMNIBUS_CONFIG` или файл `/etc/gitlab/gitlab.rb` на томе конфигурации;
- Helm-чарт — значения чарта;
- модуль Deckhouse Kubernetes Platform — ресурс `CodeInstance`.

Ключи и манифесты для каждого типа приведены на странице «Настройка».

Параметры ниже названы так, как их читает продукт. Ключ, под которым записывается каждый из них, указан на странице вашего типа установки.

Ограничения входа и аутентификация по паролю

Вход локальных учётных записей настраивается в разделе «Admin» → «Настройки» → «Основные» → «Ограничения входа». Флажок «Разрешить аутентификацию по паролю через веб-интерфейс» показывает, входит ли учётная запись с паролем в веб-интерфейс наряду с внешними провайдерами; в интерфейсе флажок доступен только для чтения.

Чтобы включить аутентификацию по паролю для веб-интерфейса, откройте Rails-консоль:

Пакет для ОС Linux

```
sudo gitlab-rails console
```

Omnibus Docker

```
docker exec -it code gitlab-rails console
```

Helm-чарт

```
d8 k -n code exec -it deploy/code-toolbox -- gitlab-rails console -e production
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rails console -e
production
```

Выполните в ней команду:

```
Gitlab::CurrentSettings.update!(password_authentication_enabled_for_web: true)
```

Аутентификация по паролю для Git через HTTPS и двухфакторная аутентификация для всего инстанса задаются в том же разделе административной панели и описаны на странице «Пользователи и доступ».

Конфигурация OmniAuth

Общие параметры OmniAuth действуют для всех провайдеров; OpenID Connect и SAML добавляют собственные параметры.

Поддерживаемые провайдеры

В параметре `name` записи провайдера указывается одно из следующих значений:

- `openid_connect` — OpenID Connect (описан ниже);
- `saml` — SAML (описан ниже);
- `oauth2_generic` — произвольный провайдер OAuth 2.0;
- `jwt` — аутентификация по JWT;
- `github` — GitHub;
- `gitlab` — GitLab.com;
- `google_oauth2` — Google;
- `azure_activedirectory_v2` — Microsoft Entra ID (Azure AD);
- `atlassian_oauth2` — Atlassian;
- `crowd` — Atlassian Crowd;
- `auth0` — Auth0;
- `alicloud` — AliCloud;
- `salesforce` — Salesforce;
- `shibboleth` — Shibboleth.

Вход через LDAP настраивается отдельно, в секции LDAP-серверов (подробнее — в разделе «Синхронизация с LDAP»).

Общие параметры OmniAuth

Следующие параметры действуют для всех провайдеров:

- `enabled` — разрешает вход через внешних провайдеров. По умолчанию — `true`.
- `providers` — список провайдеров, через которых разрешён вход. По умолчанию — `[]`.
- `allow_single_sign_on` — список провайдеров, для которых при первом входе автоматически создаётся учётная запись (например, `['openid_connect']`). Также принимает значения `true` (все провайдеры) и `false`. Если автоматическое создание отключено, пользователь должен сначала получить учётную запись в Deckhouse Code, а затем связать её с провайдером. По умолчанию — `false`.
- `block_auto_created_users` — если `true`, автоматически созданные учётные записи блокируются до одобрения администратором. По умолчанию — `true`.
- `auto_link_ldap_user` — связывает учётную запись с учётной записью LDAP при первом входе (подробнее — в разделе «Связывание учётных записей OIDC с LDAP»). По умолчанию — `false`.
- `auto_link_user` — связывает вход через провайдера с существующей учётной записью Deckhouse Code по адресу электронной почты. Принимает список провайдеров или значения `true` и `false`. По умолчанию — `false`.
- `auto_sign_in_with_provider` — имя провайдера, на страницу входа которого пользователь перенаправляется автоматически, минуя страницу входа Deckhouse Code. По умолчанию — `false`.
- `external_providers` — список провайдеров, учётные записи которых создаются как внешние. По умолчанию — `[]`.
- `allow_bypass_two_factor` — список провайдеров, вход через которых не требует двухфакторной аутентификации. Также принимает значения `true` и `false`. По умолчанию — `false`.
- `sync_profile_from_provider` — список провайдеров, из данных которых обновляется профиль пользователя при каждом входе. Также принимает значения `true` и `false`. По умолчанию — `false`.
- `sync_profile_attributes` — список атрибутов профиля, которые обновляются при синхронизации: `name`, `email`, `location`. Синхронизируемые атрибуты становятся доступными только для чтения. По умолчанию — `['email']`.

OpenID Connect (OIDC)

Провайдеры перечисляются в параметре `providers`. Для провайдера OIDC доступны следующие параметры:

- `name` — тип провайдера. Для OIDC — всегда `'openid_connect'`.
- `label` — подпись кнопки входа. По умолчанию — `'Openid Connect'`.
- `icon` — адрес изображения, которое будет показано на кнопке входа.

- `args` — параметры подключения к провайдеру:
  - `name` — имя стратегии OmniAuth, совпадает со значением параметра `name` провайдера;
  - `scope` — список запрашиваемых областей доступа, например `['openid', 'profile', 'email']`;
  - `response_type` — тип ответа OAuth 2.0. Для потока Authorization Code — `'code'`;
  - `issuer` — адрес OIDC-провайдера;
  - `discovery` — если `true`, настройки провайдера запрашиваются автоматически по адресу `<issuer>/.well-known/openid-configuration`;
  - `client_auth_method` — способ аутентификации клиента на token endpoint: `'basic'` или `'query'`;
  - `uid_field` — поле из данных пользователя, которое используется как `uid` учётной записи (например, `preferred_username`). Если параметр не задан или поле отсутствует, используется поле `sub`;
  - `send_scope_to_token_endpoint` — передавать ли параметр `scope` в запросах к token endpoint. Задайте `false`, если провайдер не принимает этот параметр. По умолчанию — `true`;
  - `pkce` — включает Proof Key for Code Exchange (PKCE);
  - `client_options`:
    - `identifier` — идентификатор клиента, зарегистрированного у провайдера;
    - `secret` — секрет клиента;
    - `redirect_uri` — адрес установки Deckhouse Code с путём `/users/auth/openid_connect/callback`. Тот же адрес должен быть указан в настройках клиента на стороне провайдера.

Дополнительно Deckhouse Code поддерживает следующие параметры. Они указываются на верхнем уровне записи провайдера, рядом с параметром `name`:

- `allowed_groups` — список групп, пользователям которых разрешён вход. Пользователи вне этих групп будут заблокированы. По умолчанию — `null` (разрешены все группы).
- `admin_groups` — список групп, пользователи которых получают административные права. По умолчанию — `null` (права администратора не выдаются ни одной группе).
- `auditor_groups` — список групп, пользователи которых получают роль аудитора: доступ только на чтение ко всем группам и проектам, без доступа к административному разделу. По умолчанию — `null` (роль аудитора не выдаётся ни одной группе).
- `groups_attribute` — имя атрибута, из которого извлекаются группы пользователя. По умолчанию — `'groups'`.

Параметры `admin_groups` и `auditor_groups` учитываются, только если задан параметр `allowed_groups`. Если пользователь входит и в `admin_groups`, и в `auditor_groups`, ему назначаются права администратора.

Примеры записи провайдера для каждого типа установки: «Провайдер OpenID Connect» и «Провайдер SAML».

SAML

Для провайдеров SAML доступны аналогичные параметры:

- `allowed_groups` — список групп с разрешённым входом. По умолчанию — `null` (разрешены все группы).
- `admin_groups` — группы с административными правами. По умолчанию — `null` (права администратора не выдаются ни одной группе).
- `auditor_groups` — группы, пользователи которых получают роль аудитора: доступ только на чтение ко всем группам и проектам. По умолчанию — `null` (роль аудитора не выдаётся ни одной группе).
- `groups_attribute` — имя атрибута, содержащего группы. По умолчанию — `'Groups'`.

> Если пользователь входит в `admin_groups`, но не указан в `allowed_groups`, доступ будет запрещён. В этом случае административные права также не будут назначены.

Синхронизация с LDAP

Deckhouse Code поддерживает синхронизацию пользователей, групп и прав доступа с LDAP-сервером. Синхронизация выполняется автоматически раз в час либо с периодичностью, заданной расписанием задачи `ldap_sync_worker`.

Примеры записи LDAP-сервера для каждого типа установки: «LDAP-серверы» и «Расписание синхронизации».

Ограничения на стороне LDAP-сервера

Во время синхронизации выполняются LDAP-запросы ко всем пользователям и группам, указанным в конфигурации. При необходимости используется постраничная загрузка (pagination). Если на стороне LDAP установлены ограничения на число возвращаемых объектов, это может привести к ошибкам синхронизации или удалению прав доступа у пользователей.

Несколько LDAP-серверов

В секции LDAP можно описать несколько серверов. Ключ записи — имя сервера; из него формируется имя провайдера `ldap<ключ>` в нижнем регистре. Например, ключу `main` соответствует провайдер `ldapmain`. Это имя хранится в идентификаторе учётной записи и попадает в поле `server` записей о синхронизации в логах.

У каждого сервера собственные параметры подключения. Параметры синхронизации — секция `group_sync` со всем её содержимым, включая `role_mapping`, — учитываются только у сервера

под ключом `main`; у остальных серверов эта секция игнорируется (раздел «Область синхронизации»).

Вход работает через любой из описанных серверов: на странице входа и на странице входа в административный раздел появляется отдельная вкладка для каждого сервера. Отдельно включать это не требуется.

Серверы задействуются по-разному в зависимости от сценария:

Сценарий	Какие серверы используются
Вход через форму LDAP в веб-интерфейсе	Все серверы, отдельной вкладкой на каждый
Поиск учётной записи при связывании с OIDC (`auto_link_ldap_user`)	Все серверы, по очереди до первого совпадения
Синхронизация пользователей, групп и прав доступа	Только сервер `ldapmain`
Периодическая перепроверка доступа	`ldapmain`, если у учётной записи есть его идентификатор; иначе — каталоги её собственных идентификаторов

Область синхронизации

Синхронизация пользователей, групп и прав доступа охватывает только один сервер — тот, чьё имя провайдера равно `ldapmain`, то есть сервер под ключом `main`. Остальные серверы остаются источником входа, но не источником групп и прав. Ограничение действует одинаково и в задаче синхронизации по расписанию, и при синхронизации, которая выполняется при входе пользователя.

> ⚠ Имя `ldapmain` фиксировано и не настраивается. Если сервера под ключом `main` в секции нет, синхронизация не выполняется ни для одного из описанных серверов.

Следствия:

- пользователь, заведённый только в неглавном каталоге, входит в Deckhouse Code, но групп и прав из LDAP не получает — их назначают вручную;
- у пользователя с двумя идентификаторами (одинаковый основной адрес электронной почты в обоих каталогах) группы и права всегда приходят из `ldapmain`, независимо от того, через какой сервер он вошёл;
- секция `group_sync` неглавных серверов не читается, поэтому ошибка в их параметре `group_sync.filter` не мешает запуску приложения. Ошибка в фильтре сервера `main` по-прежнему обнаруживается сразу при старте и не даёт приложению подняться.

Дойдя до неглавного сервера, синхронизация пропускает его и пишет об этом одну запись уровня `INFO` с именем пропущенного сервера в поле `server`:

```
Server is not synchronized: users, groups and permissions come from 'ldapmain' only
```

Идентификация пользователей при нескольких серверах

Пользователь ищется сначала по идентификатору учётной записи, затем по основному адресу электронной почты:

- если основной адрес в двух каталогах совпадает, при входе через второй сервер к существующей учётной записи добавляется второй LDAP-идентификатор; новая учётная запись не создаётся;
- если адреса разные, будут созданы две разные учётные записи.

Список идентификаторов учётной записи доступен в административном разделе на странице `/admin/users/<username>/identities`.

Решение о доступе при нескольких идентификаторах

Deckhouse Code периодически, не чаще одного раза в час, перепроверяет доступ пользователя LDAP: обращается в каталог и блокирует учётную запись, если запись в каталоге не найдена, либо разблокирует, если она нашлась снова. Если у учётной записи несколько LDAP-идентификаторов, каталог для проверки выбирается так:

- есть идентификатор `ldapmain` — решает только он, остальные каталоги не опрашиваются. Пользователь есть в `ldapmain` и не отключён там — доступ есть; его там нет или он отключён — доступа нет, что бы ни происходило в остальных каталогах;
- идентификатора `ldapmain` нет, то есть пользователь заведён только в неглавных каталогах, — он проверяется по своим идентификаторам, и доступ сохраняется, пока запись найдена хотя бы в одном из соответствующих каталогов.

«Запись найдена» означает, что она существует в каталоге и не отключена средствами Active Directory. Второе проверяется только для каталогов, сконфигурированных как AD.

Правило согласовано с областью синхронизации: синхронизация блокирует пользователей по данным `ldapmain`, и перепроверка доступа блокирует ровно тех же. Поэтому вход через другой каталог не снимает блокировку, поставленную синхронизацией.

> ⚠ При переводе пользователя из каталога `ldapmain` в другой каталог удалите у его учётной записи устаревший идентификатор `ldapmain` на странице `/admin/users/<username>/identities`. Пока этот идентификатор на месте, пользователь остаётся заблокированным, даже если он заведён в другом каталоге. Автоматически идентификатор не удаляется.

Удаление или отключение пользователя в неглавном каталоге само по себе доступ не закрывает, пока пользователь остаётся в `ldapmain`. Отзывать доступ нужно в каталоге `ldapmain`.

Группы и права доступа

LDAP-группы сопоставляются с группами GitLab. При этом можно назначать роли пользователям на основе имени группы.

Обязательные параметры:

- `group_sync.base` — DN, с которого начинается поиск LDAP-групп.

Опциональные параметры:

- `group_sync.create_groups` — если `true`, группы будут создаваться в Deckhouse Code.
- `group_sync.filter` — LDAP-фильтр для поиска групп.
- `group_sync.scope` — область поиска групп (0 — Base, 1 — SingleLevel, 2 — WholeSubtree).
- `group_sync.prefix` — определяет, из какого атрибута брать имя родительской группы. Если атрибут отсутствует — используется значение по умолчанию.
- `group_sync.top_level_group` — имя группы верхнего уровня, в которую будут добавлены все синхронизированные группы.
- `group_sync.name_mask` — регулярное выражение для извлечения имени группы из атрибута CN (Common Name).
- `group_sync.owner` — имя пользователя, который будет добавлен как владелец группы (по умолчанию — `root`).

Секция role_mapping

Назначает права пользователям на основе имени группы (`cn`):

- `role_mapping.by_name` — регулярное выражение; если имя группы совпадает, пользователю назначается соответствующая роль.
- `role_mapping.gitlab_role` — имя роли, доступной на инстансе.

Список доступных ролей берётся из штатного справочника ролей — того же, по которому определяется, какие роли можно назначить участнику группы или проекта. Справочник читается заново при каждой синхронизации, поэтому роль, отключённая на уровне инстанса, в него не попадает, и указать её в `role_mapping` нельзя.

В настоящее время справочник содержит семь имён:

`gitlab_role`	Роль	Уровень доступа
`guest`	Guest	10
`planner`	Planner	15
`reporter`	Reporter	20
`security_manager`	Security Manager	25
`developer`	Developer	30
`maintainer`	Maintainer	40
`owner`	Owner	50

Имя роли записывается так, как указано в колонке `gitlab_role`: в нижнем регистре, слова разделяются подчёркиванием. Регистр и пробелы не нормализуются, поэтому, например, `Security Manager` считается нераспознанным именем.

Роль `security_manager` присутствует в справочнике, только если она включена на инстансе (переменная окружения `GITLAB_SECURITY_MANAGER_ROLE`, по умолчанию включена). Если роль отключена, её имя считается нераспознанным.

Если под имя LDAP-группы подошло несколько правил, назначается численно наибольший уровень доступа. Например, если группа подошла и под правило с ролью `security_manager` (уровень 25), и под правило с ролью `developer` (уровень 30), её участники получат роль Developer.

Проверка имён ролей

Имена ролей проверяются один раз, в начале назначения ролей пользователям по их LDAP-группам (далее — назначение членств), включая правила, под которые при текущей синхронизации не подошла ни одна LDAP-группа. Это позволяет обнаруживать и скрытые опечатки, которые проявились бы только при появлении подходящей группы.

Проверка не прерывает назначение членств:

- LDAP-группы, все подошедшие правила которых распознаны, обрабатываются полностью и получают членства;
- нераспознанное правило не участвует в вычислении уровня доступа;
- если под LDAP-группу подошли только нераспознанные правила, уровень доступа для неё не определяется, и группа целиком пропускается при текущей синхронизации: её текущие участники не пересчитываются и не удаляются.

При этом синхронизация завершается ошибкой — уже после того, как группы, пользователи и корректные членства записаны. Следствия:

- синхронизация учитывается как аварийно завершённая на странице метрик задачи синхронизации, а отметка об успешной синхронизации не обновляется (раздел «Ручной запуск синхронизации»);
- в логи попадает запись уровня `ERROR`, в которой перечислены и нераспознанные имена, и полный список допустимых:

```
Unknown gitlab_role in group_sync.role_mapping: 'security_manger'. Available
roles: guest, planner, reporter, security_manager, developer, maintainer, owner
```

Назначение членств выполняется и при входе пользователя через LDAP-провайдера. На этом пути та же ошибка только записывается в логи: вход отрабатывает штатно, остальные членства проставляются.

Определение членов группы

> ⚠ LDAP Sync не поддерживает транзитивность для вложенных групп. Подробная информация в разделе «Вложенные группы и транзитивность».

Deckhouse Code поддерживает следующие атрибуты для определения членов группы (все значения — массив DN):

- `member`;
- `uniquemember`;
- `memberof`;
- `memberuid`;
- `submember`.

Синхронизация пользователей

Во время синхронизации обновляются имена и email-адреса пользователей, а также статус блокировки.

Опциональные параметры:

- `sync_name` — если `true`, имя пользователя будет обновлено по данным LDAP.

Блокировка пользователей по данным LDAP

Если пользователь удалён из LDAP, очередная плановая синхронизация заблокирует его учётную запись. Если пользователя вернули в LDAP, следующая синхронизация разблокирует учётную запись автоматически.

Блокировка выполняется всегда и не требует дополнительных настроек: источником истины для статуса учётной записи остаётся LDAP. Вход заблокированного пользователя через OIDC-провайдера завершится отказом, даже если в самом провайдере пользователь активен.

Если в вашем каталоге заблокированных пользователей помечают атрибутом и оставляют в каталоге, исключите их из выдачи LDAP с помощью параметра `user_filter` сервера. Укажите один фильтр, соответствующий вашему каталогу:

Каталог	**Значение `user_filter`**
OpenLDAP с ppolicy	`(!(pwdAccountLockedTime=*))`
389-DS	`(!(nsAccountLock=TRUE))`
Active Directory	`(!(userAccountControl:1.2.840.113556.1.4.803:=2))`
Собственный атрибут	`(!(employeeType=blocked))`

Связывание учётных записей OIDC с LDAP

Если пользователи входят через OIDC-провайдера (например, Keycloak), а права выдаются по LDAP-группам, включите автоматическое связывание OIDC-учётной записи с учётной записью LDAP параметром `auto_link_ldap_user`.

Как ищется учётная запись LDAP

При первом входе через OIDC Deckhouse Code ищет пользователя в LDAP. Для поиска используются два значения из данных OIDC-провайдера:

- `uid` — значение поля, указанного в параметре `uid_field` провайдера (например, `preferred_username`);
- email — адрес электронной почты пользователя.

Настроенные LDAP-серверы опрашиваются по очереди. На каждом сервере выполняется до четырёх попыток поиска, до первого совпадения:

Искомое значение	**Атрибут для поиска**
`uid`	Атрибут, указанный в параметре `uid` LDAP-сервера (например, `cn`)
`uid`	Почтовые атрибуты: `mail`, `email`, `userPrincipalName`
email	Те же почтовые атрибуты
`uid`	DN — если значение `uid` само является DN

Если пользователь найден, к его учётной записи добавляется LDAP-идентификатор с найденным DN. Если ни одна попытка не дала результата, учётная запись создаётся без привязки к LDAP.

Таким образом, связывание сработает, только если `uid` или email из OIDC-провайдера совпадает со значением соответствующего атрибута в LDAP.

Первый и последующие входы

При первом успешном входе учётная запись связывается с LDAP. При последующих входах:

- LDAP не опрашивается — используется ранее установленная связь;
- доступ и административные права переоцениваются по группам из данных OIDC-провайдера (параметры `allowed_groups` и `admin_groups`). Если пользователя убрали из разрешённой группы, учётная запись будет заблокирована;
- членства в группах и проектах, а также роли в них при входе через OIDC не меняются — их обновляет фоновая синхронизация с LDAP по расписанию.

Поэтому сразу после первого входа пользователь может войти, но членств в группах и проектах у него ещё нет: они появятся после ближайшей синхронизации. Чтобы не ждать её, запустите синхронизацию вручную (подробнее — в разделе «Ручной запуск синхронизации»).

> Синхронизация групп и членств пользователя при входе выполняется только при входе через LDAP-провайдера (имя провайдера начинается с `ldap`). Вход через OIDC-провайдера её не запускает, даже если учётная запись уже связана с LDAP.

Вход пользователей, найденных в LDAP

Чтобы пользователи, найденные в LDAP, могли входить сразу, а остальные отправлялись на одобрение администратору, сочетайте два параметра:

- `block_auto_created_users` на уровне OmniAuth со значением `true` — пользователь, не найденный в LDAP, получает учётную запись, заблокированную до одобрения администратором;
- `block_auto_created_users` у LDAP-сервера `main` со значением `false` — пользователь, найденный в LDAP, входит сразу.

Особенности связывания

- Если пользователя блокируют переименованием `cn` в каталоге, при первом входе он всё равно может быть найден по email и связан с учётной записью LDAP. Для блокировки используйте удаление пользователя из каталога или параметр `user_filter` (подробнее — в разделе «Блокировка пользователей по данным LDAP»).
- Если пользователь не был связан с LDAP и его заблокировала задача `Ldap::BlockNonLdapUsersWorker`, автоматическая разблокировка не сработает. Такого пользователя нужно разблокировать вручную и связать с учётной записью LDAP.

Устранение проблем с синхронизацией

Если предыдущая задача синхронизации завершилась некорректно, Redis может сохранить блокировку на его выполнение (по умолчанию параметр `concurrency = 1`). Это помешает запуску новой задачи.

Чтобы снять блокировку:

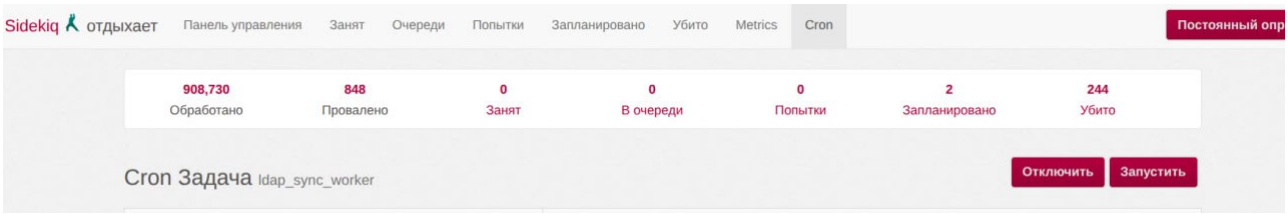
1. Подключитесь к Redis, используя базы, указанные в `config/redis.shared_state.yml` и `config/redis.queues.yml`.
2. Удалите ключ `sidekiq:concurrency_limit:throttled_jobs:{ldap/sync_worker}` следующими командами:

```
keys *ldap*
del "sidekiq:concurrency_limit:throttled_jobs:{ldap/sync_worker}"
```

Ручной запуск синхронизации

Чтобы синхронизировать группы сразу после их изменения на стороне LDAP, выполните следующие шаги:

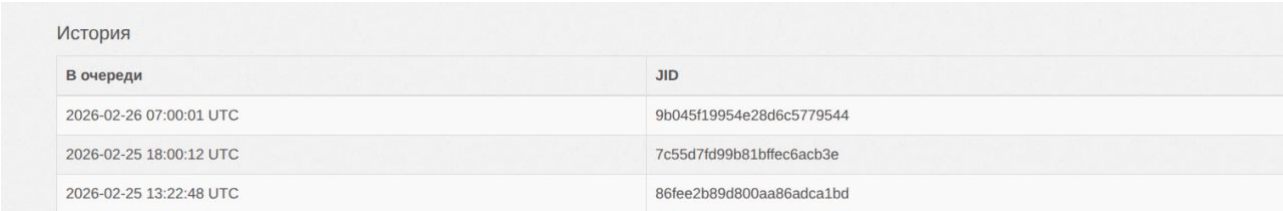
1. Зайдите на страницу worker'а синхронизации LDAP `/admin/sidekiq/cron/namespaces/default/jobs/ldap_sync_worker`.
2. В верхнем правом углу нажмите кнопку «Запустить» и подтвердите в диалоговом окне.



Чтобы посмотреть, как завершилась запущенная синхронизация, откройте страницу метрик задачи синхронизации LDAP: `/admin/sidekiq/metrics?substr=SyncWorker&period=8h`. На графике отображается статистика вызовов, в таблице ниже — количество успешно и аварийно завершённых вызовов синхронизации LDAP.

Чтобы посмотреть полные логи процесса синхронизации:

1. На странице worker’а `/admin/sidekiq/cron/namespaces/default/jobs/ldap_sync_worker` найдите таблицу событий запусков «История». Первая строка соответствует последнему запуску. Скопируйте значение в колонке JID (Job ID) — оно понадобится для поиска по логам.



2. Соберите записи логов Sidekiq для этого запуска, подставив скопированный JID:

Пакет для ОС Linux

```
sudo grep '<JID>' /var/log/gitlab/sidekiq/current
```

Omnibus Docker

```
docker exec code grep '<JID>' /var/log/gitlab/sidekiq/current
```

Helm-чарт

```
d8 k -n code logs -l app.kubernetes.io/component=sidekiq | jq 'select(.jid=="
<JID>")'
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code -l app.kubernetes.io/component=sidekiq get pod -o NAME
d8 k -n d8-code logs <POD_NAME> -c sidekiq | jq 'select(.jid=="<JID>")'
```

Со временем старые логи удаляются при ротации, и получить их будет нельзя. При необходимости запустите синхронизацию повторно и соберите актуальные логи.

Особенности работы LDAP Sync

Алгоритм синхронизации

LDAP Sync использует плоский алгоритм синхронизации:

1. Получение групп. Выполняется LDAP-запрос, который получает все группы по параметрам `base`, `filter` и `scope`.
2. Извлечение участников. Для каждой найденной группы читаются атрибуты членства: `member`, `uniquemember`, `memberof`, `memberuid`, `submember`.
3. Сопоставление пользователей. Каждый DN из атрибутов членства сопоставляется со значением `Identity.extern_uid` в базе данных.
4. Игнорирование неизвестных DN. Если DN не соответствует известному пользователю, он пропускается. Например, это может быть DN вложенной группы.

Циклические зависимости между группами

Циклические зависимости в иерархии LDAP-групп не приводят к ошибкам синхронизации. LDAP Sync обрабатывает группы плоско, по заданному фильтру, и не пытается воспроизводить древовидную структуру LDAP.

Рекурсивный обход вложенности не выполняется, поэтому циклы не влияют на результат синхронизации.

Вложенные группы и транзитивность

LDAP Sync не поддерживает транзитивность для вложенных групп.

Во время синхронизации LDAP Sync обрабатывает только прямые значения атрибутов членства группы и не выполняет рекурсивный обход вложенных групп.

Если одна LDAP-группа содержит другую как участника, пользователи вложенной группы не будут автоматически добавлены в родительскую группу.

Для корректной синхронизации все необходимые DN пользователей должны присутствовать непосредственно в атрибутах членства группы.

Если требуется учитывать вложенные группы, это нужно реализовать на стороне LDAP-сервера. Например, можно заполнять атрибут `submember` полным списком транзитивных участников.

Такой подход упрощает процесс синхронизации и исключает проблемы, связанные с рекурсивной обработкой групп.

Создание локальной учётной записи при включённой синхронизации с LDAP

Локальные учётные записи можно создавать и использовать даже при включённой синхронизации с LDAP.

Такие пользователи входят через веб-интерфейс, пока включена аутентификация по паролю для веб-интерфейса (раздел «Ограничения входа и аутентификация по паролю»).

Настройка

Назначение каждого параметра описано на странице «Обзор». Здесь для каждого типа установки указано место, в котором записываются параметры, общие параметры OmniAuth и способ применения изменений. Записи провайдеров и серверов приведены на страницах OpenID Connect, SAML и LDAP.

Где хранится конфигурация

Пакет для ОС Linux. Параметры задаются ключами файла `/etc/gitlab/gitlab.rb`.

Omnibus Docker. Контейнер читает те же ключи `gitlab.rb`. Они попадают в него из переменной окружения `GITLAB_OMNIBUS_CONFIG`, которая вычисляется при каждом запуске, и из файла `/etc/gitlab/gitlab.rb` на томе конфигурации, который читается после переменной. Ключ, записанный в файле, переопределяет тот же ключ в переменной. Переменная задаётся при создании контейнера:

```
docker run -d --name code \
  --shm-size 256m \
  -p 80:80 -p 443:443 -p 22:22 \
  -e GITLAB_OMNIBUS_CONFIG="external_url 'https://<HOSTNAME>';
gitlab_rails['omniauth_enabled'] = true; gitlab_rails['omniauth_allow_single_sign_on']
= ['openid_connect']; gitlab_rails['omniauth_auto_link_ldap_user'] = true" \
  -v /srv/code/config:/etc/gitlab \
  -v /srv/code/logs:/var/log/gitlab \
  -v /srv/code/data:/var/opt/gitlab \
  <REGISTRY>/<FLAVOR>:<VERSION>
```

Переменная фиксируется при создании контейнера, поэтому её изменение означает удаление контейнера и создание нового с теми же томами. Записи провайдеров и LDAP-серверов занимают несколько строк, а том конфигурации сохраняет их при замене контейнера. С томами из быстрого старта файл `/etc/gitlab/gitlab.rb` внутри контейнера — это `/srv/code/config/gitlab.rb` на хосте, и редактируется он на хосте:

```
sudo vi /srv/code/config/gitlab.rb
```

Helm-чарт. Параметры задаются значениями чарта в файле `values.yaml` релиза.

Модуль Deckhouse Kubernetes Platform. Параметры задаются в ресурсе `CodeInstance`, описанном на страницах «Настройка OmniAuth и LDAP» и Custom Resources в документации модуля.

Общие параметры OmniAuth

Параметры и их действие описаны в разделе «Общие параметры OmniAuth».

Пакет для ОС Linux

У каждого общего параметра OmniAuth собственный ключ в `gitlab.rb`:

Параметр	Ключ в `gitlab.rb`
`enabled`	`gitlab_rails['omniauth_enabled']`
`providers`	`gitlab_rails['omniauth_providers']`
`allow_single_sign_on`	`gitlab_rails['omniauth_allow_single_sign_on']`
`block_auto_created_users`	`gitlab_rails['omniauth_block_auto_created_users']`
`auto_link_ldap_user`	`gitlab_rails['omniauth_auto_link_ldap_user']`
`auto_link_user`	`gitlab_rails['omniauth_auto_link_user']`
`auto_sign_in_with_provider`	`gitlab_rails['omniauth_auto_sign_in_with_provider']`
`external_providers`	`gitlab_rails['omniauth_external_providers']`
`allow_bypass_two_factor`	`gitlab_rails['omniauth_allow_bypass_two_factor']`
`sync_profile_from_provider`	`gitlab_rails['omniauth_sync_profile_from_provider']`
`sync_profile_attributes`	`gitlab_rails['omniauth_sync_profile_attributes']`

```
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_allow_single_sign_on'] = ['openid_connect']
gitlab_rails['omniauth_block_auto_created_users'] = true
gitlab_rails['omniauth_auto_link_ldap_user'] = true
gitlab_rails['omniauth_sync_profile_attributes'] = ['email']
```

Omnibus Docker

Ключи и их значения те же, что во вкладке «Пакет для ОС Linux». Однострочный ключ помещается в `GITLAB_OMNIBUS_CONFIG` и отделяется от следующего символом `;`, как в команде `docker run` выше; ключ, значение которого занимает несколько строк, записывается в файл `/srv/code/config/gitlab.rb`.

Helm-чарт

Общие параметры — это значения в секции `global.appConfig.omniauth`:

```
global:
  appConfig:
    omniauth:
      enabled: true
      allowSingleSignOn: ['openid_connect']
      blockAutoCreatedUsers: true
      autoLinkLdapUser: true
      syncProfileAttributes: ['email']
```

Модуль Deckhouse Kubernetes Platform

Общие параметры задаются в секции `spec.appConfig.omniauth`, а провайдеры перечисляются в её параметре `providers`:

```
enabled: true
allowSingleSignOn: ['openid_connect']
blockAutoCreatedUsers: true
autoLinkLdapUser: true
syncProfileAttributes: ['email']
```

Применение конфигурации

Пакет для ОС Linux

Каждое изменение файла `/etc/gitlab/gitlab.rb` вступает в силу после выполнения команды:

```
sudo gitlab-ctl reconfigure
```

Omnibus Docker

Примените изменённый файл в работающем контейнере:

```
docker exec code gitlab-ctl reconfigure
```

Перезапуск контейнера тоже применяет конфигурацию; так вступает в силу изменённая переменная `GITLAB_OMNIBUS_CONFIG` пересозданного контейнера:

```
docker restart code
```

Helm-чарт

Примените изменённый файл `values.yaml` к релизу:

```
helm upgrade code deckhouse-code \
  -n code \
  -f values.yaml
```

Модуль Deckhouse Kubernetes Platform

Изменённый ресурс `CodeInstance` применяет оператор.

OpenID Connect

Назначение параметров провайдера описано на странице «Обзор»; место записи и применение изменений — на странице «Настройка».

Провайдер OpenID Connect

Параметры подключения провайдера и их действие описаны в разделе «OpenID Connect (OIDC)».

Пакет для ОС Linux

Провайдер — это запись массива `gitlab_rails['omniauth_providers']`. Ключи `allowed_groups`, `admin_groups`, `auditor_groups` и `groups_attribute` указываются рядом с `name`, на верхнем уровне записи; параметры подключения — внутри `args`:

```
gitlab_rails['omniauth_providers'] = [
  {
    name: 'openid_connect',   # Не изменяйте это значение.
    label: 'Keycloak',        # Подпись кнопки входа.
    allowed_groups: ['gitlab'],
    admin_groups: ['admin'],
    auditor_groups: ['audit'],
    groups_attribute: 'gitlab_group',
    args: {
      name: 'openid_connect',
      scope: ['openid', 'profile', 'email'],
      response_type: 'code',
      issuer: 'https://keycloak.example.com/realms/example',
      discovery: true,
      client_auth_method: 'query',
      uid_field: 'preferred_username',
      send_scope_to_token_endpoint: false,
      pkce: true,
      client_options: {
        identifier: '<client_id>',
        secret: '<client_secret>',
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
      }
    }
  }
]
```

Omnibus Docker

Запись та же, что во вкладке «Пакет для ОС Linux». Она занимает несколько строк, поэтому записывайте её в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Запись провайдера хранится в объекте Secret и подключается из значений, поэтому секрет клиента остаётся вне `values.yaml`. Создайте Secret с записью провайдера:

```
d8 k -n code create secret generic code-omniauth-keycloak \
  --from-file=provider=keycloak.yaml
```

Файл содержит ту же запись, что и при остальных типах установки:

```
name: 'openid_connect'
label: 'Keycloak'
allowed_groups:
  - 'gitlab'
admin_groups:
  - 'admin'
groups_attribute: 'gitlab_group'
args:
  name: 'openid_connect'
  scope:
    - 'openid'
    - 'profile'
    - 'email'
  response_type: 'code'
  issuer: 'https://keycloak.example.com/realms/example'
  discovery: true
  client_auth_method: 'query'
  uid_field: 'preferred_username'
  send_scope_to_token_endpoint: false
  pkce: true
  client_options:
    identifier: '<client_id>'
    secret: '<client_secret>'
    redirect_uri: 'https://code.example.com/users/auth/openid_connect/callback'
```

Подключите Secret в значениях:

```
global:
  appConfig:
    omniauth:
      providers:
        - secret: code-omniauth-keycloak
          key: provider
```

Модуль Deckhouse Kubernetes Platform

Провайдер — это запись параметра `providers` в секции `spec.appConfig.omniauth`. Ключи `allowedGroups`, `adminGroups` и `groupsAttribute` указываются рядом с `name`; параметра для групп аудиторов в ресурсе CodeInstance нет. Параметры подключения указываются внутри `args`:

```
providers:
  - name: 'openid_connect'   # Не изменяйте это значение.
    label: 'Keycloak'        # Подпись кнопки входа.
    allowedGroups:
      - 'gitlab'
    adminGroups:
      - 'admin'
    groupsAttribute: 'gitlab_group'
    args:
      name: 'openid_connect'
      scope:
        - 'openid'
        - 'profile'
        - 'email'
      response_type: 'code'
      issuer: 'https://keycloak.example.com/realms/example'
      discovery: true
      client_auth_method: 'query'
      uid_field: 'preferred_username'
      send_scope_to_token_endpoint: false
      pkce: true
      client_options:
        identifier: '<client_id>'
        secret: '<client_secret>'
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
```

Интеграция с Deckhouse Platform

В кластере Deckhouse Platform модуль `user-authn` запускает Dex — провайдер OpenID Connect с адресом `https://dex.<домен>/`. Deckhouse Code подключается к нему как клиент Dex, описанный в разделе «Настройка OAuth2-клиента в Dex для подключения приложения».

1. Создайте DexClient в любом неймспейсе кластера. В `allowedGroups` перечислите группы Dex, пользователям которых разрешён вход; без этого параметра войти может любой пользователь Dex:

```
apiVersion: deckhouse.io/v1
kind: DexClient
metadata:
  name: code
  namespace: code
spec:
  redirectURIs:
    - https://code.example.com/users/auth/openid_connect/callback
  allowedGroups:
    - Everyone
    - admins
```

Dex регистрирует клиента с идентификатором `dex-client-<name>@<namespace>` и сохраняет его секрет в объекте Secret `dex-client-<name>` того же неймспейса.

2. Получите секрет клиента:

```
d8 k -n code get secret dex-client-code -o jsonpath='{.data.clientSecret}' |
base64 -d
```

3. Добавьте запись провайдера. Значение `issuer` заканчивается на `/` и должно в точности совпадать с адресом, который публикует Dex, иначе вход завершается ошибкой.

Пакет для ОС Linux

```
gitlab_rails['omniauth_providers'] = [
  {
    name: 'openid_connect',
    label: 'Deckhouse',
    args: {
      name: 'openid_connect',
      scope: ['openid', 'profile', 'email'],
      response_type: 'code',
      issuer: 'https://dex.example.com/',
      discovery: true,
      client_auth_method: 'query',
      client_options: {
        identifier: 'dex-client-code@code',
        secret: '<clientSecret>',
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
      }
    }
  }
]
```

Omnibus Docker

Запись та же, что во вкладке «Пакет для ОС Linux»; она указывается в файле `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Файл `dex.yaml` объекта Secret `code-omniauth-dex`, который подключается в значениях так же, как в разделе «Провайдер OpenID Connect»:

```
name: 'openid_connect'
label: 'Deckhouse'
args:
  name: 'openid_connect'
  scope:
    - 'openid'
    - 'profile'
    - 'email'
  response_type: 'code'
  issuer: 'https://dex.example.com/'
  discovery: true
  client_auth_method: 'query'
  client_options:
    identifier: 'dex-client-code@code'
    secret: '<clientSecret>'
    redirect_uri: 'https://code.example.com/users/auth/openid_connect/callback'
```

Модуль Deckhouse Kubernetes Platform

Запись параметра `providers` в секции `spec.appConfig.omniauth`:

```
providers:
  - name: 'openid_connect'
    label: 'Deckhouse'
    args:
      name: 'openid_connect'
      scope:
        - 'openid'
        - 'profile'
        - 'email'
      response_type: 'code'
      issuer: 'https://dex.example.com/'
      discovery: true
      client_auth_method: 'query'
      client_options:
        identifier: 'dex-client-code@code'
        secret: '<clientSecret>'
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
```

SAML

Назначение параметров провайдера описано на странице «Обзор»; место записи и применение изменений — на странице «Настройка».

Провайдер SAML

Параметры провайдера SAML и их действие описаны в разделе «SAML».

Пакет для ОС Linux

Провайдер SAML записывается ещё одной записью массива `gitlab_rails['omniauth_providers']`:

```
gitlab_rails['omniauth_providers'] = [
  {
    name: 'saml',
    allowed_groups: ['gitlab'],
    admin_groups: ['admin'],
    groups_attribute: 'gitlab_group'
  }
]
```

Omnibus Docker

Запись та же, что во вкладке «Пакет для ОС Linux»; она записывается в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Запись SAML хранится в объекте Secret и подключается в `global.appConfig.omniauth.providers` так же, как запись OpenID Connect.

Модуль Deckhouse Kubernetes Platform

Провайдер — это запись параметра `providers` в секции `spec.appConfig.omniauth`:

```
providers:
  - name: 'saml'
    allowedGroups:
      - 'gitlab'
    adminGroups:
      - 'admin'
    groupsAttribute: 'gitlab_group'
```

LDAP

Назначение параметров LDAP описано на странице «Обзор»; место записи и применение изменений — на странице «Настройка».

LDAP-серверы

Ключ записи сервера является его именем. Назначение параметров подключения описано в разделе «Синхронизация с LDAP»; секция `group_sync`, которая создаёт группы и назначает роли, описана в разделах «Группы и права доступа» и «Секция role_mapping». Секция `group_sync` учитывается только у сервера под ключом `main` (раздел «Область синхронизации»).

Пакет для ОС Linux

Вход через LDAP включается ключом `gitlab_rails['ldap_enabled']`. Серверы описываются в ключе `gitlab_rails['ldap_servers']`, значение которого — документ YAML.

> ⚠ Значение разбирается как YAML, поэтому отступы внутри блока значимы, а символы табуляции его ломают.

```
gitlab_rails['ldap_enabled'] = true
gitlab_rails['ldap_servers'] = YAML.load <<-'EOS'
  main:
    label: 'Головной офис'
    host: ldap-main.example.com
    port: 3389
    uid: 'cn'
    bind_dn: 'uid=viewer,ou=People,dc=example,dc=com'
    password: 'viewer123'
    base: 'ou=People,dc=example,dc=com'
    # Не учитывать пользователей, заблокированных в каталоге (опционально).
    user_filter: '(!(nsAccountLock=TRUE))'
    # Пользователь найден в LDAP — вход разрешён сразу.
    block_auto_created_users: false
    sync_name: true
    group_sync:
      create_groups: true
      base: 'ou=Groups,dc=example,dc=com'
      filter: '(objectClass=groupOfNames)'
      prefix:
        attribute: 'businessCategory'
        default: 'default-program'
      top_level_group: 'LdapGroups'
      name_mask: '(?<=-)[A-z0-9]*$'
      owner: 'root'
      role_mapping:
        - by_name: '.*-project_manager-.*'
          gitlab_role: 'maintainer'
        - by_name: '.*-developer-.*'
          gitlab_role: 'developer'
        - by_name: '.*-participant-.*'
          gitlab_role: 'reporter'
  contractors:
    label: 'Подрядчики'
    host: ldap-contractors.example.com
    port: 3389
    uid: 'cn'
    bind_dn: 'uid=viewer,ou=People,dc=contractors,dc=example,dc=com'
    password: 'viewer123'
    base: 'ou=People,dc=contractors,dc=example,dc=com'
EOS
```

Omnibus Docker

Ключи `gitlab_rails['ldap_enabled']` и `gitlab_rails['ldap_servers']` те же, что во вкладке «Пакет для ОС Linux». Значение ключа `gitlab_rails['ldap_servers']` занимает несколько строк, поэтому записывайте его в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Серверы описываются значениями в секции `global.appConfig.ldap.servers` теми же ключами, которые читает продукт. Пароль сервисной учётной записи берётся из объекта Secret:

```
d8 k -n code create secret generic code-ldap-password \
  --from-literal=password='viewer123'
```

```
global:
  appConfig:
    ldap:
      servers:
        main:
          label: 'Головной офис'
          host: ldap-main.example.com
          port: 3389
          uid: 'cn'
          bind_dn: 'uid=viewer,ou=People,dc=example,dc=com'
          password:
            secret: code-ldap-password
            key: password
          base: 'ou=People,dc=example,dc=com'
          user_filter: '(!(nsAccountLock=TRUE))'
          block_auto_created_users: false
          sync_name: true
          group_sync:
            create_groups: true
            base: 'ou=Groups,dc=example,dc=com'
            filter: '(objectClass=groupOfNames)'
            top_level_group: 'LdapGroups'
            owner: 'root'
            role_mapping:
              - by_name: '.*-developer-.*'
                gitlab_role: 'developer'
        contractors:
          label: 'Подрядчики'
          host: ldap-contractors.example.com
          port: 3389
          uid: 'cn'
          bind_dn: 'uid=viewer,ou=People,dc=contractors,dc=example,dc=com'
          password:
            secret: code-ldap-password
            key: password
          base: 'ou=People,dc=contractors,dc=example,dc=com'
```

Модуль Deckhouse Kubernetes Platform

Серверы — это ключи параметра `servers` в секции `spec.appConfig.ldap`:

```
servers:
  main:
    label: ldap
    host: 127.0.0.1
    port: 3389
    bindDn: 'uid=viewer,ou=People,dc=example,dc=com'
    base: 'ou=People,dc=example,dc=com'
    uid: 'cn'
    password: 'viewer123'
    syncName: true
    groupSync:
      createGroups: true
      base: 'ou=Groups,dc=example,dc=com'
      filter: '(objectClass=groupOfNames)'
      prefix:
        attribute: 'businessCategory'
        default: 'default-program'
      topLevelGroup: 'LdapGroups'
      nameMask: '(?<=-)[A-z0-9]*$'
      owner: 'root'
      roleMapping:
        - byName: '.*-project_manager-.*'
          gitlabRole: 'Maintainer'
        - byName: '.*-developer-.*'
          gitlabRole: 'Developer'
        - byName: '.*-participant-.*'
          gitlabRole: 'Reporter'
```

Несколько LDAP-серверов

Несколько серверов описываются повторением записи сервера под другим ключом. У каждого сервера собственные параметры подключения; секция `group_sync` со всем её содержимым учитывается только у сервера под ключом `main` (раздел «Область синхронизации»). Идентификация пользователя и решение о доступе, когда один и тот же пользователь найден на нескольких серверах, описаны в разделе «Несколько LDAP-серверов».

Пакет для ОС Linux

Записи являются ключами того же документа YAML в `gitlab_rails['ldap_servers']`, как `main` и `contractors` в разделе «LDAP-серверы» выше.

Omnibus Docker

Записи те же, что во вкладке «Пакет для ОС Linux»; они записываются в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Записи являются ключами секции `global.appConfig.ldap.servers`, как `main` и `contractors` в разделе «LDAP-серверы» выше.

Модуль Deckhouse Kubernetes Platform

Записи являются ключами параметра `servers` секции `spec.appConfig.ldap`:

```
servers:
  main:
    label: 'Головной офис'
    host: ldap-main.example.com
    base: 'ou=People,dc=example,dc=com'
    uid: 'cn'
    # Остальные параметры подключения.
    groupSync:
      base: 'ou=Groups,dc=example,dc=com'
      roleMapping:
        - byName: '.*-developer-.*'
          gitlabRole: 'Developer'
  contractors:
    label: 'Подрядчики'
    host: ldap-contractors.example.com
    base: 'ou=People,dc=contractors,dc=example,dc=com'
    uid: 'cn'
    # Остальные параметры подключения.
```

Связывание учётных записей OIDC с LDAP

Автоматическое связывание OIDC-учётной записи с учётной записью LDAP включается параметром `auto_link_ldap_user`. Как ищется учётная запись LDAP и что происходит при первом и последующих входах, описано в разделе «Связывание учётных записей OIDC с LDAP».

Пакет для ОС Linux

```
gitlab_rails['omniauth_auto_link_ldap_user'] = true
```

Omnibus Docker

Ключ тот же, что во вкладке «Пакет для ОС Linux». Его значение занимает одну строку, поэтому помещается и в `GITLAB_OMNIBUS_CONFIG`, и в файл `/srv/code/config/gitlab.rb`.

Helm-чарт

```
global:
  appConfig:
    omniauth:
      autoLinkLdapUser: true
```

Модуль Deckhouse Kubernetes Platform

Параметр задаётся в секции `spec.appConfig.omniauth`:

```
autoLinkLdapUser: true
```

Вход пользователей, найденных в LDAP

Чтобы пользователи, найденные в LDAP, могли входить сразу, а остальные отправлялись на одобрение администратору, сочетайте параметры `auto_link_ldap_user` и `block_auto_created_users` секции OmniAuth с параметром `block_auto_created_users` LDAP-сервера `main` (раздел «Вход пользователей, найденных в LDAP»).

Пакет для ОС Linux

```
gitlab_rails['omniauth_auto_link_ldap_user'] = true
# Пользователь не найден в LDAP — учётная запись создаётся заблокированной,
# до одобрения администратором.
gitlab_rails['omniauth_block_auto_created_users'] = true
gitlab_rails['ldap_servers'] = YAML.load <<-'EOS'
  main:
    # Пользователь найден в LDAP — вход разрешён сразу.
    block_auto_created_users: false
    # Остальные параметры подключения.
EOS
```

Omnibus Docker

Ключи те же, что во вкладке «Пакет для ОС Linux»; они записываются в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

```
global:
  appConfig:
    omniauth:
      autoLinkLdapUser: true
      # Пользователь не найден в LDAP — учётная запись создаётся заблокированной,
      # до одобрения администратором.
      blockAutoCreatedUsers: true
    ldap:
      servers:
        main:
          # Пользователь найден в LDAP — вход разрешён сразу.
          block_auto_created_users: false
```

Модуль Deckhouse Kubernetes Platform

Параметры задаются в секции `spec.appConfig`:

```
omniauth:
  autoLinkLdapUser: true
  # Пользователь не найден в LDAP — учётная запись создаётся заблокированной,
  # до одобрения администратором.
  blockAutoCreatedUsers: true
ldap:
  servers:
    main:
      # Пользователь найден в LDAP — вход разрешён сразу.
      blockAutoCreatedUsers: false
```

Расписание синхронизации

Расписание задачи синхронизации записывается в формате cron. Что задача делает при каждом запуске, описано в разделе «Синхронизация с LDAP»; запуск вручную описан в разделе «Ручной запуск синхронизации».

Пакет для ОС Linux

```
gitlab_rails['ldap_sync_worker_cron'] = "0 * * * *"
```

Omnibus Docker

Ключ тот же, что во вкладке «Пакет для ОС Linux». Его значение занимает одну строку, поэтому помещается и в `GITLAB_OMNIBUS_CONFIG`, и в файл `/srv/code/config/gitlab.rb`.

Helm-чарт

```
global:
  appConfig:
    cron_jobs:
      ldap_sync_worker:
        cron: "0 * * * *"
```

Модуль Deckhouse Kubernetes Platform

Периодичность задаётся параметром `cronJobs` секции `spec.appConfig`:

```
cronJobs:
  ldapSyncWorker:
    cron: "0 * * * *"
```

Пример настройки: вход через OIDC и права из LDAP

Ниже приведена последовательность настройки, при которой пользователи входят через OIDC-провайдера (например, Keycloak), а группы, членства в них и роли приходят из LDAP.

Настройка включает в себя следующие шаги:

1. Настройка LDAP-провайдера.
2. Настройка OIDC-провайдера и включение связывания с LDAP.
3. Проверка связывания при первом входе.
4. Синхронизация прав.

Требования для настройки

Для настройки требуются:

- OIDC-провайдер.
- LDAP-каталог с теми же пользователями и с группами, имена которых позволяют определить роль.
- Сервисная учётная запись LDAP с правами на чтение каталога (параметры `bind_dn` и `password`).

> Значение `uid` или email в OIDC-провайдере должно совпадать со значением соответствующего атрибута в LDAP, иначе связывание не сработает.

Порядок поиска учётной записи описан в разделе «Как ищется учётная запись LDAP».

Настройка LDAP-провайдера

В записи сервера указываются каталог, база поиска и сервисная учётная запись, а её секция `group_sync` превращает имена групп в роли.

Пакет для ОС Linux

```
gitlab_rails['ldap_enabled'] = true
gitlab_rails['ldap_servers'] = YAML.load <<-'EOS'
  main:
    label: ldap
    host: ldap.example.com
    port: 3389
    bind_dn: 'uid=viewer,ou=People,dc=example,dc=com'
    password: 'viewer123'
    base: 'ou=People,dc=example,dc=com'
    uid: 'cn'
    sync_name: true
    # Не учитывать пользователей, заблокированных в каталоге (опционально).
    user_filter: '(!(nsAccountLock=TRUE))'
    # Пользователь найден в LDAP — вход разрешён сразу.
    block_auto_created_users: false
    group_sync:
      create_groups: true
      base: 'ou=Groups,dc=example,dc=com'
      filter: '(objectClass=groupOfNames)'
      top_level_group: 'LdapGroups'
      name_mask: '(?<=-)[A-z0-9]*$'
      owner: 'root'
      role_mapping:
        - by_name: '.*-maintainer-.*'
          gitlab_role: 'maintainer'
        - by_name: '.*-developer-.*'
          gitlab_role: 'developer'
        - by_name: '.*-participant-.*'
          gitlab_role: 'reporter'
EOS
```

Omnibus Docker

Ключи `gitlab_rails['ldap_enabled']` и `gitlab_rails['ldap_servers']` те же, что во вкладке «Пакет для ОС Linux»; они записываются в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

```
global:
  appConfig:
    ldap:
      servers:
        main:
          label: ldap
          host: ldap.example.com
          port: 3389
          bind_dn: 'uid=viewer,ou=People,dc=example,dc=com'
          password:
            secret: code-ldap-password
            key: password
          base: 'ou=People,dc=example,dc=com'
          uid: 'cn'
          sync_name: true
          # Не учитывать пользователей, заблокированных в каталоге (опционально).
          user_filter: '(!(nsAccountLock=TRUE))'
          # Пользователь найден в LDAP — вход разрешён сразу.
          block_auto_created_users: false
          group_sync:
            create_groups: true
            base: 'ou=Groups,dc=example,dc=com'
            filter: '(objectClass=groupOfNames)'
            top_level_group: 'LdapGroups'
            name_mask: '(?<=-)[A-z0-9]*$'
            owner: 'root'
            role_mapping:
              - by_name: '.*-maintainer-.*'
                gitlab_role: 'maintainer'
              - by_name: '.*-developer-.*'
                gitlab_role: 'developer'
              - by_name: '.*-participant-.*'
                gitlab_role: 'reporter'
```

Модуль Deckhouse Kubernetes Platform

Настройте конфигурацию в параметре `servers` секции `spec.appConfig.ldap`:

```
servers:
  main:
    label: ldap
    host: ldap.example.com
    port: 3389
    bindDn: 'uid=viewer,ou=People,dc=example,dc=com'
    password: 'viewer123'
    base: 'ou=People,dc=example,dc=com'
    uid: 'cn'
    syncName: true
    # Не учитывать пользователей, заблокированных в каталоге (опционально).
    userFilter: '(!(nsAccountLock=TRUE))'
    # Пользователь найден в LDAP — вход разрешён сразу.
    blockAutoCreatedUsers: false
    groupSync:
      createGroups: true
      base: 'ou=Groups,dc=example,dc=com'
      filter: '(objectClass=groupOfNames)'
      topLevelGroup: 'LdapGroups'
      nameMask: '(?<=-)[A-z0-9]*$'
      owner: 'root'
      roleMapping:
        - byName: '.*-maintainer-.*'
          gitlabRole: 'Maintainer'
        - byName: '.*-developer-.*'
          gitlabRole: 'Developer'
        - byName: '.*-participant-.*'
          gitlabRole: 'Reporter'
```

Настройка OIDC-провайдера и включение связывания с LDAP

Значение `uid_field` используется при поиске пользователя в LDAP. Оно должно совпадать со значением атрибута, указанного в параметре `uid` LDAP-сервера, или с адресом электронной почты — подробнее в разделе «Как ищется учётная запись LDAP».

Остальные параметры провайдера описаны в разделе «OpenID Connect (OIDC)».

Пакет для ОС Linux

```
gitlab_rails['omniauth_enabled'] = true
gitlab_rails['omniauth_auto_link_ldap_user'] = true
gitlab_rails['omniauth_providers'] = [
  {
    name: 'openid_connect',   # Не изменяйте это значение.
    label: 'Keycloak',        # Подпись кнопки входа.
    groups_attribute: 'gitlab_group',
    args: {
      name: 'openid_connect',
      scope: ['openid', 'profile', 'email'],
      response_type: 'code',
      issuer: 'https://keycloak.example.com/realms/example',
      discovery: true,
      client_auth_method: 'query',
      uid_field: 'preferred_username',
      send_scope_to_token_endpoint: false,
      pkce: true,
      client_options: {
        identifier: '<client_id>',
        secret: '<client_secret>',
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
      }
    }
  }
]
```

Omnibus Docker

Ключи `gitlab_rails['omniauth_enabled']`, `gitlab_rails['omniauth_auto_link_ldap_user']` и `gitlab_rails['omniauth_providers']` те же, что во вкладке «Пакет для ОС Linux»; они записываются в файл `/srv/code/config/gitlab.rb` на томе конфигурации.

Helm-чарт

Запись провайдера из этого примера хранится в объекте Secret и подключается из значений, рядом с которыми включается связывание:

```
global:
  appConfig:
    omniauth:
      enabled: true
      autoLinkLdapUser: true
      providers:
        - secret: code-omniauth-keycloak
          key: provider
```

Модуль Deckhouse Kubernetes Platform

Настройте конфигурацию в секции `spec.appConfig.omniauth`:

```
autoLinkLdapUser: true
providers:
  - name: 'openid_connect'   # Не изменяйте это значение.
    label: 'Keycloak'        # Подпись кнопки входа.
    groupsAttribute: 'gitlab_group'
    args:
      name: 'openid_connect'
      scope:
        - 'openid'
        - 'profile'
        - 'email'
      response_type: 'code'
      issuer: 'https://keycloak.example.com/realms/example'
      discovery: true
      client_auth_method: 'query'
      uid_field: 'preferred_username'
      send_scope_to_token_endpoint: false
      pkce: true
      client_options:
        identifier: '<client_id>'
        secret: '<client_secret>'
        redirect_uri: 'https://code.example.com/users/auth/openid_connect/
callback'
```

Проверка связывания при первом входе

Для проверки связывания выполните следующие шаги:

1. Войдите под тестовым пользователем через OIDC-провайдера.
2. Откройте страницу пользователя в административном разделе (`/admin/users/<username>/identities`). У связанной учётной записи должно быть два идентификатора: `openid_connect` и `ldapmain` (имя LDAP-идентификатора складывается из префикса `ldap` и имени LDAP-сервера, в примере — `main`).

Если LDAP-идентификатора нет, проверьте следующее:

- значения `uid` или email в OIDC-провайдере и в LDAP совпадают;
- пользователь попадает в область поиска `base` и не отсекается фильтром `user_filter` ;
- параметр `auto_link_ldap_user` включён.

Синхронизация прав

Сразу после первого входа у пользователя есть учётная запись, но нет членства в группах и проектах. Дождитесь ближайшей синхронизации или запустите её вручную (раздел «Ручной запуск синхронизации»), после чего проверьте, что:

- группы созданы внутри группы, указанной в `group_sync.top_level_group` ;
- пользователь добавлен в них с ролью, соответствующей `role_mapping` .

Настройки безопасности

Учётная запись администратора

Инстанс запускается с учётной записью администратора `root` , пароль которой создаётся при установке.

1. Войдите под `root` с первоначальным паролем.
2. Откройте меню пользователя и выберите «Редактировать профиль».
3. Перейдите в «Доступ» → «Пароль и аутентификация» и задайте новый пароль.
4. Сохраните новый пароль вне инстанса.

После смены пароля удалите первоначальный пароль оттуда, где его хранит поставка:

Пакет для ОС Linux

Файл `/etc/gitlab/initial_root_password` , удаляется командой `sudo rm -f /etc/gitlab/initial_root_password` . Команда `gitlab-ctl reconfigure` , выполненная больше чем через 24 часа после установки, удаляет файл сама, значение пароля при этом не меняется.

Omnibus Docker

Файл `/etc/gitlab/initial_root_password` внутри контейнера, удаляется командой `docker exec code rm -f /etc/gitlab/initial_root_password`. Файл лежит в томе с конфигурацией: с томами из раздела «Быстрый старт» это `/srv/code/config/initial_root_password` на хосте.

Helm-чарт

Секрет `code-initial-root-password` в неймспейсе релиза, читается командой `d8 k -n code get secret code-initial-root-password -o jsonpath='{.data.password}' | base64 -d`. После смены пароля значение из секрета больше не открывает учётную запись.

Модуль Deckhouse Kubernetes Platform

Секрет `initial-root-password` в неймспейсе `d8-code`, читается командой `d8 k -n d8-code get secret initial-root-password -o jsonpath='{.data.password}' | base64 -d`. После смены пароля значение из секрета больше не открывает учётную запись.

Регистрация и вход

Новые учётные записи

После установки пакета для ОС Linux, Omnibus Docker или Helm-чарта любой, у кого есть сетевой доступ к инстансу, заводит себе учётную запись. Чтобы закрыть регистрацию:

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Ограничения аккаунтов новых пользователей».
3. Снимите флажок «Разрешить новые учётные записи пользователей».
4. Нажмите «Сохранить изменения».

В модуле Deckhouse Kubernetes Platform регистрация закрыта после установки; оператор задаёт её из поля `appConfig.signUpEnabled` ресурса CodeInstance.

В остальных типах установки то же самое делается из консоли:

Пакет для ОС Linux

```
sudo gitlab-rails runner 'ApplicationSetting.current.update!(signup_enabled:
false)'
sudo gitlab-rails runner 'puts ApplicationSetting.current.signup_enabled'
```

Omnibus Docker

```
docker exec code gitlab-rails runner 'ApplicationSetting.current.update!
(signup_enabled: false)'
docker exec code gitlab-rails runner 'puts
ApplicationSetting.current.signup_enabled'
```

Helm-чарт

```
d8 k -n code exec deploy/code-toolbox -- gitlab-rails runner 'ApplicationSetting.c
urrent.update!(signup_enabled: false)'
d8 k -n code exec deploy/code-toolbox -- gitlab-rails runner 'puts
ApplicationSetting.current.signup_enabled'
```

Вторая команда выводит `false`, пока регистрация закрыта.

Пока регистрация открыта, тот же раздел административной панели ограничивает круг тех, кто получает учётную запись:

- «Требовать одобрение администратора для новых учётных записей» — учётная запись ждёт администратора до первого входа;
- «Настройки подтверждения электронной почты» — в режиме «Жесткий режим» адрес подтверждается до первого входа;
- «Разрешённые домены для новых пользователей» и «Список запрещённых доменов» — почтовые домены, с которыми регистрация разрешена и запрещена;
- «Ограничения электронной почты» → «Включить ограничения электронной почты для новых пользователей» — с установленным флажком адрес новой учётной записи проверяется по регулярному выражению из поля «Ограничения электронной почты для новых пользователей», несовпадающий адрес отклоняется;
- «Минимальная длина пароля (количество символов)» — действует для учётных записей, которые входят по паролю, по умолчанию 8 символов.

Ни одна из этих настроек не действует на учётную запись, созданную через внешнего провайдера.

Ограничения входа

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «Ограничения входа».
3. Установите флажок «Требовать двухфакторную аутентификацию для администраторов».
4. Нажмите «Сохранить изменения».

В том же разделе находятся:

- «Включить режим администратора» — администратор проходит аутентификацию ещё раз перед открытием административной панели;
- «Уведомление по email о неизвестных входах» — пользователь получает письмо, когда вход выполнен с устройства или IP-адреса, которые раньше не использовались;
- «Требовать подтверждение email при блокировке учётной записи» — заблокированная учётная запись подтверждает почтовый ящик перед следующим входом.

Двухфакторная аутентификация для всех пользователей инстанса, включённые протоколы доступа к Git и аутентификация по паролю для Git через HTTPS задаются в том же разделе и описаны на странице «Пользователи и доступ». Аутентификация по паролю для веб-интерфейса описана на странице «Аутентификация».

Внешний провайдер аутентификации

При использовании LDAP, SAML или OIDC учётные записи и членство в группах хранятся в каталоге, и закрытая там учётная запись теряет доступ к Deckhouse Code без действий внутри инстанса. Провайдеры, синхронизация и связывание учётных записей описаны на странице «Аутентификация».

Видимость и контроль доступа

Настройки находятся в разделе «Admin» → «Настройки» → «Основные» → «Видимость и контроль доступа»:

- «Уровень видимости проектов по умолчанию», «Уровень видимости сниппетов по умолчанию» и «Уровень видимости групп по умолчанию» — задайте значение «Приватный», чтобы новый проект, сниппет или группа были закрыты, пока их не откроют;
- «Ограниченные уровни видимости» — уровни, которые не может выбрать неадминистратор. Если отметить здесь публичный и внутренний уровни, обычный пользователь создаёт только приватные проекты;
- срок хранения удалённой группы или проекта — время, в течение которого удаление ещё обратимо;
- «Минимальная роль по умолчанию, необходимая для создания проектов» и «Включенные протоколы доступа к Git» — описаны на странице «Пользователи и доступ»;
- «RSA SSH keys», «DSA SSH keys», «ECDSA SSH keys» и остальные списки ключей — каждый из них запрещает свой алгоритм или задаёт минимальную длину ключа, которую инстанс принимает от пользователя.

Запретите ключи DSA, задайте минимальную длину ключа RSA 3072 бита, ключа ECDSA — 256 бит. Списки `ECDSA_SK` и `ED25519_SK` оставьте включёнными, пока используются аппаратные ключи FIDO2 или U2F.

Ограничения учётных записей

Настройки находятся в разделе «Admin» → «Настройки» → «Основные» → «Аккаунт и ограничения»:

- «Лимит проектов по умолчанию» — максимальное количество проектов, которые создаёт один пользователь;
- «Максимальная продолжительность сессии» — длительность сессии в минутах, по умолчанию 7 дней; изменение применяется после перезапуска инстанса;
- «Настройки сессии» — момент, с которого отсчитывается длительность сессии: «Истечение срока с момента создания сессии» или «Истечение срока с момента последней активности сессии»;
- «Запомнить меня» → «Разрешить пользователям использовать функцию „Запомнить меня“» — со снятым флажком сессия заканчивается по истечении срока независимо от действий пользователя;
- «Требовать дату истечения» — новый персональный, групповой или проектный токен получает дату истечения;
- «Пользовательские OAuth-приложения» — со снятым флажком пользователь не регистрирует в инстансе собственный клиент OAuth;
- «Сделать профили новых пользователей приватными по умолчанию» в группе «Ограничения пользователя» — новая учётная запись получает приватный профиль, и другие пользователи видят страницу профиля без активности, групп и проектов;
- остальные настройки группы «Ограничения пользователя» — создание групп и проектов, описано на странице «Пользователи и доступ».

Перезапустите инстанс после изменения длительности сессии:

Пакет для ОС Linux

```
sudo gitlab-ctl restart
```

Omnibus Docker

```
docker restart code
```

Helm-чарт

```
d8 k -n code rollout restart deploy/<WEBSERVICE_DEPLOYMENT>
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code rollout restart deploy -l 'app.kubernetes.io/component in
(webservice, sidekiq)'
```

Импорт и экспорт

Настройки находятся в разделе «Admin» → «Настройки» → «Основные» → «Настройки импорта и экспорта»:

- «Источники импорта» — системы, из которых импортируется проект; по умолчанию выключены все источники, а источник «Repository by URL» принимает любой репозиторий Git по адресу;
- «Экспорт проекта» — со снятым флажком пользователь не собирает архив `.tar.gz` с данными проекта для переноса на другой инстанс;
- «Разрешить перенос групп и проектов GitLab путем прямого переноса» — выключено по умолчанию;
- «Экспорты администраторами без аудита» — пока флажок установлен, экспорт, выполненный администратором, не попадает в журнал аудита;
- «Максимальное количество одновременных заданий импорта для GitHub», «Максимальное количество одновременных заданий импорта для Bitbucket Cloud» и «Максимальное количество одновременных заданий импорта для Bitbucket Server» — количество задач импорта каждого источника, которые выполняются одновременно.

Настройки репозиториев

Ветка по умолчанию

1. Перейдите в «Admin» → «Настройки» → «Репозиторий».
2. Разверните раздел «Ветка по умолчанию».
3. В поле «Имя начальной ветки по умолчанию» укажите имя ветки, с которой начинается каждый новый репозиторий.
4. В группе «Защита начальной ветки по умолчанию» выберите защиту, с которой создаётся ветка по умолчанию, и задайте в полях «Разрешен push» и «Разрешен merge» роль, от которой ветка принимает изменения.

5. Снимите флажки «Разрешен force push» и «Разрешить разработчикам отправлять начальный коммит».
6. Нажмите «Сохранить изменения».

Настройки действуют на репозитории, созданные после изменения; в существующем репозитории правила веток остаются прежними.

Правила отправки изменений

Раздел «Push-правила» на той же странице задаёт правила, с которыми создаётся каждый проект инстанса: регулярные выражения для сообщения коммита, адреса электронной почты автора, имени файла и имени ветки, проверки коммитера и запрет на коммит секретов, например файлов `pem` или `id_rsa`. Правило со снятым разрешением на переопределение нельзя изменить в группе или проекте. Сами правила описаны на странице «Правила отправки изменений», уровень инстанса — на странице «Пользователи и доступ».

Настройки CI/CD

Токены аутентификации раннеров

Раздел «Admin» → «Настройки» → «CI/CD» → «Непрерывная интеграция и деплой» задаёт срок жизни токенов аутентификации раннеров уровня инстанса, группы и проекта; пока значение не задано, срок не ограничен. Раннер в сети сам ротирует токен по мере приближения срока истечения; раннер, который в этот момент был офлайн, регистрируется заново.

Раннеры

Раздел «Admin» → «Настройки» → «CI/CD» → «Раннеры» содержит:

- «Получить данные о версиях релизов раннеров с GitLab.com» — пока флажок установлен, инстанс запрашивает данные о версиях раннеров с GitLab.com; снимите его в установке без доступа в интернет;
- «Разрешить токен регистрации раннера» — со снятым флажком раннер регистрируется собственным токеном аутентификации вместо общего токена регистрации инстанса, группы или проекта.

Права токена задач

Раздел «Admin» → «Настройки» → «CI/CD» → «Права доступа токена заданий в CI/CD» содержит настройку «Включить и применять белый список токенов заданий для всех проектов». Пока она включена, задача обращается к другому проекту только тогда, когда тот перечислил проект задачи в своём списке, а вариант с доступом от всех групп и проектов скрыт.

Защита от спама и ботов

Раздел «Admin» → «Настройки» → «Отчетность» → «Защита от спама и ботов» содержит reCAPTCHA и ограничение количества учётных записей на один IP-адрес. Раздел применяется для инстанса, доступного из интернета.

Статистика использования

В разделе «Admin» → «Настройки» → «Метрики и профилирование» подраздел «Статистика использования» содержит проверку обновлений и Service Ping, а подраздел «Отслеживание событий» — отправку событий. Отслеживание событий выключено, его флажок доступен только для чтения; выключенную проверку обновлений нельзя включить обратно из интерфейса.

Сетевые ограничения

Раздел «Admin» → «Настройки» → «Сеть» содержит ограничения частоты запросов: «Ограничения скорости пользователей и IP-адресов» для инстанса в целом, а также «Ограничения скорости реестра пакетов», «Ограничения частоты запросов прокси зависимостей», «Лимиты скорости файлов API», «Ограничения скорости поиска», «Устаревшие ограничения скорости API», «Ограничения на частоту запросов Git HTTP», «Ограничения на частоту запросов Git LFS», «Ограничения на частоту запросов для обсуждений», «Ограничения на частоту запросов для заметок», «Ограничения скорости импорта и экспорта» и «Ограничения скорости создания пайплайна» для своего типа запросов, каждое из которых переопределяет общее ограничение. Настройка «Защищенные пути» ограничивает запросы к перечисленным в ней путям. Настройка «Исходящие запросы» ограничивает адреса, к которым обращаются хуки и интеграции инстанса.

Защита на уровне платформы

В модуле Deckhouse Kubernetes Platform трафик между компонентами, доступ к неймспейсу `d8-code`, сетевые политики и экспорт событий аудита настраиваются в кластере. Они описаны на странице «Настройки безопасности» в документации модуля.

Ключи доступа

Раздел ключей доступа предоставляет полный обзор всех токенов (персональные, групповые, проектные) и ключей (SSH, GPG) на платформе Code. Администраторы могут переключать вкладки с различными видами ключей, применять фильтры и опции сортировки, и удалять или отзывать ключи, чтобы ограничить доступ пользователей и повысить безопасность.

Цели

Раздел ключей доступа используется в следующих целях:

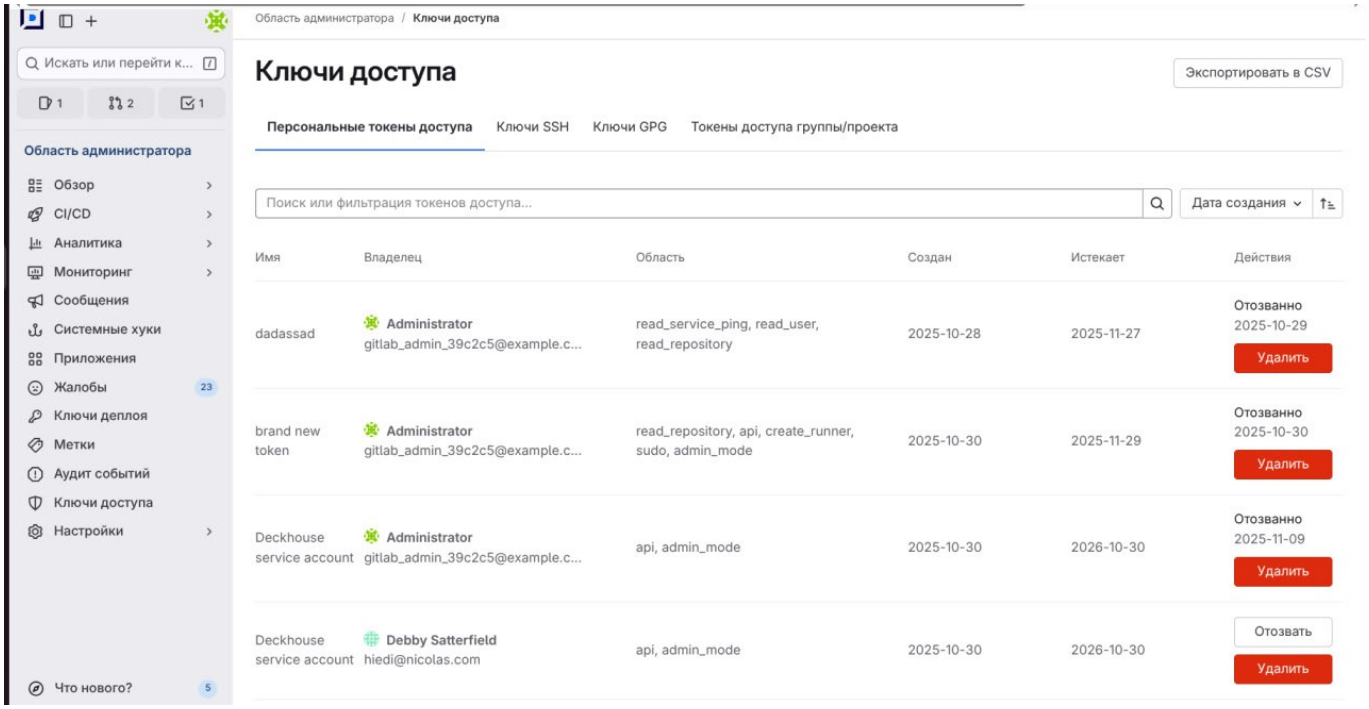
- Предоставление полной информации обо всех ключах доступа (владелец, область использования, даты создания/истечения срока/отзыва).

- Фильтрация и сортировка ключей доступа.
- Управление жизненным циклом ключей: отзыв, удаление.

Работа с разделом ключей доступа через UI

Чтобы начать работу с ключами доступа, перейдите в режим администратора и в боковом меню выберите пункт «Ключи доступа». Откроется таблица ключей доступа. Раздел состоит из 4 вкладок:

- **Персональные токены доступа** — список токенов, принадлежащих реальным пользователям
- **Ключи SSH** — список ключей SSH, принадлежащих реальным пользователям.
- **Ключи GPG** — список ключей GPG, принадлежащих реальным пользователям.
- **Токены доступа группы/проекта** — список токенов, принадлежащих группам верхнего уровня или проектам.



Наборы действий над ключами доступа

Набор действий, которые можно совершить над ключом доступа, зависит от его типа. В таблице представлена зависимость действий от типа ключа.

Операция\Тип ключа	Персональный токен доступа	Ключ SSH	Ключ GPG	Токен доступа группы/ проекта
Удаление	Да	Да	Да	Да
Отзыв	Да	Нет	Нет	Да

Сервисные аккаунты

Сервисный аккаунт — это учётная запись, предназначенная для использования в автоматизированных скриптах. Такие аккаунты применяются в CI/CD-пайплайнах и интеграциях. Сервисный аккаунт нельзя использовать для аутентификации в веб-интерфейсе, а также для выполнения действий от его имени через имперсонацию.

Создание сервисного аккаунта

Rails-консоль

Сервисный аккаунт создаётся из Rails-консоли. Откройте консоль для используемого типа установки:

Пакет для ОС Linux

```
sudo gitlab-rails console
```

Omnibus Docker

```
docker exec -it code gitlab-rails console
```

`code` — имя контейнера из команды `docker run` в разделе «Быстрый старт».

Helm-чарт

```
d8 k -n code exec -it deploy/code-toolbox -- gitlab-rails console -e production
```

Модуль Deckhouse Kubernetes Platform

Консоль входит в набор служебных инструментов Toolbox:

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rails console -e
production
```

Создание аккаунта

1. Используя Rails-консоль, подготовьте параметры с описанием создаваемого аккаунта. Заполните поля `name`, `username`, `email` и `admin` и задайте остальные параметры на основе следующего примера:

   ```
   user_args = {
   name: 'kaiten_sa',
   username: 'kaiten_sa',
   email: 'kaiten_sa@flant.com',
   admin: false,
   user_type: :service_account,
   organization_id: Organizations::Organization.default_organization.id,
   password_automatically_set: true,
   force_random_password: true,
   skip_confirmation: true
   }
   ```

2. Выберите пользователя, от имени которого будет создан сервисный аккаунт, после чего выполните создание пользователя:

   ```
   user = User.find_by_username('root')
   Users::CreateService.new(user, user_args).execute
   ```

Генерация токена доступа

Чтобы сгенерировать токен доступа, используйте Personal access tokens API от GitLab.

События аудита безопасности

Аудит безопасности — это детальный анализ вашей инфраструктуры, направленный на выявление потенциальных уязвимостей и небезопасных практик. Code помогает упростить процесс аудита с помощью событий аудита, которые позволяют отслеживать широкий спектр действий, происходящих в системе.

Назначение и область применения

Механизм регистрации событий безопасности (события аудита) используется для:

- фиксации действий пользователей и администраторов, связанных с изменением конфигурации системы;
- регистрации инцидентов информационной безопасности;
- обеспечения возможности расследования происшествий и восстановления полной картины действий в системе.

Область применения — все уровни платформы Code, от отдельных проектов и групп до конфигурации инстанса. События фиксируются независимо от прав пользователя (при условии, что у него есть доступ к совершению действия) и сохраняются централизованно.

Технические меры регистрации событий

Система аудита реализует следующие меры:

- **Централизованная регистрация событий** — все события фиксируются в едином журнале аудита.
- **Непрерывный сбор данных** — событие фиксируется в реальном времени по ходу выполнения бизнес-операции. Например, при входе в систему, смене адреса электронной почты пользователя или включении возможности выполнять `force push` в защищённой ветке.
- **Защита целостности** — журнал событий доступен только в режиме чтения для администраторов. Удалить или изменить события в журнале невозможно.
- **Доступ через UI и API** — просмотр и фильтрация событий возможны как в интерфейсе администратора, так и через специализированный API.

Сценарии использования

События аудита безопасности позволяют отследить:

- кто и когда изменил уровень доступа пользователя в проекте Code;
- случаи создания и удаления пользователей;
- признаки подозрительной активности — например, массовая смена адресов электронной почты сотрудников или удаление репозиториев;
- случаи изменения переменных окружения в CI/CD;
- случаи изменения уровня видимости проектов и групп.

События аудита помогают:

- оценивать риски и усиливать меры безопасности;
- своевременно реагировать на инциденты.

Доступ к событиям аудита

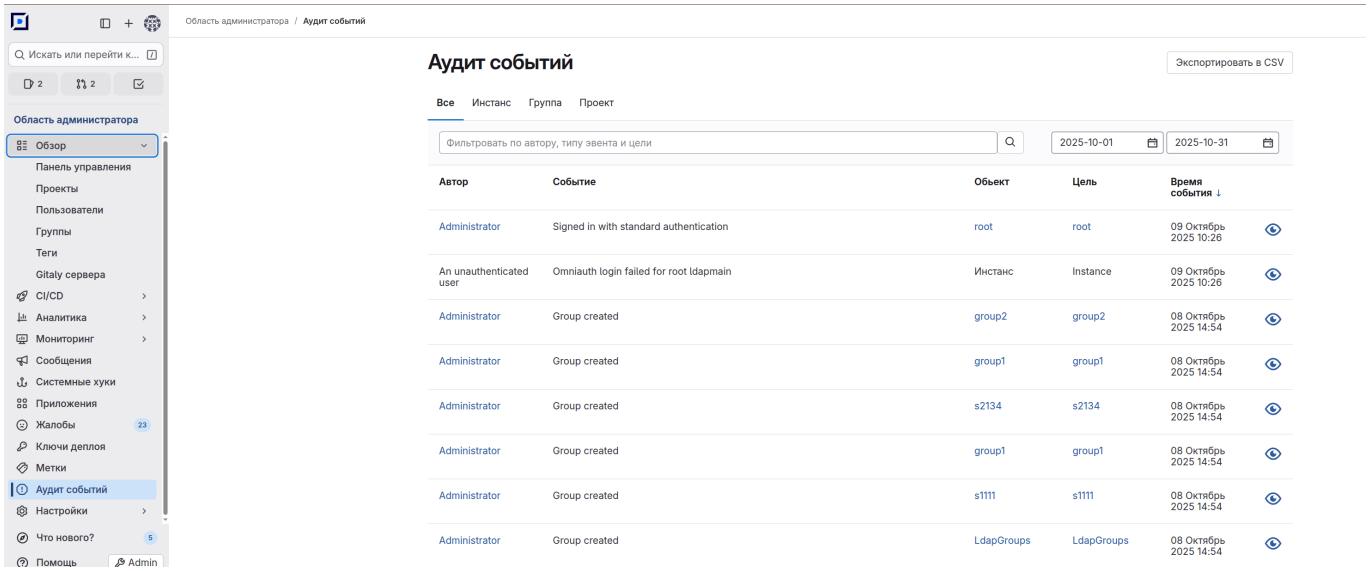
Доступ через UI

Чтобы получить доступ к событиям аудита, войдите в режим администратора и в боковом меню выберите пункт «Аудит событий». Откроется таблица событий аудита безопасности.

Описание столбцов в таблице:

- **Автор** — имя пользователя, запустившего событие.
- **Событие** — системное сообщение с информацией о событии.

- **Объект** — область, к которой относится событие (название инстанса, группы, проекта, имя пользователя).
- **Цель** — сущность, которая была изменена (название проекта, защищённой ветки, CI-переменной, имя пользователя, токена, и т. д.).
- **Время события** — дата и время наступления события.



Доступ через API

Deckhouse Code предоставляет следующий API-метод для получения списка событий аудита:

`POST /api/v4/admin/audit_events/search`

Допускается фильтрация по датам, текстовому поиску и типам сущностей.

> ⚠ Диапазон дат должен находиться в пределах одного календарного месяца. Если значения `created_after` и `created_before` относятся к разным месяцам, параметр `created_before` будет автоматически приведён к последнему дню месяца, в который входит `created_after`.

Параметры запроса

Параметр	Тип	Обязательный	Описание
`created_after`	Строка	Нет	Начальная дата (включительно) в формате ISO8601. По умолчанию — начало текущего месяца
`created_before`	Строка	Нет	Конечная дата (включительно) в формате ISO8601. По умолчанию — конец текущего месяца
`q`	Строка	Нет	Полнотекстовый поиск по сообщению события
`sort`	Строка	Нет	Сортировка по дате создания. Возможные значения: `created_asc`, `created_desc` (по умолчанию)
`entity_types`	Массив строк	Нет	Список типов сущностей для фильтрации: `User`, `Project`, `Group`, `Gitlab::Audit::InstanceScope`

Пример запроса:

```
curl --request POST "https://example.com/api/v4/admin/audit_events/search" \
     --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
     --header "Content-Type: application/json" \
     --data '{
       "created_after": "2025-08-01",
       "created_before": "2025-08-31",
       "q": "repository",
       "sort": "created_desc",
       "entity_types": ["Project"]
     }'
```

Содержимое событий аудита

Каждое событие аудита содержит дату, время, данные об IP-адресе и учётной записи пользователя, а также всю необходимую информацию об области изменения, измененном объекте и о том, что именно было изменено.

Перечень событий аудита

В таблице приведены примеры системных сообщений. В production-среде события аудита содержат полную информацию в самом сообщении или в дополнительном JSON-поле с данными.

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`2fa_login_failed`	User 2fa login failed	Обнаружена неудачная попытка входа с двухфакторной аутентификацией.	
`access_approved`	User access was approved	Пользователю одобрен запрос на доступ в инстанс.	
`access_token_created`	Project/Group access token created	Создан токен доступа для проекта или группы.	
`access_token_revoked`	Project/Group access token revoked	Токен доступа был отозван.	
`added_gpg_key`	Added new gpg key to user	Пользователь добавил новый GPG-ключ.	
`added_ssh_key`	User added new ssh key	Пользователь добавил новый SSH-ключ.	
`application_created`	Application was created	Создано приложение (OAuth или интеграция).	
`application_deleted`	Application deleted	Приложение было удалено.	
`application_secret_renew`	Application secret renew	Обновлён секрет приложения.	
`application_updated`	Application Updated	Изменены параметры приложения.	
`ci_cd_job_token_removed_from_allowlist`	Disallow group to use job token	В проекте ограничено использование CI/CD Job Token	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
		определённой группой.	
`ci_cd_job_token_added_to_allowlist`	Allow group to use job token	В проекте разрешено использование CI/CD Job Token определённой группой.	
`ci_variable_created`	Ci variable `#{key}` created	Создана новая переменная CI/CD.	
`ci_variable_deleted`	Ci variable `#{key}` deleted	Удалена переменная CI/CD.	
`ci_variable_updated`	Ci variable updated (Value, Protected)	Изменено значение или параметры защиты переменной CI/CD.	
`deploy_key_created`	Deploy key added	Создан новый ключ развёртывания (deploy key) для проекта/инстанса.	
`deploy_key_deleted`	Deploy key was deleted	Удалён ключ развертывания.	
`deploy_key_disabled`	Deploy key disabled	Ключ развёртывания отключён.	
`deploy_key_enabled`	Deploy key enabled	Ключ развёртывания включён.	
`deploy_token_created`	Deploy token created	Создан токен развёртывания (deploy token) для доступа к данным.	
`deploy_token_deleted`	Deploy token deleted	Удалён токен развёртывания.	
	Deploy token revoked	Токен развёртывания был отозван	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`deploy_token_revoked`		пользователем или системой.	
`feature_flag_created`	Created feature flag with description	Создан новый флаг функций (feature flag).	
`feature_flag_deleted`	Feature flag was deleted	Флаг функций был удалён.	
`feature_flag_updated`	Feature flag was updated	Обновлены параметры флага функций.	
`group_created`	Group was created	Создана новая группа.	
`group_export_created`	Group file export was created	Создан файл экспорта группы.	
`group_invite_via_group_link_created`	Invited group to group	В группу приглашена другая группа через групповую ссылку.	
`group_invite_via_group_link_deleted`	Revoked group from group	Доступ группы, приглашенной по ссылке, был отозван.	
`group_invite_via_group_link_updated`	Group access changed	Изменены параметры доступа группы через групповую ссылку.	
`group_invite_via_project_group_link_created`	Invited group to project	В проект приглашена группа через групповую ссылку.	
`group_invite_via_project_group_link_deleted`	Revoked group from project	Доступ группы к проекту был отозван.	
	Group access for project changed		

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`group_invite_via_project_group_link_updated`		Изменены параметры доступа группы к проекту.	
`group_updated`	Group updated (visibility, 2FA grace period)	Изменения в настройках группы (видимость, безопасность, лимиты, политика доступа).	• `repository_size_limit` • `two_factor_grace_period` • `lfs_enabled` • `membership_lock` • `path` • `require_two_factor_authentication` • `request_access_enabled` • `shared_runners_minutes_limit` • `share_with_group_lock` • `mentions_disabled` • `max_personal_access_token_lifetime` • `visibility_level` • `name` • `description` • `project_creation_level` • `default_branch_protected` • `seat_control` • `duo_features_enabled`

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
			• `prevent_forking_outside_group` • `allow_mfa_for_subgroups` • `default_branch_name` • `resource_access_token_creation_allowed` • `new_user_signups_cap` • `show_diff_preview_in_email` • `enabled_git_access_protocol` • `runner_registration_enabled` • `allow_runner_registration_token` • `emails_enabled` • `service_access_tokens_expiration_enforced` • `enforce_ssh_certificates` • `disable_personal_access_tokens` • `remove_dormant_members` • `remove_dormant_members_period` • `prevent_sharing_groups_outside_hierarchy`

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
			• `default_branch_protection_defaults` • `wiki_access_level`
`impersonation_initiated`	User root impersonated another user	Администратор начал сессию от имени другого пользователя.	
`impersonation_stopped`	User root stopped impersonation	Администратор завершил сессию от имени другого пользователя.	
`instance_settings_updated`	Instance settings updated: Signup enabled turned on	Изменены глобальные настройки инстанса.	Все поля с настройками инстанса, кроме зашифрованных.
`login_failed`	Attempt to login failed	Неудачная попытка входа в систему.	
`manually_trigger_housekeeping`	Housekeeping task	Запущена задача обслуживания репозитория вручную.	
`member_permissions_created`	New member access granted	Пользователю предоставлен доступ (роль) к группе или проекту.	
`member_permissions_destroyed`	Member access revoked	Доступ пользователя к проекту или группе отозван.	
`member_permissions_updated`	Member access updated	Изменены права или срок действия доступа пользователя.	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`merge_request_closed_by_project_bot`	Merge request `#{merge_request.title}` closed by project bot	Запрос на merge закрыт системным ботом проекта.	
`merge_request_created_by_project_bot`	Merge request `#{merge_request.title}` created by project bot	Запрос на merge создан системным ботом проекта.	
`merge_request_merged_by_project_bot`	Merge request `#{merge_request.title}` merged by project bot	Запрос на merge обработан системным ботом проекта.	
`merge_request_reopened_by_project_bot`	Merge request `#{merge_request.title}` reopened by project bot	Запрос на merge переоткрыт системным ботом проекта.	
`omniauth_login_failed`	Omniauth login failed for `#{user}` `#{provider}`	Ошибка входа через внешний OAuth/ Omniauth-провайдер.	
`password_reset_failed`	Password reset failed	Неудачная попытка сброса пароля пользователем.	
`personal_access_token_issued`	Personal access token issued	Выпущен новый токен доступа (personal access token).	
`personal_access_token_revoked`	Personal access token revoked	Токен доступа был отозван.	
`pipeline_deleted`	Pipeline deleted	Пайплайн CI/CD был удалён.	
`project_blobs_removal`	Project blobs removed	Массовое удаление объектов (blobs) из проекта.	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`project_created`	Project was created	Создан новый проект.	
`project_default_branch_changed`	Project default branch updated	Изменена ветка по умолчанию в проекте.	
`project_export_created`	Project export created	Создан файл экспорта проекта.	
`project_feature_updated`	Project features updated	Изменены уровни доступа к функциям проекта (issues, wiki и т. д.).	
`project_setting_updated`	Project settings updated	Изменены шаблоны merge commit и squash commit.	
`project_text_replacement`	Project text replaced	В проекте выполнена массовая замена текста.	
`project_topic_changed`	Project topic changed	Изменена тема проекта.	
`project_updated`	Project updated (name, namespace)	Изменены настройки проекта (имя, пространство имён, политики).	• `name` • `packages_enabled` • `reset_approvals_on_push` • `path` • `merge_requests_author_approval` • `merge_requests_disable_committers_approval` • `only_allow_merge_if_all_disc`

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
			`ussions_are_resolved` • `only_allow_merge_if_pipeline_succeeds` • `require_password_to_approve` • `disable_overriding_approvers_per_merge_request` • `repository_size_limit` • `project_namespace_id` • `namespace_id` • `printing_merge_request_link_enabled` • `resolve_outdated_diff_discussions` • `merge_requests_ff_only_enabled` • `merge_requests_rebase_enabled` • `remove_source_branch_after_merge` • `merge_requests_template` • `visibility_level` • `builds_access_level`

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
			• `container_registry_access_level` • `environments_access_level` • `feature_flags_access_level` • `forking_access_level` • `infrastructure_access_level` • `issues_access_level` • `merge_requests_access_level` • `metrics_dashboard_access_level` • `monitor_access_level` • `operations_access_level` • `package_registry_access_level` • `pages_access_level` • `releases_access_level` • `repository_access_level` • `requirements_access_level` • `security_and_compliance_access_level` • `snippets_access_level`

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
			• `wiki_access_level` • `merge_commit_template` • `squash_commit_template` • `runner_registration_enabled` • `show_diff_preview_in_email` • `selective_code_owner_removals`
`protected_branch_created`	Protected branch created	Создана защищённая ветка.	
`protected_branch_deleted`	Protected branch was deleted	Удалена защищённая ветка.	
`protected_branch_updated`	Protected branch was updated:	Обновлены правила защищённой ветки.	
`protected_tag_created`	Protected tag created	Создан защищённый тег.	
`protected_tag_deleted`	Protected tag was deleted	Удалён защищённый тег.	
`protected_tag_updated`	Protected tag updated:	Обновлены правила защищённого тега.	
`removed_gpg_key`	Removed gpg key from user	Удалён GPG-ключ пользователя.	
`removed_ssh_key`	User removed ssh key	Удалён SSH-ключ пользователя.	
`requested_password_reset`	User requested password change	Пользователь запросил сброс пароля.	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`revoked_gpg_key`	Revoked gpg key from user	GPG-ключ пользователя был отозван.	
`unban_user`	User was unban	Пользователь разблокирован (unban).	
`unblock_user`	User was unblocked	С пользователя снята блокировка (unblock).	
`user_access_locked`	User access locked	Учётная запись пользователя заблокирована.	
`user_access_unlocked`	User access unlocked	Учётная запись пользователя разблокирована.	
`user_activated`	User was activated	Учётная запись пользователя активирована.	
`user_banned`	User was banned	Пользователь забанен.	
`user_blocked`	User was blocked	Учётная запись пользователя заблокирована.	
`user_created`	User was created	Создан новый пользователь.	
`user_deactivated`	User was deactivated	Учётная запись пользователя деактивирована.	
`user_destroyed`	User was destroyed	Учётная запись пользователя удалена.	
`user_email_updated`	User email updated		

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
		Изменён адрес электронной почты пользователя.	
`user_logged_in`	User logged in	Успешный вход пользователя.	
`user_password_updated`	Password updated	Пароль пользователя изменён.	
`user_rejected`	User was rejected	Учётная запись пользователя отклонена (например, при регистрации).	
`user_removed_two_factor`	Two factor disabled	Пользователь отключил двухфакторную аутентификацию.	
`user_settings_updated`	User settings updated	Обновлены настройки профиля пользователя.	• `name` • `public_email` • `otp_secret` • `otp_required_for_login` • `admin` • `private_profile`
`user_signup`	User was registered	Пользователь зарегистрирован.	
`user_switched_to_admin_mode`	User switched to admin mode	Пользователь включил режим администратора.	
`user_username_updated`	Username updated	Изменено имя пользователя (username).	
`webhook_created`	Webhook was created		

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
		Создан вебхук для проекта, группы или инстанса.	
`webhook_destroyed`	System hook removed	Вебхук удалён.	
`group_deleted`	Group was deleted	Группа удалена.	
`project_deleted`	Project was deleted	Проект удалён.	
`logout`	User logged out	Пользователь вышел из системы.	
`unauthenticated_session`	Redirected to login	Система перенаправила неаутентифицированного пользователя на страницу входа.	
`ci_runners_bulk_deleted`	CI runner bulk deleted: Errors:	Массовое удаление CI runners.	
`ci_runner_registered`	CI runner created via API	Регистрация CI runner через API.	
`ci_runner_unregistered`	CI runner unregistered	Отмена регистрации CI runner.	
`ci_runner_token_reset`	CI runner registration token reset	Сброшен токен CI runner.	
`ci_runner_assigned_to_project`	CI runner assigned to project	Runner был привязан к проекту.	
`ci_runner_unassigned_from_project`	CI runner unassigned from project	Runner был отвязан от проекта.	
`ci_runner_created`	CI runner created via UI	Runner был создан через графический интерфейс.	
`package_registry_package_published`	`#{name}` package version `#{version}` has been published	В реестре пакетов опубликован новый пакет.	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
`package_registry_package_deleted`	package version `#{package.version}` has been deleted	Пакет удалён из реестра пакетов.	
`group_access_token_destroyed`	Group access token destroyed	Токен доступа группы был удалён.	
`group_access_token_revoked`	Group access token revoked	Токен доступа группы был отозван.	
`personal_access_token_destroyed`	Personal access token destroyed	Персональный токен доступа был удалён.	
`project_access_token_destroyed`	Project access token destroyed	Токен доступа проекта был удалён.	
`project_access_token_revoked`	Project access token revoked	Токен доступа проекта был отозван.	
`approval_rule_created`	Merge request\Project\Group approval rule created	Правило апрува для запроса на слияние, проекта или группы было создано.	
`approval_rule_updated`	Merge request\Project\Group approval rule updated	Правило апрува для запроса на слияние, проекта или группы было обновлено.	
`approval_rule_deleted`	Merge request\Project\Group approval rule deleted	Правило апрува для запроса на слияние, проекта или группы было удалено.	
`approval_rule_propagated`	Approval rules `#{rule_names}` propagated to `#{project_and_group_names}`	Правила апрува были распространены в эти проекты и группы.	
`variable_viewed_api`	User viewed all CI/CD variables via API for this scope or User	Все CI/CD переменные были просмотрены через	

Название	Системное сообщение	Назначение	Отслеживаемые атрибуты
	viewed CI/CD variable `#{variable.key}` via API	API для данной области или CI/CD переменная `#{variable.key}` была просмотрена через API.	
`group_unarchived`	User unarchived group	Группа была разархивирована.	
`group_archived`	User archived group	Группа была архивирована.	
`comment_deleted`	Comment deleted	Комментарий был удалён.	
`comment_created`	Comment created	Комментарий был создан.	
`comment_updated`	Comment updated	Комментарий был обновлён.	
`user_records_migrated_to_ghost`	User records migrated to ghost	Записи пользователя были перенесены ghost-пользователю.	
`user_enable_passkey`	User enable new passkey	Новый passkey был включён.	
`user_disable_passkey`	User disable passkey	Passkey был отключён.	
`push_rule_propagated`	Push rule propagated	Push-правило было распространено.	
`push_rule_updated`	Push rule updated	Существующее push-правило было обновлено.	

Мониторинг и лимиты

Мониторинг инстанса Deckhouse Code складывается из лимитов, заданных в веб-интерфейсе, и встроенного Prometheus каждого типа установки.

Управление ресурсами

Лимиты и настройки производительности инстанса задаются администратором в веб-интерфейсе:

1. Настройте лимиты на репозитории и артефакты:
    - Перейдите в «Admin» → «Настройки» → «Аккаунт и ограничения».
2. Настройте производительность системы:
    - Перейдите в «Admin» → «Настройки» → «Сеть».

Встроенный мониторинг

У каждого типа установки свой источник метрик или оповещений; вкладка называет тип.

Пакет для ОС Linux

Prometheus и экспортёры (node, postgres, redis и другие) устанавливаются вместе с пакетом и слушают только адрес `127.0.0.1`, наружу порты не открыты. Проверьте их состояние:

```
sudo gitlab-ctl status | grep -E "prometheus|exporter"
# Готовность встроенного Prometheus.
curl -s http://127.0.0.1:9090/-/ready
```

Чтобы метрики забирала внешняя система мониторинга, задайте адрес прослушивания Prometheus в файле `/etc/gitlab/gitlab.rb`:

```
prometheus['listen_address'] = '0.0.0.0:9090'
```

Примените настройку:

```
sudo gitlab-ctl reconfigure
```

Ограничьте доступ к порту `9090` на файрволе адресами системы мониторинга.

firewalld (РЕД ОС, AlmaLinux)

```
sudo firewall-cmd --permanent --new-zone=monitoring
sudo firewall-cmd --permanent --zone=monitoring --add-source=<MONITORING_IP>/32
sudo firewall-cmd --permanent --zone=monitoring --add-port=9090/tcp
# Зона принимает весь трафик с этого адреса, поэтому открываются и базовые
сервисы.
sudo firewall-cmd --permanent --zone=monitoring --add-service={ssh,http,https}
sudo firewall-cmd --reload
```

ufw (Ubuntu)

```
sudo ufw allow from <MONITORING_IP> to any port 9090 proto tcp
```

Omnibus Docker

В контейнере работают тот же Prometheus и те же экспортёры, кроме `node_exporter`: образ его отключает. Образ объявляет только порты `80`, `443` и `22`, поэтому порт Prometheus нужно опубликовать при создании контейнера: добавьте `-p 9090:9090` в команду `docker run`, у запущенного контейнера список опубликованных портов не меняется.

Задайте `prometheus['listen_address']` в файле `/etc/gitlab/gitlab.rb` в томе с конфигурацией; значение, его смысл и ограничение доступа к порту описаны на вкладке «Пакет для ОС Linux». Настройка применится при следующем старте контейнера: команда `gitlab-ctl reconfigure` выполняется при каждом запуске.

Helm-чарт

Helm-чарт создаёт объект ServiceMonitor (кастомный ресурс Prometheus Operator) для компонента, если для этого компонента заданы `metrics.enabled` и `metrics.serviceMonitor.enabled`. Метрики Sidekiq собираются через PodMonitor, который включается ключом `metrics.podMonitor.enabled`. Объекты создаются, если в кластере доступен API `monitoring.coreos.com/v1` или `global.monitoring.enabled` равно `true`. Ключ `additionalLabels` задаёт лейблы, по которым объект находит Prometheus.

```
global:
  monitoring:
    enabled: true
gitlab:
  webservice:
    metrics:
      serviceMonitor:
        enabled: true
        additionalLabels:
          release: <PROMETHEUS_RELEASE_LABEL>
  sidekiq:
    metrics:
      podMonitor:
        enabled: true
        additionalLabels:
          release: <PROMETHEUS_RELEASE_LABEL>
```

Блок `metrics.serviceMonitor` повторяется для остальных компонентов, которые отдают метрики: `gitaly`, `praefect`, `gitlab-shell`, `gitlab-pages`, `kas` и `gitlab-exporter` в секции `gitlab`, а также `registry` на верхнем уровне значений.

Экземпляр Prometheus, настроенный на отслеживание объектов ServiceMonitor с такими лейблами — в неймспейсе `code` или во всём кластере, в зависимости от его собственной настройки, — обнаруживает их и собирает метрики без дополнительной настройки со стороны релиза.

Модуль Deckhouse Kubernetes Platform

Модуль поставляет собственные правила оповещений: список и действия по каждому оповещению приведены в разделе «Алерты» документации модуля.

Режим обслуживания

Режим обслуживания позволяет администраторам свести операции записи к минимуму на время проведения регламентных работ. Его основная цель — заблокировать все внешние действия, изменяющие внутреннее состояние базы данных PostgreSQL, а также файлов, Git-репозиториев и хранилищ образов контейнеров.

Когда режим обслуживания включён, уже запущенные действия завершаются за короткое время, поскольку новые действия перестают поступать, а изменения внутреннего состояния минимальны. В таком состоянии проще выполнять различные регламентные работы.

Режим обслуживания разрешает большинство внешних действий, которые не изменяют внутреннее состояние. На уровне HTTP блокируются запросы с методами `POST`, `PUT`, `PATCH` и `DELETE`.

Включение режима обслуживания

Включите режим обслуживания от имени администратора одним из следующих способов:

- **Через веб-интерфейс**:

  1. Перейдите в раздел «Admin» → «Настройки» → «Основные».
  2. Разверните раздел «Режим обслуживания» и установите флажок «Включить режим обслуживания».
  3. При необходимости добавьте текст сообщения, которое будет отображаться в баннере.
  4. Нажмите «Сохранить изменения».

- **Через API**:

  Выполните следующий запрос:

  ```
  curl --request PUT --header "PRIVATE-TOKEN: $ADMIN_TOKEN" \
    "<INSTANCE_URL>/api/v4/application/settings?maintenance_mode=true"
  ```

Отключение режима обслуживания

Отключите режим обслуживания одним из следующих способов:

- **Через Веб-интерфейс**:

  1. Перейдите в раздел «Admin» → «Настройки» → «Основные».
  2. Разверните раздел «Режим обслуживания» и снимите флажок «Включить режим обслуживания».
  3. Нажмите «Сохранить изменения».

- **Через API**:

Выполните следующий запрос:

```
curl --request PUT --header "PRIVATE-TOKEN: $ADMIN_TOKEN" \
  "<INSTANCE_URL>/api/v4/application/settings?maintenance_mode=false"
```

Особенности работы в режиме обслуживания

Когда режим обслуживания включён, в верхней части интерфейса отображается баннер. При необходимости для него можно задать собственное сообщение.

При попытке выполнить недопустимую операцию записи пользователь получит сообщение об ошибке.

> В некоторых случаях визуальная реакция на действие может вводить в заблуждение. Например, при добавлении проекта в избранное кнопка «В избранное» меняется на «Убрать из избранного». Это связано с тем, что интерфейс обновляется до получения результата `POST`-запроса.

Действия администратора

Администраторы могут по-прежнему редактировать настройки приложения. Это позволяет отключить режим обслуживания после его включения.

Аутентификация

Все пользователи могут входить в систему и выходить из неё, но создание новых пользователей блокируется.

Если на время обслуживания запланирована синхронизация LDAP, она будет завершена с ошибкой, поскольку создание пользователей отключено. По той же причине будет завершено с ошибкой создание пользователей на основе SAML и других провайдеров OmniAuth.

> Когда создание пользователя блокируется при входе через LDAP, пользователь видит сообщение о том, что инстанс находится в режиме обслуживания (только для чтения), а не стандартную ошибку запрета доступа.

Операции с Git

Все операции чтения Git продолжают работать, включая `git clone` и `git pull`.

Все операции записи завершаются с ошибкой — как через командную строку, так и через web-редактор. При этом отображается следующее сообщение:

```
Git push is not allowed because this instance is currently in (read-only) maintenance
mode.
```

Запросы на слияние и задачи

Все операции записи, не перечисленные в исключениях, завершаются с ошибкой. Например, пользователь не может изменить запрос на слияние или задачу.

Входящая почта

Создание ответов на задачи, новых задач (включая задачи Service Desk) и запросов на слияние по электронной почте завершается с ошибкой. Обработчики входящей почты пропускают обработку, пока включён режим обслуживания.

Исходящая почта

Уведомления по электронной почте продолжают приходить, но письма, для отправки которых требуется запись в базу данных (например, для сброса пароля), не доставляются.

REST API

Для большинства JSON-запросов методы `POST`, `PUT`, `PATCH` и `DELETE` блокируются. API возвращает ответ `503` с сообщением об ошибке `system is in maintenance mode` и заголовком `Retry-After`.

Разрешены только следующие запросы:

HTTP-запрос	Разрешённые маршруты	Примечания
`POST`	`/admin/application_settings/general`	Обновление настроек приложения в интерфейсе администратора
`PUT`	`/api/v4/application/settings`	Обновление настроек приложения через API
`POST`	`/users/sign_in`	Вход пользователей в систему
`POST`	`/users/sign_out`	Выход пользователей из системы
`POST`	`/oauth/token`	Получение OAuth-токенов
`POST`	`/admin/session`, `/admin/session/destroy`	Использование режима администратора (Admin Mode)
`POST`	Пути, оканчивающиеся на `/compare`	Сравнение ревизий Git
`POST`	`*.git/git-upload-pack`	Выполнение `git pull` и `git clone`
`POST`	`/api/v4/internal`	Маршруты внутреннего API
`POST`	`/admin/sidekiq`	Управление фоновыми задачами в разделе «Admin»

GraphQL API

Запросы `POST /api/graphql` разрешены, но GraphQL-мутации блокируются со следующим сообщением об ошибке: `You cannot perform write operations on a read-only instance`.

Непрерывная интеграция

Во время обслуживания действуют следующие ограничения:

- новые задачи и пайплайны не запускаются ни по расписанию, ни вручную;
- задачи, которые уже выполнялись на момент включения режима обслуживания, продолжают отображаться в интерфейсе со статусом `running`, даже если они завершились на раннере;
- задачи в состоянии `running` не завершаются по тайм-ауту после превышения лимита времени проекта;

- пайплайны нельзя запустить, перезапустить или отменить. Новые задачи в существующих пайплайнах также не создаются;
- статус раннеров в разделе «Admin» → «Раннеры» не обновляется.

После отключения режима обслуживания новые задачи снова начинают выполняться.

Задачи, которые были в состоянии `running` до включения режима обслуживания, возобновляются, а их логи снова обновляются.

> После отключения режима обслуживания рекомендуется перезапустить пайплайны, которые были в состоянии `running` на момент включения режима обслуживания.

Развёртывания

Развёртывания не выполняются, поскольку связанные с ними пайплайны не завершаются. Отключите автоматические развёртывания на время режима обслуживания и снова включите их после завершения работ.

Хранилище образов контейнеров

Команда `docker push` завершается следующей ошибкой:

```
denied: requested access to the resource is denied
```

При этом операция `docker pull` продолжает работать.

Реестр пакетов

Реестр пакетов позволяет устанавливать пакеты, но не публиковать их.

Фоновые задачи

Фоновые задачи (включая cron-задачи и задачи Sidekiq) продолжают выполняться как обычно, поскольку они не отключаются автоматически.

Так как фоновые задачи выполняют операции, которые могут изменить внутреннее состояние инстанса, на время режима обслуживания может потребоваться отключить часть из них или все задачи целиком.

Чтобы отслеживать очереди и отключать задачи:

1. Перейдите в раздел «Admin» → «Мониторинг» → «Фоновые задачи».
2. В панели Sidekiq выберите «Cron» и отключите задачи по отдельности или все сразу, нажав «Disable All».

Управление инцидентами

Функции управления инцидентами ограничены. Создание алертов и инцидентов полностью приостановлено, а связанные с ними уведомления и оповещения не отправляются.

Флаги функций

- Флаги функций разработки нельзя включать или отключать через API, но можно переключать через консоль Rails.
- Сервис флагов функций продолжает отвечать на запросы проверки состояния флагов, но переключать сами флаги нельзя.

Расширенный поиск

Документация для администратора по расширенному поиску в Deckhouse Code: эксплуатация индексации после подключения OpenSearch. В модуле Deckhouse Kubernetes Platform OpenSearch подключается в ресурсе CodeInstance, это описано в разделе «Расширенный поиск» документации модуля. Инструкции для пользователей — в руководстве пользователя.

Эксплуатация

Управление индексацией, мониторинг и устранение неполадок.

Управление на уровне инстанса

Чтобы управлять индексацией на уровне инстанса, перейдите в «Admin» → «Настройки» → «Поиск».

Раздел доступен после подключения OpenSearch.

Приостановка индексации

Установите флаг «Приостановить индексирование OpenSearch», чтобы приостановить фоновые задачи индексации и переиндексации. Чтобы снова включить индексацию, отключите флаг «Приостановить индексирование OpenSearch». После этого Sidekiq pause control автоматически возобновит работу в течение нескольких минут.

Режим индексации веток

В проектах могут использоваться следующие режимы индексации веток:

Режим	Описание
Только ветка по умолчанию	Индексируется только ветка по умолчанию для всех проектов
Разрешить регулярное выражение для веток на уровне проекта	Для проектов можно задать regex для индексации дополнительных веток

⚠ После смены режима индексации веток выполните ручную переиндексацию кода. Результаты поиска могут быть неполными до завершения индексации.

Статус индексов OpenSearch

На странице отображается таблица индексов:

Суффикс индекса	Область поиска
`code`	Код (`blobs`)
`commits`	Коммиты
`wiki`	Wiki-страницы
`notes`	Комментарии
`milestones`	Этапы (milestones)
`merge-requests`	Запросы на слияние
`work-items`	Задачи (work items)

Имя индекса в OpenSearch состоит из общего префикса инстанса и суффикса из таблицы, например `deckhouse-development-code`.

Для каждого индекса показаны: имя в OpenSearch, наличие, количество документов, состояние индекса.

Операции над индексами

На той же странице доступны следующие операции:

- **Переиндексировать** — переиндексация одного индекса (удаляет существующие документы и ставит фоновые задачи).
- **Переиндексировать все индексы** — переиндексация всех индексов.

⚠ Операция «Переиндексировать» удаляет существующие документы. Результаты поиска могут быть неполными до завершения фоновой индексации.

Индекс `commits` переиндексируется вместе с индексом `code`: отдельной операции для коммитов нет. По этой же причине у коммитов нет отдельного значения `schema_class`.

Операции над индексами также можно выполнять с помощью запросов к API OpenSearch.

Мониторинг

Метрики

Обращения к OpenSearch учитываются в Prometheus отдельно для HTTP-запросов (поиск в UI и API) и для фоновых задач Sidekiq (индексация). Имена метрик содержат `elasticsearch` — это историческое название в GitLab, метрики относятся к OpenSearch.

HTTP-запросы (поиск пользователей):

Метрика	Описание
`http_elasticsearch_requests_total`	Число обращений к OpenSearch за один HTTP-запрос
`http_elasticsearch_requests_duration_seconds`	Суммарное время обращений к OpenSearch за один HTTP-запрос
`http_elasticsearch_requests_failed_total`	Число неудачных обращений за один HTTP-запрос (ошибки подключения или авторизации) — **добавлено в Deckhouse Code**

Sidekiq (фоновая индексация):

Метрика	Описание
`sidekiq_elasticsearch_requests_total`	Число обращений к OpenSearch за выполнение одной Sidekiq-задачи
`sidekiq_elasticsearch_requests_duration_seconds`	Суммарное время обращений к OpenSearch за выполнение одной Sidekiq-задачи
`sidekiq_elasticsearch_requests_failed_total`	Число неудачных обращений за выполнение одной Sidekiq-задачи (ошибки подключения или авторизации) — **добавлено в Deckhouse Code**

Индексатор репозиториев (`Search::RepositoryIndexerWorker` — код, коммиты, wiki):

Метрика	Лейблы	Описание
`search_repository_indexer_starts_total`	`indexer_class`	Число запусков индексации после прохождения проверки `advanced_search_enabled`
`search_repository_indexer_runs_total`	`outcome`, `indexer_class`	Число завершённых запусков после получения exclusive lock (`outcome`: `success` или `error`)
`search_repository_indexer_duration_seconds`	`outcome`, `indexer_class`	Длительность фазы индексации под exclusive lock
`search_repository_indexer_lock_contention_total`	—	Число случаев, когда lock не получен и задача перенесена

Значение `indexer_class` — тип выполняемой индексации:

`indexer_class`	Когда используется
`Search::RepositoryIndexer::IncrementalIndexService`	Инкрементальная индексация после изменений в репозитории
`Search::RepositoryIndexer::FullIndexService`	Полная переиндексация (force)
`Search::RepositoryIndexer::MaintainsService`	Обновление индекса по событию
`Search::RepositoryIndexer::DeleteService`	Удаление документов из индекса (пустой или удалённый репозиторий/wiki)

Рост `search_repository_indexer_lock_contention_total` — признак конкуренции за lock между задачами одного проекта. Рост `search_repository_indexer_runs_total{outcome="error"}` — ошибки go-indexer или сервисов индексации; детали в логах Sidekiq.

Метрики `*_failed_total` увеличиваются при ошибках подключения к OpenSearch или ошибках авторизации. Рост `*_failed_total` указывает на недоступность OpenSearch или на использование неверных данных для доступа. Рост `*_duration_seconds` при стабильном `*_total` — на медленные ответы OpenSearch.

Для мониторинга индексации репозиториев ориентируйтесь на `search_repository_indexer_*`, для обращений к OpenSearch из Sidekiq — на `sidekiq_elasticsearch_*`. Для мониторинга пользовательского поиска — на `http_elasticsearch_*`.

На странице «Admin» → «Настройки» → «Поиск» виджет прогресса индексации показывает число оставшихся задач переиндексации. Те же данные доступны через эндпоинт `indexing_queue_stats`.

Очередь Sidekiq

Задачи индексации OpenSearch выполняются в отдельной очереди `global-search-indexing`, а не в общей очереди `default`. Маршрутизация настраивается правилом Sidekiq: все воркеры с категорией `fe_global_search` попадают в эту очередь. Отдельная очередь изолирует нагрузку индексации от остальных фоновых задач Deckhouse Code.

Cron-задачи

Для автоматизации индексации регулярно выполняются следующие cron-задачи:

Расписание	Назначение
Каждую минуту	Индексация комментариев — обрабатывает накопившуюся очередь изменений notes
Ежедневно в 03:00	Запускает индексацию проектов

Cron-задачи не выполняют индексацию напрямую: они запускают или возобновляют соответствующие воркеры в очереди `global-search-indexing`.

Логи

Задачи индексации OpenSearch пишутся в логи Sidekiq. Для фильтрации используйте имя очереди `global-search-indexing`.

Устранение неполадок

OpenSearch недоступен

- На странице «Admin» → «Настройки» → «Поиск» появится сообщение о невозможности подключения.
- Поиск вернёт ошибку.

Адрес и учётные данные OpenSearch задаются в конфигурации типа установки, а не в интерфейсе:

Пакет для ОС Linux

Ключ `gitlab_rails['fe_search']` в файле `/etc/gitlab/gitlab.rb`, применяется командой `sudo gitlab-ctl reconfigure`.

Omnibus Docker

Тот же ключ в файле `/etc/gitlab/gitlab.rb` в томе с конфигурацией, применяется командой `docker exec code gitlab-ctl reconfigure`.

Helm-чарт

Значения `global.codeHelm.search` релиза, применяются командой `helm upgrade`.

Модуль Deckhouse Kubernetes Platform

Ресурс CodeInstance, описан в разделе «Расширенный поиск» документации модуля.

Неполные результаты поиска

- Дождитесь завершения фоновой индексации (виджет прогресса на странице «Admin» → «Настройки» → «Поиск»).
- Запустите переиндексацию на уровне проекта или операцию «Переиндексировать» для нужного индекса в «Admin» → «Настройки» → «Поиск».

Задачи индексации не появляются

Если новые задачи не ставятся в очередь `global-search-indexing`:

1. Проверьте, не включена ли пауза (включён флаг «Приостановить индексирование OpenSearch») в «Admin» → «Настройки» → «Поиск». Снимите флаг и дождитесь возобновления (cron-задача выполняется каждые 5 минут).
2. Если пауза снята, а задачи по-прежнему не появляются, выполните очистку Redis — возможны зависшие lease или duplicate-ключи Sidekiq:

   Пакет для ОС Linux

   ```
   sudo gitlab-rails runner /opt/gitlab/embedded/service/gitlab-rails/fe/
   scripts/clear_search_opensearch_worker_redis.rb
   ```

   Omnibus Docker

   ```
   docker exec code gitlab-rails runner /opt/gitlab/embedded/service/gitlab-
   rails/fe/scripts/clear_search_opensearch_worker_redis.rb
   ```

Helm-чарт

```
d8 k -n code exec deploy/code-toolbox -- gitlab-rails runner /srv/gitlab/fe/scripts/clear_search_opensearch_worker_redis.rb
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rails runner /srv/gitlab/fe/scripts/clear_search_opensearch_worker_redis.rb
```

Скрипт снимает exclusive lease для `Search::RepositoryIndexerWorker`, concurrency limit и dedup-ключи очереди `global-search-indexing`.

API OpenSearch

В этом разделе описаны административные OpenSearch-эндпоинты Deckhouse Code. Параметры пользовательского поиска — [в разделе «API поиска»](#).

POST /api/v4/admin/opensearch/recreate_indices

Синхронно пересоздаёт индекс(ы) OpenSearch и ставит фоновые задачи повторной индексации. Чтобы пересоздать (переиндексировать) все индексы, выполните запрос без тела. Чтобы пересоздать конкретный индекс, укажите его в теле запроса.

Права доступа: только администратор (`authenticated_as_admin!`).

Тело запроса

Поле	Тип	Обязательное	Допустимые значения
`schema_class`	string	Да	`recreate_all`, `Search::Opensearch::IndicesSchema::Code`, `Search::Opensearch::IndicesSchema::Wiki`, `Search::Opensearch::IndicesSchema::Note`, `Search::Opensearch::IndicesSchema::Milestone`, `Search::Opensearch::IndicesSchema::WorkItem`, `Search::Opensearch::IndicesSchema::MergeRequest`

Ответы

Тексты полей `message` в примерах ниже возвращаются API на английском языке.

- `202 Accepted`

```
{
  "message": "OpenSearch indices were reset; reindex jobs were enqueued."
}
```

- `400 Bad Request` (например, OpenSearch выключен или сервис вернул ошибку)

```
{
  "message": "OpenSearch is disabled"
}
```

Пример запроса

Этот запрос пересоздает все индексы:

```
curl --request POST \
  --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
  --header "Content-Type: application/json" \
  --data '{"schema_class":"recreate_all"}' \
  --url "https://gitlab.example.com/api/v4/admin/opensearch/recreate_indices"
```

GET /api/v4/admin/opensearch/indexing_queue_stats

Возвращает статистику Sidekiq-очереди индексации OpenSearch.

Права доступа: пользователь с правом `read_admin_search_indexing_queue_stats` на `:global`.

Ответ (200 OK)

```
{
  "total": 42,
  "updated_at": "2026-07-01T12:34:56.789Z"
}
```

Поля ответа:

- `total` — общее количество задач индексации в очереди;
- `updated_at` — timestamp ISO8601 с миллисекундами (или `null`).

Пример запроса

```
curl --request GET \
  --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
  --url "https://gitlab.example.com/api/v4/admin/opensearch/indexing_queue_stats"
```

Relay (KAS)

Relay (KAS) — сервер, к которому подключаются агенты для Kubernetes, установленные в кластерах. Пока Relay не настроен, диалог «Подключение кластера Kubernetes» на странице «Кластеры Kubernetes» проекта вместо формы регистрации агента показывает сообщение «В вашем инстансе не настроен GitLab Relay (KAS). Попросите администратора Deckhouse Code установить его».

Включение Relay

В пакете для ОС Linux и в Omnibus Docker Relay установлен и запущен по умолчанию. В Helm-чарте Relay не разворачивается, пока не включён в значениях.

Пакет для ОС Linux

Relay работает как служба `gitlab-kas` и не требует настройки. Чтобы отключить его, укажите в `/etc/gitlab/gitlab.rb`:

```
gitlab_kas['enable'] = false
```

Примените изменение:

```
sudo gitlab-ctl reconfigure
```

Omnibus Docker

Relay работает как служба `gitlab-kas` внутри контейнера и не требует настройки. Чтобы отключить его, добавьте `gitlab_kas['enable'] = false` в файл `/srv/code/config/gitlab.rb` на хосте и примените изменение:

```
docker exec code gitlab-ctl reconfigure
```

Ключ, заданный в `GITLAB_OMNIBUS_CONFIG`, вступает в силу только после пересоздания контейнера с теми же томами. Оба места описаны на странице «Настройка» раздела аутентификации.

Helm-чарт

Включите Relay в `values.yaml`:

```
global:
  kas:
    enabled: true
```

Примените значения к релизу:

```
helm upgrade code oci://<CHART_REGISTRY>/code \
  --version <CHART_VERSION> \
  -n code \
  -f values.yaml
```

Адрес подключения

Агенты подключаются к Relay по протоколу WebSocket по его внешнему адресу.

Пакет для ОС Linux

По умолчанию Relay отвечает на адресе инстанса по пути `/-/kubernetes-agent/`. При `external_url 'https://code.example.com'` адрес Relay — `wss://code.example.com/-/kubernetes-agent/`; при адресе инстанса со схемой `http` схема адреса — `ws`.

Чтобы Relay отвечал на отдельном хосте, укажите его в `/etc/gitlab/gitlab.rb`:

```
gitlab_kas_external_url 'wss://kas.example.com/'
```

Адрес принимает только схемы `ws` и `wss`. Для адреса со схемой `wss` поместите сертификат и ключ хоста в `/etc/gitlab/ssl/kas.example.com.crt` и `/etc/gitlab/ssl/kas.example.com.key` и создайте DNS-запись для `kas.example.com`, указывающую на сервер. Если включён Let's Encrypt и этих файлов нет, хост Relay добавляется в сертификат Let's Encrypt инстанса. Затем выполните `sudo gitlab-ctl reconfigure`.

Omnibus Docker

Адрес задаётся теми же ключами, что и в пакете для ОС Linux, в файле `/srv/code/config/gitlab.rb` на хосте. С томами из быстрого старта сертификат и ключ отдельного хоста Relay размещаются в `/srv/code/config/ssl/` на хосте. Изменение вступает в силу после `docker exec code gitlab-ctl reconfigure`; ключ, заданный в `GITLAB_OMNIBUS_CONFIG`, вступает в силу только после пересоздания контейнера.

Helm-чарт

Relay отвечает на отдельном хосте `kas.<domain>`, где `<domain>` — значение `global.hosts.domain`. Другое имя задаётся в `global.hosts.kas.name`. Создайте для этого хоста DNS-запись, указывающую на тот же адрес, что и веб-интерфейс.

В режиме `certManager` параметра `global.codeHelm.ingress.tls.mode` чарт выпускает сертификат хоста в Secret `code-kas-tls`. В режиме `manual` хост Relay использует Secret, указанный в `gitlab.kas.ingress.tls.secretName`, или Secret из `global.ingress.tls.secretName`, если первое значение пустое; сертификат в этом Secret покрывает имя хоста Relay. Значения описаны на странице «Расширенная настройка».

Проверка Relay

На странице «Admin» → «Обзор инстанса» → «Дашборд» блок «Компоненты» показывает строку «GitLab KAS» с версией Relay. Пока Relay отключён, строки нет. Значение «Неизвестно» означает, что инстанс не получил версию от Relay.

Проверьте состояние Relay:

Пакет для ОС Linux

```
sudo gitlab-ctl status gitlab-kas
sudo gitlab-ctl tail gitlab-kas
```

Журналы службы находятся в `/var/log/gitlab/gitlab-kas/`.

Omnibus Docker

```
docker exec code gitlab-ctl status gitlab-kas
docker exec code gitlab-ctl tail gitlab-kas
```

Helm-чарт

```
d8 k -n code get pods -l app=kas
```

Авторизация агентов на уровне инстанса

Раздел `ci_access` файла конфигурации агента `.gitlab/agents/<AGENT_NAME>/config.yaml` в проекте агента, где `<AGENT_NAME>` — имя агента, перечисляет проекты и группы, задания CI/CD которых могут использовать агент; каждый список принимает до 500 записей. По умолчанию раздел может перечислять проекты и группы внутри группы верхнего уровня, в которую входит проект агента; если проект агента находится в личном неймспейсе, — только проекты этого неймспейса. С авторизацией на уровне инстанса раздел может перечислять проекты и группы из любых групп верхнего уровня, а его ключ `instance` авторизует агент для всех проектов инстанса.

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните раздел «GitLab Agent для Kubernetes».
3. Установите флажок «Включить авторизацию на уровне инстанса».
4. Нажмите «Сохранить изменения».

Агенты для Kubernetes

Агент для Kubernetes (`agentk`) работает в кластере и подключает его к Relay. Агент разворачивает в кластере пользователь, который его регистрирует, или администратор Deckhouse Code; установки Deckhouse Code агенты не разворачивают.

После регистрации агента диалог «Подключение кластера Kubernetes» показывает команду `helm upgrade --install` , которая устанавливает агент из чарта `gitlab/gitlab-agent` . Deckhouse Code предоставляет образ агента `registry.deckhouse.ru/deckhouse/ee/packages/code/images/agentk` . Чтобы установить агент из этого образа, добавьте в команду из диалога строку `--set image.repository` :

```
helm repo add gitlab https://charts.gitlab.io
helm repo update
helm upgrade --install <AGENT_NAME> gitlab/gitlab-agent \
    --namespace gitlab-agent-<AGENT_NAME> \
    --create-namespace \
    --set image.repository=registry.deckhouse.ru/deckhouse/ee/packages/code/images/
agentk \
    --set image.tag=v<VERSION> \
    --set config.token=<TOKEN> \
    --set config.kasAddress=<RELAY_ADDRESS>
```

Значения `<AGENT_NAME>` , `<VERSION>` , `<TOKEN>` и `<RELAY_ADDRESS>` берутся из команды в диалоге.

Резервное копирование и восстановление

Резервная копия Deckhouse Code состоит из архива с данными инстанса и файлов, которые в архив не попадают. Состав архива одинаков для пакета для ОС Linux и Omnibus Docker: обе поставки работают на одном движке, и копию в них создаёт команда `gitlab-backup` . В контейнере те же команды выполняются через `docker exec` , а архив записывается на том данных.

Что входит в резервную копию

Команда `gitlab-backup create` собирает в один архив:

- репозитории;
- базу данных;
- загруженные файлы (uploads);
- артефакты CI;
- LFS-объекты;
- реестр пакетов.

Что хранится отдельно

Эти файлы в архив не попадают, копируйте их вместе с каждым архивом:

- `/etc/gitlab/gitlab-secrets.json` — ключи шифрования базы данных, без которых при восстановлении теряются настройки двухфакторной аутентификации, токены и зашифрованные переменные CI;
- `/etc/gitlab/gitlab.rb` — конфигурация инстанса;
- сертификаты и ключи из каталога `/etc/gitlab/ssl/`.

Файлы секретов храните с тем же уровнем защиты, что и пароли.

Пакет для ОС Linux

Файлы лежат в каталоге `/etc/gitlab/` на машине инстанса. Копируйте их вместе с каждым архивом.

Omnibus Docker

Файлы находятся на томе конфигурации. Архивируйте их отдельной командой:

```
docker exec -t code gitlab-ctl backup-etc
```

Команда записывает файл `/etc/gitlab/config_backup/gitlab_config_<TIMESTAMP>_<DATE>.tar` с содержимым каталога `/etc/gitlab`: `gitlab.rb`, `gitlab-secrets.json` и TLS-сертификатами. Каталог находится на томе конфигурации, на хосте это `/srv/code/config/config_backup`.

Создание резервной копии

Пакет для ОС Linux

Копия снимается одной командой:

```
sudo gitlab-backup create
```

Архив появится в каталоге `/var/opt/gitlab/backups/` под именем вида `<TIMESTAMP>_<DATE>_<ENGINE_VERSION>_gitlab_backup.tar`. Версия в имени — версия встроенного движка, она не совпадает с версией пакета `deckhouse-code`, поэтому при снятии копии записывайте версию пакета рядом с архивом:

```
# РЕД ОС, AlmaLinux.
rpm -q deckhouse-code
# Ubuntu.
dpkg -l deckhouse-code
```

Omnibus Docker

Выполните создание копии в работающем контейнере:

```
docker exec -t code gitlab-backup create
```

Архив появится в каталоге `/var/opt/gitlab/backups` внутри контейнера, с томами из раздела «Быстрый старт» на хосте это `/srv/code/data/backups`. Копируйте его за пределы хоста вместе с архивом конфигурации и секретов.

Helm-чарт

Утилита `backup-utility` запускается внутри пода toolbox, который разворачивает чарт:

```
d8 k -n code exec deploy/code-toolbox -- backup-utility
```

Команда создаёт архив резервной копии и загружает его в бакет объектного хранилища, настроенный через `global.appConfig.object_store` (раздел «Объектное хранилище»).

Срок хранения архивов и расписание

Архивы старше `backup_keep_time` удаляются при создании следующей резервной копии.

Пакет для ОС Linux

Срок хранения задаётся в файле `/etc/gitlab/gitlab.rb` в секундах:

```
# Семь дней.
gitlab_rails['backup_keep_time'] = 604800
```

Изменение вступает в силу после `sudo gitlab-ctl reconfigure`.

Регулярное снятие копии задаётся задачей cron, например ежедневно в 02:00:

```
echo '0 2 * * * root /opt/gitlab/bin/gitlab-backup create CRON=1' \
  | sudo tee /etc/cron.d/deckhouse-code-backup
```

Omnibus Docker

Срок хранения архивов задаёт параметр `gitlab_rails['backup_keep_time']` в конфигурации на томе конфигурации; его значение и поведение описаны на вкладке «Пакет для ОС Linux».

Для расписания выполняйте ту же команду из `cron` на хосте:

```
echo '0 2 * * * root /usr/bin/docker exec -t code gitlab-backup create CRON=1' \
  | sudo tee /etc/cron.d/deckhouse-code-backup
```

Копирование за пределы машины

Регулярно копируйте свежие архивы на отдельный сервер.

Пакет для ОС Linux

Забирайте архивы из каталога `/var/opt/gitlab/backups/`, например командой `rsync`. Вместе с архивом копируйте файлы `/etc/gitlab/gitlab-secrets.json` и `/etc/gitlab/gitlab.rb`: без них архив восстанавливается не полностью.

Omnibus Docker

Забирайте архивы из каталога `/srv/code/data/backups` на хосте вместе с архивом конфигурации из каталога `/srv/code/config/config_backup`: без конфигурации и секретов архив восстанавливается не полностью.

Восстановление

Восстановление заменяет данные инстанса содержимым архива. До запуска проверьте, что выполнены условия:

- установлена ровно та версия, на которой сделана резервная копия;
- файлы `gitlab.rb` и `gitlab-secrets.json` исходного инстанса лежат на своих местах. При переносе на новый сервер скопируйте их до восстановления и выполните настройку;
- сертификаты и ключи из каталога `/etc/gitlab/ssl/` лежат на своих местах: `gitlab.rb` ссылается на них по путям, и без этих файлов настройка завершается ошибкой запуска nginx;
- на диске достаточно свободного места для распакованного архива.

В командах для пакета для ОС Linux и Omnibus Docker `<BACKUP_NAME>` — имя файла архива без суффикса `_gitlab_backup.tar` : для архива `1784800000_2026_07_23_19.1.2_gitlab_backup.tar` это `1784800000_2026_07_23_19.1.2` .

Пакет для ОС Linux

Проверьте установленную версию пакета командой `rpm -q deckhouse-code` на РЕД ОС и AlmaLinux или `dpkg -l deckhouse-code` на Ubuntu. Если конфигурация и секреты перенесены с другой машины, после копирования выполните `sudo gitlab-ctl reconfigure`.

1. Скопируйте архив в каталог резервных копий и передайте файл пользователю `git`:

   ```
   sudo cp <BACKUP_NAME>_gitlab_backup.tar /var/opt/gitlab/backups/
   sudo chown git:git /var/opt/gitlab/backups/<BACKUP_NAME>_gitlab_backup.tar
   ```

2. Остановите сервисы, которые пишут в базу данных; остальные сервисы продолжают работать:

   ```
   sudo gitlab-ctl stop puma
   sudo gitlab-ctl stop sidekiq
   ```

3. Восстановите данные из архива:

   ```
   sudo gitlab-backup restore BACKUP=<BACKUP_NAME>
   ```

   Команда дважды запросит подтверждение: перед пересозданием файла `authorized_keys` и перед удалением текущих таблиц базы данных. Оба раза ответьте `yes`.

4. Перезапустите инстанс и выполните самопроверку:

   ```
   sudo gitlab-ctl reconfigure
   sudo gitlab-ctl restart
   sudo gitlab-rake gitlab:check SANITIZE=true
   ```

Omnibus Docker

Проверьте, что контейнер запущен с тегом образа той версии, на которой сделана копия, и что файлы конфигурации и секретов исходного инстанса лежат на томе конфигурации.

1. Скопируйте архив в каталог резервных копий на томе данных и назначьте ему владельца, под которым работает инстанс:

   ```
   sudo cp <BACKUP_NAME>_gitlab_backup.tar /srv/code/data/backups/
   docker exec code chown git:git /var/opt/gitlab/backups/
   <BACKUP_NAME>_gitlab_backup.tar
   ```

2. Остановите сервисы, которые пишут в базу данных:

   ```
   docker exec code gitlab-ctl stop puma
   docker exec code gitlab-ctl stop sidekiq
   ```

3. Запустите восстановление. Команда дважды запросит подтверждение — перед пересозданием файла `authorized_keys` и перед удалением текущих таблиц базы данных; оба раза ответьте `yes`:

   ```
   docker exec -it code gitlab-backup restore BACKUP=<BACKUP_NAME>
   ```

4. Примените конфигурацию и перезапустите контейнер:

   ```
   docker exec code gitlab-ctl reconfigure
   docker restart code
   ```

5. Проверьте инстанс:

   ```
   docker exec code gitlab-ctl status
   docker exec code gitlab-rake gitlab:check SANITIZE=true
   ```

Helm-чарт

1. Переведите инстанс в режим обслуживания.
2. Убедитесь, что бакет объектного хранилища, настроенный через `global.appConfig.object_store`, содержит архив, который нужно восстановить.
3. Выполните команду восстановления внутри пода toolbox:

   ```
   d8 k -n code exec deploy/code-toolbox -- backup-utility restore --backup-id <BACKUP_ID>
   ```

4. Дождитесь завершения команды и убедитесь, что все поды находятся в состоянии `Running`:

   ```
   d8 k -n code get pods
   ```

5. Выведите инстанс из режима обслуживания.

После восстановления войдите в веб-интерфейс и склонируйте репозиторий, чтобы убедиться, что инстанс работает со своими данными.

Модуль Deckhouse Kubernetes Platform

В поставке модулем резервные копии создаёт оператор по расписанию из ресурса CodeInstance и выгружает их в объектное хранилище: состав архива, предусловия восстановления и матрица команд приведены в разделе «Бекапы и восстановление» документации модуля.

Обновление

Обновление Deckhouse Code устанавливает новую версию поверх текущей: конфигурация и данные сохраняются, настройка и миграции базы данных запускаются автоматически. Правила ниже действуют для всех типов установки.

Порядок версий

Обновляйтесь последовательно, без пропуска минорных версий: миграции базы данных рассчитаны на переход с предыдущей минорной версии. Например, обновление с 1.14 до 1.16 проходит через 1.15: установите 1.15, затем 1.16. Если между текущей и целевой версией несколько минорных релизов, устанавливайте их по очереди и дожидайтесь между шагами завершения фоновых миграций. Путь обновления для вашей пары версий уточните у вендора при получении новой версии.

Фоновые миграции

Обновление применяет основные миграции базы данных и не дожидается фоновых: они продолжают выполняться на уже работающем инстансе. Перед следующим обновлением все фоновые миграции должны завершиться, иначе обновление прервётся с ошибкой.

Чтобы проверить состояние миграций, в правом верхнем углу выберите «Admin», затем в левой панели перейдите в «Мониторинг» → «Фоновые миграции». Все миграции должны быть в конечном статусе: в списке не осталось задач в очереди и в процессе завершения, а задач, завершившихся с ошибкой, нет. На типовом инстансе это занимает минуты, на больших объёмах данных — дольше.

Перед обновлением

Пакет для ОС Linux

Создайте резервную копию по разделу «Резервное копирование и восстановление» и проверьте, что есть свежие копии файлов `/etc/gitlab/gitlab-secrets.json` и `/etc/gitlab/gitlab.rb`:

```
sudo gitlab-backup create
```

Omnibus Docker

Создайте резервную копию данных и конфигурации по разделу «Резервное копирование и восстановление» и убедитесь, что архивы хранятся за пределами хоста.

Запишите тег, с которым запущен контейнер: к нему возвращает откат.

```
docker inspect --format '{{.Config.Image}}' code
```

Helm-чарт

Каждый релиз чарта указывает версию Deckhouse Code, которую он разворачивает. Проверьте её перед обновлением:

```
helm show chart deckhouse-code --version <CHART_VERSION>
```

Создайте резервную копию по разделу «Резервное копирование и восстановление»: она нужна для отката изменения схемы базы данных.

Обновление инстанса

Пакет для ОС Linux

Конфигурация в файле `/etc/gitlab/gitlab.rb` и данные сохраняются.

1. Установите новый пакет; настройка и миграции запустятся автоматически:

```
# РЕД ОС.
sudo rpm -Uvh ./deckhouse-code-<NEW_VERSION>.el8.x86_64.rpm
# AlmaLinux.
sudo rpm -Uvh ./deckhouse-code-<NEW_VERSION>.el9.x86_64.rpm
# Ubuntu.
sudo apt install -y ./deckhouse-code_<NEW_VERSION>_amd64.deb
```

2. Проверьте состояние инстанса и установленную версию:

```
# Все сервисы в состоянии run.
sudo gitlab-ctl status
# Установленная версия на РЕД ОС и AlmaLinux.
rpm -q deckhouse-code
# Установленная версия на Ubuntu.
dpkg -l deckhouse-code
# Ожидается health=200.
curl --cacert /etc/gitlab/ssl/<HOSTNAME>.crt \
     --resolve '<HOSTNAME>:443:127.0.0.1' -o /dev/null \
     -w "health=%{http_code}\n" https://<HOSTNAME>/-/health
```

На время миграций инстанс отвечает медленнее; на типовом инстансе обновление занимает минуты.

Omnibus Docker

Контейнер заменяется контейнером из нового тега образа; тома остаются, а данные и конфигурация инстанса на них переносятся при первом запуске. Каждый тег называет версию, отдельного тега для последней версии нет, поэтому целевая версия указывается явно.

1. Получите образ целевой версии:

   ```
   docker pull <REGISTRY>/<FLAVOR>:<VERSION>
   ```

2. Остановите и удалите контейнер. Тома — это каталоги хоста, они остаются на месте:

   ```
   docker stop code
   docker rm code
   ```

3. Создайте контейнер из нового тега с тем же именем, портами, томами и переменными окружения, что и в разделе «Быстрый старт»:

   ```
   docker run -d --name code \
     --shm-size 256m \
     -p 80:80 -p 443:443 -p 22:22 \
     -e GITLAB_OMNIBUS_CONFIG="external_url 'http://<HOSTNAME>'" \
     -v /srv/code/config:/etc/gitlab \
     -v /srv/code/logs:/var/log/gitlab \
     -v /srv/code/data:/var/opt/gitlab \
     <REGISTRY>/<FLAVOR>:<VERSION>
   ```

4. Следите за запуском. Миграции базы данных выполняются во время настройки, и до её окончания инстанс отвечает 502:

   ```
   docker logs -f code
   ```

5. Проверьте версию и сервисы:

   ```
   docker logs code | grep 'Current version'
   docker exec code gitlab-ctl status
   docker inspect --format '{{.State.Health.Status}}' code
   ```

Стартовый скрипт читает версию данных на томе и сравнивает её с версией образа. Если перейти к новой версии за один шаг нельзя, в выводе указывается версия, которую нужно установить сначала, и запуск прекращается: запустите тег этой версии, дождитесь завершения фоновых миграций и затем запустите целевой тег.

После настройки скрипт приводит базу данных к версии PostgreSQL из нового образа. Неудачное обновление базы отменяется, и контейнер завершает работу с сообщением `Upgrading the existing database failed and was reverted.` Создайте контейнер с параметром `-e GITLAB_SKIP_PG_UPGRADE=true`, чтобы запустить инстанс на текущей версии PostgreSQL, и обновите базу данных отдельно.

Helm-чарт

Обновите релиз:

```
helm repo update
helm upgrade code deckhouse-code \
  -n code \
  -f values.yaml \
  --version <CHART_VERSION>
```

Команда переиспользует существующий `values.yaml`. Если по таблице соответствия версий нужно пройти через несколько версий Deckhouse Code, соблюдайте порядок версий: пропуск версии может привести к тому, что более поздняя миграция не сможет выполниться.

При каждом обновлении один раз выполняется задача миграций — перед перезапуском компонентов, которые зависят от новой схемы базы данных. Проверьте её ход:

```
d8 k -n code get jobs
d8 k -n code logs job/<MIGRATIONS_JOB_NAME>
```

Фоновые миграции продолжаются на работающем инстансе после обновления и должны завершиться до следующего обновления.

Откат

Откат возможен только из резервной копии, снятой перед обновлением: новая версия перенесла данные.

Пакет для ОС Linux

Установите пакет той версии, на которой сделана копия, и восстановите данные по разделу «Восстановление». Точную процедуру понижения версии для вашей конфигурации уточните у вендора.

Omnibus Docker

1. Остановите и удалите контейнер:

   ```
   docker stop code
   docker rm code
   ```

2. Создайте контейнер из предыдущего тега с теми же томами.
3. Восстановите данные из резервной копии, сделанной перед обновлением, по разделу «Восстановление».

Helm-чарт

Откатите релиз на предыдущую ревизию:

```
helm rollback code <REVISION>
```

> ⚠ Команда `helm rollback` откатывает только объекты Kubernetes релиза. Она не отменяет изменение схемы базы данных, сделанное миграциями обновления.

Если обновление, которое вы откатываете, изменило схему, после завершения отката восстановите резервную копию, созданную до обновления, по разделу «Восстановление».

Модуль Deckhouse Kubernetes Platform

В поставке модулем обновление запускает платформа. Если в ресурсе CodeInstance заданы `backup.enabled` и `backup.backupBeforeUpdate`, оператор сначала создаёт резервную копию и обновляет остальные компоненты после успешного завершения задачи копирования; неуспешная задача откладывает обновление и вызывает оповещение `D8CodeOperatorUpdatePostpone`. Порядок описан в разделе «Автоматическое создание бэкапов перед обновлениями модуля» документации модуля.

Раннеры

> ⚠ Deckhouse Code поставляет два образа раннера: `registry.deckhouse.ru/deckhouse/ee/packages/code/images/runner` и `registry.deckhouse.ru/deckhouse/ee/packages/code/images/runner-helper`. Развёртывание и эксплуатацию раннеров (серверы, манифесты, Helm-чарты, обновление, мониторинг) выполняет администратор.

Раннер — процесс GitLab Runner, который получает задания CI/CD от Deckhouse Code и выполняет их в контейнерах, подах Kubernetes или виртуальных машинах в зависимости от своего executor.

Образы раннера

Тег образа — версия GitLab Runner, например `v19.3.1`. Оба образа собраны только для `linux/amd64`.

Образ	Содержимое
`runner`	GitLab Runner, плагин `fleeting-plugin-dvp` в `/usr/local/bin`. Процесс работает от пользователя `gitlab-runner` (UID и GID `1001`) и читает конфигурацию из `/etc/gitlab-runner/config.toml`
`runner-helper`	Вспомогательный процесс, который Docker executor и Kubernetes executor запускают рядом с каждым заданием, чтобы клонировать репозиторий и загрузить артефакты и кеш; содержит `git`, `git-lfs` и `bash`

Docker executor и Kubernetes executor берут вспомогательный образ из параметра `helper_image`. В примерах ниже в нём указан образ `runner-helper` той же версии, что и раннер.

Создание раннера

Раннер подключается к Deckhouse Code с токеном аутентификации. Чтобы создать раннер инстанса и получить его токен:

1. Откройте «Admin» → «CI/CD» → «Раннеры» и нажмите «Создать раннер инстанса».
2. В поле «Теги» перечислите теги заданий, которые выполняет раннер, или выберите «Запуск заданий без тегов».
3. Нажмите «Создать раннер».
4. Скопируйте токен аутентификации раннера (начинается с `glrt-`) на открывшейся странице.

В примерах ниже `<TOKEN>` — этот токен, `<VERSION>` — тег образа, `https://code.example.com` — адрес Deckhouse Code.

Автомасштабирование на Deckhouse Virtualization Platform

Плагин `fleeting-plugin-dvp` позволяет executor `instance` выполнять задания на виртуальных машинах из VirtualMachinePool в Deckhouse Virtualization Platform. Раннер задаёт число виртуальных машин в пуле (`spec.replicas`): по одной на задание плюс резерв простаивающих.

Он подключается к виртуальной машине по SSH и выполняет задание в её оболочке. Ненужную виртуальную машину плагин удаляет и уменьшает `spec.replicas`.

Плагин не создаёт пул и пользователя на виртуальных машинах: пул и доступ по SSH подготавливаются заранее. В примере ниже раннер работает как Deployment в кластере Deckhouse Virtualization Platform, в неймспейсе `code-runner-dvp` вместе с пулом.

Неймспейс и SSH-ключ

Создайте неймспейс, пару ключей и Secret с закрытым ключом:

```
d8 k create namespace code-runner-dvp
ssh-keygen -t ed25519 -N "" -f code-runner-ssh
d8 k -n code-runner-dvp create secret generic code-runner-ssh --from-file=id_ed25519=c
ode-runner-ssh
```

Открытый ключ `code-runner-ssh.pub` указывается в манифесте пула.

Пул виртуальных машин

Пул создаётся пустым и удаляет виртуальные машины по имени (`scaleDownPolicy: Explicit`), как это делает плагин при уменьшении пула. В образе диска нужны SSH-сервер, `bash` и `git`; cloud-init создаёт пользователя, от которого подключается раннер. Сохраните манифест в файл `code-runner-pool.yaml`:

```
apiVersion: virtualization.deckhouse.io/v1alpha2
kind: VirtualMachinePool
metadata:
  name: code-runner
  namespace: code-runner-dvp
spec:
  replicas: 0
  scaleDownPolicy: Explicit
  virtualMachineTemplate:
    spec:
      virtualMachineClassName: generic
      runPolicy: AlwaysOn
      cpu:
        cores: 2
      memory:
        size: 2Gi
      blockDeviceRefs:
        - kind: VirtualDisk
          name: disk
      provisioning:
        type: UserData
        userData: |
          #cloud-config
          package_update: true
          packages:
            - git
          users:
            - name: gitlab-runner
              shell: /bin/bash
              sudo: ALL=(ALL) NOPASSWD:ALL
              lock_passwd: false
              ssh_authorized_keys:
                - <PUBLIC_KEY>
  virtualDiskTemplates:
    - name: disk
      spec:
        dataSource:
          type: ObjectRef
          objectRef:
            kind: VirtualImage
            name: ubuntu-2204
        persistentVolumeClaim:
          size: 20Gi
          storageClassName: <STORAGE_CLASS_NAME>
```

`<PUBLIC_KEY>` — содержимое `code-runner-ssh.pub`; `ubuntu-2204` — VirtualImage в том же неймспейсе. Задания записывают на диск рабочие копии и кеш, поэтому размер диска ограничивает размер задания.

Создайте пул:

```
d8 k apply -f code-runner-pool.yaml
```

Развёртывание раннера

1. Сохраните следующий манифест в файл `code-runner-dvp.yaml`. Он содержит конфигурацию раннера, сервисный аккаунт с правами на пул и Deployment раннера со смонтированным закрытым ключом:

```
apiVersion: v1
kind: Secret
metadata:
  name: code-runner-config
  namespace: code-runner-dvp
stringData:
  config.toml: |
    concurrent = 4

    [[runners]]
      name = "dvp"
      url = "https://code.example.com"
      token = "<TOKEN>"
      executor = "instance"
      shell = "bash"

      [runners.autoscaler]
        plugin = "fleeting-plugin-dvp"
        max_instances = 5

        [runners.autoscaler.plugin_config]
          namespace = "code-runner-dvp"
          name = "code-runner"

        [runners.autoscaler.connector_config]
          username = "gitlab-runner"
          key_path = "/ssh/id_ed25519"
          use_static_credentials = true
          use_external_addr = false

        [[runners.autoscaler.policy]]
          idle_count = 1
          idle_time = "20m"
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: code-runner
  namespace: code-runner-dvp
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: code-runner
  namespace: code-runner-dvp
rules:
  - apiGroups: ["virtualization.deckhouse.io"]
    resources: ["virtualmachinepools"]
```

```
    verbs: ["get", "patch"]
  - apiGroups: ["virtualization.deckhouse.io"]
    resources: ["virtualmachines"]
    verbs: ["get", "list", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: code-runner
  namespace: code-runner-dvp
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: code-runner
subjects:
  - kind: ServiceAccount
    name: code-runner
    namespace: code-runner-dvp
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: code-runner
  namespace: code-runner-dvp
spec:
  replicas: 1
  selector:
    matchLabels:
      app: code-runner
  template:
    metadata:
      labels:
        app: code-runner
    spec:
      serviceAccountName: code-runner
      securityContext:
        runAsUser: 1001
        runAsGroup: 1001
        fsGroup: 1001
      initContainers:
        - name: config
          image: registry.deckhouse.ru/deckhouse/ee/packages/code/images/
runner:<VERSION>
          command: ["cp", "/config-secret/config.toml", "/etc/gitlab-runner/
config.toml"]
          volumeMounts:
            - name: config-secret
              mountPath: /config-secret
            - name: config
```

```
              mountPath: /etc/gitlab-runner
      containers:
        - name: runner
          image: registry.deckhouse.ru/deckhouse/ee/packages/code/images/
runner:<VERSION>
          volumeMounts:
            - name: config
              mountPath: /etc/gitlab-runner
            - name: ssh
              mountPath: /ssh
      volumes:
        - name: config-secret
          secret:
            secretName: code-runner-config
        - name: config
          emptyDir: {}
        - name: ssh
          secret:
            secretName: code-runner-ssh
            defaultMode: 0440
```

Параметры, которые связывают раннер с пулом:

- `plugin_config.namespace` и `plugin_config.name` — неймспейс и имя VirtualMachinePool; без `kubeconfig_path` плагин использует сервисный аккаунт пода;
- `connector_config.username` — пользователь, которого создаёт cloud-init;
- `connector_config.key_path` — закрытый ключ, открытая часть которого указана в `ssh_authorized_keys`;
- `use_external_addr = false` — раннер подключается к внутреннему адресу виртуальной машины (`status.ipAddress`);
- `shell = "bash"` — в Debian и Ubuntu `/bin/sh` — это `dash`, и скрипт подготовки задания в нём завершается с ошибкой;
- `max_instances` — верхний предел числа виртуальных машин в пуле; `idle_count` и `idle_time` задают, сколько свободных виртуальных машин держать и как долго.

Role разрешает плагину читать и изменять пул, читать и удалять его виртуальные машины. Init-контейнер копирует `config.toml` из Secret в доступную для записи директорию, где раннер хранит и свой системный идентификатор. `defaultMode: 0440` разрешает чтение ключа группе `gitlab-runner`, которую `fsGroup` назначает файлу ключа.

2. Примените манифест:

```
d8 k apply -f code-runner-dvp.yaml
```

3. Проверьте подключение к Deckhouse Code:

```
d8 k -n code-runner-dvp exec deploy/code-runner -- gitlab-runner verify
```

Раннер вне кластера обращается к пулу через файл kubeconfig, указанный в `kubeconfig_path`; его пользователю нужны права из Role выше.

Перезапущенный раннер заменяет виртуальные машины пула новыми. Чтобы он подхватывал работающие виртуальные машины, включите хранилище состояния автомасштабирования в директории на постоянном томе, смонтированном в контейнер раннера:

```
[runners.autoscaler.state_storage]
  enabled = true
  dir = "/home/gitlab-runner/state"
  keep_instance_with_acquisitions = true
```

Docker executor

Docker executor выполняет каждое задание в контейнере Docker-демона на сервере раннера. Сам раннер работает в контейнере и обращается к демону через его сокет.

1. Создайте том для конфигурации раннера:

   ```
   docker volume create code-runner-config
   ```

2. Зарегистрируйте раннер. Команда записывает `config.toml` в том:

   ```
   docker run --rm -v code-runner-config:/etc/gitlab-runner \
     registry.deckhouse.ru/deckhouse/ee/packages/code/images/runner:<VERSION> \
     register --non-interactive \
       --url https://code.example.com \
       --token <TOKEN> \
       --executor docker \
       --docker-image alpine:3 \
       --docker-helper-image registry.deckhouse.ru/deckhouse/ee/packages/code/
   images/runner-helper:<VERSION>
   ```

   `--docker-image` — образ для заданий, в `.gitlab-ci.yml` которых не указан `image`.

3. Запустите раннер:

   ```
   docker run -d --name code-runner --restart always \
     --group-add "$(stat -c %g /var/run/docker.sock)" \
     -v code-runner-config:/etc/gitlab-runner \
     -v /var/run/docker.sock:/var/run/docker.sock \
     registry.deckhouse.ru/deckhouse/ee/packages/code/images/runner:<VERSION>
   ```

Раннер работает от пользователя `gitlab-runner`; `--group-add` добавляет его в группу — владельца сокета Docker.

4. Проверьте подключение к Deckhouse Code:

```
docker exec code-runner gitlab-runner verify
```

Kubernetes executor

Kubernetes executor выполняет каждое задание в отдельном поде. Сам раннер работает как Deployment в том же кластере и создаёт поды заданий в неймспейсе, указанном в `config.toml`.

1. Создайте неймспейс:

```
d8 k create namespace code-runner
```

2. Сохраните следующий манифест в файл `code-runner.yaml`. Он содержит конфигурацию раннера, сервисный аккаунт с правами на управление подами заданий и Deployment раннера:

```
apiVersion: v1
kind: Secret
metadata:
  name: code-runner-config
  namespace: code-runner
stringData:
  config.toml: |
    concurrent = 4

    [[runners]]
      name = "kubernetes"
      url = "https://code.example.com"
      token = "<TOKEN>"
      executor = "kubernetes"
      [runners.kubernetes]
        namespace = "code-runner"
        image = "alpine:3"
        helper_image = "registry.deckhouse.ru/deckhouse/ee/packages/code/images/
runner-helper:<VERSION>"
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: code-runner
  namespace: code-runner
---
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: code-runner
  namespace: code-runner
rules:
  - apiGroups: [""]
    resources: ["pods", "pods/exec", "pods/attach", "pods/log", "secrets", "conf
igmaps", "services", "events"]
    verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
---
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: code-runner
  namespace: code-runner
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: code-runner
subjects:
  - kind: ServiceAccount
```

```
    name: code-runner
    namespace: code-runner
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: code-runner
  namespace: code-runner
spec:
  replicas: 1
  selector:
    matchLabels:
      app: code-runner
  template:
    metadata:
      labels:
        app: code-runner
    spec:
      serviceAccountName: code-runner
      securityContext:
        runAsUser: 1001
        runAsGroup: 1001
        fsGroup: 1001
      initContainers:
        - name: config
          image: registry.deckhouse.ru/deckhouse/ee/packages/code/images/
runner:<VERSION>
          command: ["cp", "/config-secret/config.toml", "/etc/gitlab-runner/
config.toml"]
          volumeMounts:
            - name: config-secret
              mountPath: /config-secret
            - name: config
              mountPath: /etc/gitlab-runner
      containers:
        - name: runner
          image: registry.deckhouse.ru/deckhouse/ee/packages/code/images/
runner:<VERSION>
          volumeMounts:
            - name: config
              mountPath: /etc/gitlab-runner
      volumes:
        - name: config-secret
          secret:
            secretName: code-runner-config
        - name: config
          emptyDir: {}
```

Init-контейнер копирует `config.toml` из Secret в доступную для записи директорию, где раннер хранит и свой системный идентификатор.

3. Примените манифест:

```
d8 k apply -f code-runner.yaml
```

4. Проверьте подключение к Deckhouse Code:

```
d8 k -n code-runner exec deploy/code-runner -- gitlab-runner verify
```

Ранний доступ

Часть функций Deckhouse Code поставляется выключенной: администратор включает каждую флагом функций уровня инстанса. В колонке «Редакция» указаны редакции, в которых включённый флаг действует; каждый флаг действует в пробном режиме, описанном на странице Лицензия.

Флаги функций

Флаг	Редакция	Что открывает	Документация
`fe_security_scan_policies`	EE	Страницы «Политики» групп и проектов, политики выполнения сканирования и обязательные сканирования (SAST с помощью Opengrep, обнаружение секретов с помощью Gitleaks, Trivy, CodeScoring, PT AI), сканирования по расписанию, интеграции CodeScoring и PT AI, страницы «Список зависимостей» и «Соответствие лицензий»	• SAST-сканирование с помощью Opengrep • Обнаружение секретов с помощью Gitleaks • Сканирование с помощью Trivy • Интеграция с CodeScoring • Интеграция с PT AI • Список зависимостей и соответствие лицензий • Интеграции со сканерами безопасности
`defect_dojo_integration`	EE	Интеграцию DefectDojo и загрузку отчётов безопасности в DefectDojo	Интеграция с DefectDojo
`fe_pipeline_execution_policies`	EE	Политики выполнения пайплайна, в том числе по расписанию. Действует только вместе с `fe_security_scan_policies`	Политики выполнения пайплайна
`fe_jaga_integration`	SE, EE	Интеграцию с Ягой, правила отправки	Интеграция с Ягой

Флаг	Редакция	Что открывает	Документация
		изменений и проверку слияния по задачам Яги	

Включение и выключение флага

Включите флаг командой `gitlab-rails runner`, заменив `<FLAG>` на имя флага:

Пакет для ОС Linux

```
sudo gitlab-rails runner 'Feature.enable(:<FLAG>)'
```

Omnibus Docker

```
docker exec code gitlab-rails runner 'Feature.enable(:<FLAG>)'
```

Helm-чарт

```
d8 k -n code exec deploy/code-toolbox -- gitlab-rails runner 'Feature.enable(:<FLAG>)'
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -c toolbox deploy/toolbox -- gitlab-rails runner 'Feature.enable(:<FLAG>)'
```

Чтобы выключить флаг, выполните ту же команду с `Feature.disable(:<FLAG>)`.

Использование

Обзор

Deckhouse Code хранит исходный код команды, обеспечивает ревью изменений и запускает CI/CD-пайплайны. Раздел «Использование» описывает повседневную работу участника проекта и владельца группы — от первого входа в систему до поиска по инстансу.

Войдите в веб-интерфейс с учётными данными своей учётной записи или через внешнего провайдера аутентификации, затем склонируйте репозиторий и отправляйте изменения по SSH или HTTPS. Подробнее — в разделе Вход в систему.

Группа объединяет проекты общей ролевой моделью и общими политиками, такими как защищённые ветки и правила отправки изменений; проект хранит репозиторий, его задачи и wiki. Подробнее — в разделе Группы и участники.

Репозиторий хранит историю Git проекта: ветки, теги и файлы, которые можно редактировать через веб-интерфейс или Git LFS. За то, как изменения попадают в репозиторий, отвечают защищённые ветки, правила отправки изменений и pull-зеркалирование. Подробнее — в разделе Управление репозиториями Git.

Запрос на слияние объединяет изменения одной ветки с другой через ревью: комментарии, правила апрувов и внешние проверки статуса определяют, когда его можно слить. Подробнее — в разделе Ревью кода.

Пайплайн, описанный в `.gitlab-ci.yml`, собирает, тестирует и разворачивает код проекта; секреты для него доступны из внешнего Vault или Deckhouse Stronghold. Подробнее — в разделе Непрерывная сборка и поставка ПО.

Владельцы групп и мейнтейнеры проектов настраивают уровни видимости, применение правил отправки изменений и проверки при слиянии, чтобы защитить кодовую базу. Подробнее — в разделе Настройки безопасности группы и проекта.

Вебхуки уведомляют внешние системы о событиях в группе, проекте или подгруппе. Подробнее — в разделе Вебхуки.

Расширенный поиск находит код, коммиты, задачи и запросы на слияние в доступных вам проектах. Подробнее — в разделе Расширенный поиск.

Начало работы

Вход в систему

1. Откройте веб-интерфейс Deckhouse Code.

Пакет для ОС Linux

Адрес — значение параметра `external_url`, заданное при установке. Подробнее — в разделе Быстрый старт.

Omnibus Docker

Адрес — значение параметра `external_url`, заданное при установке. Подробнее — в разделе Быстрый старт.

Модуль Deckhouse Kubernetes Platform

Введите в адресной строке браузера адрес веб-интерфейса. Это имя хоста, заданное для веб-интерфейса в ресурсе CodeInstance, или, если оно не задано, глобальный параметр платформы `publicDomainTemplate`, в котором `%s` заменён на `code`: для шаблона `%s.example.com` адрес — `code.example.com`.

Helm-чарт

Адрес задан в файле `values.yaml` Helm-релиза при установке. Подробнее — в разделе Быстрый старт.

2. Авторизуйтесь, используя учётные данные своей учётной записи.

   Если администратор настроил вход через внешнего провайдера аутентификации, войдите через него. Подробнее — в разделе Аутентификация.

Двухфакторная аутентификация

Вы можете включить двухфакторную аутентификацию для своей учётной записи: тогда при входе, помимо пароля, потребуется дополнительное подтверждение. Deckhouse Code поддерживает приложения-аутентификаторы (например, Google Authenticator или Authy) и аппаратные ключи безопасности (U2F, WebAuthn).

Чтобы включить двухфакторную аутентификацию, перейдите в «Профиль» → «Аккаунт» → «Двухфакторная аутентификация» и выполните настройку выбранным способом.

Работа с Git

Способы доступа

Операции Git из командной строки аутентифицируются SSH-ключом или личным токеном доступа по HTTPS.

- SSH-ключи — сгенерируйте пару ключей на компьютере и добавьте открытый ключ в «Профиль» → «SSH-ключи». Используйте SSH-ключ, чтобы клонировать репозиторий и отправлять изменения по URL вида `git@...` без ввода пароля при каждом запросе.
- HTTPS с личным токеном доступа — создайте токен в «Профиль» → «Токены доступа» и используйте его как пароль, когда Git запрашивает учётные данные для URL вида `https://...`.

Администратор может ограничить, какой из двух протоколов доступен на инстансе. Подробнее — в разделе Пользователи и доступ.

Клонирование репозитория

Клонирование позволяет скопировать удалённый Git-репозиторий на локальный компьютер. Для клонирования выполните команду в терминале:

```
git clone <REPOSITORY_URL>
```

Замените `<REPOSITORY_URL>` на HTTPS или SSH-URL вашего репозитория.

Получение изменений

Чтобы получить последние изменения из удалённого репозитория, перейдите в директорию репозитория на локальном компьютере и выполните команду:

```
git pull origin <BRANCH_NAME>
```

Замените `<BRANCH_NAME>` на имя ветки, например, `main` или `feature/my-new-feature`.

Отправка изменений (Push)

1. Внесите необходимые исправления в кодовую базу.
2. Зафиксируйте изменения командой:

   ```
   git commit -am "Описание ваших изменений"
   ```

3. Отправьте изменения в удалённый репозиторий:

   ```
   git push origin <BRANCH_NAME>
   ```

Замените `<BRANCH_NAME>` на имя ветки, в которую отправляются изменения.

Создание feature-ветки

Feature-ветки используются для разработки новых функций или внесения исправлений. Ниже приведены шаги по созданию feature-ветки и отправке её в Deckhouse Code:

1. Склонируйте репозиторий:

   ```
   git clone <REPOSITORY_URL>
   ```

   Замените `<REPOSITORY_URL>` на актуальный URL репозитория.

2. Перейдите в директорию репозитория:

   ```
   cd <REPOSITORY_PATH>
   ```

3. Создайте новую feature-ветку:

   ```
   git checkout -b <BRANCH_NAME>
   ```

   Замените `<BRANCH_NAME>` на имя новой ветки. Флаг `-b` указывает Git создать новую ветку и сразу на неё переключиться.

4. Отправьте новую ветку в удалённый репозиторий:

   ```
   git push origin <BRANCH_NAME>
   ```

   Убедитесь, что ветка появилась в удалённом репозитории. Для проверки откройте проект в Deckhouse Code и найдите новую ветку в меню выбора веток.

Запрос на слияние (Merge Request)

Запрос на слияние используется для объединения изменений из одной ветки в другую.

Создание запроса на слияние

1. Переход в раздел Merge Requests:
   - Откройте проект в Deckhouse Code.
   - В боковой панели выберите раздел: «Код» → «Запросы на слияние».
2. Создание нового запроса:
   - Нажмите кнопку «Новый запрос на слияние».
   - Выберите исходную ветку (с изменениями) и целевую ветку (обычно main).
   - Нажмите «Сравнить ветки и продолжить».

3. Заполнение деталей запроса:
    - Название: опишите кратко суть изменений.
    - Описание: укажите подробности, при необходимости добавьте скриншоты.
    - Ответственный: назначьте пользователя, который будет проводить ревью.
4. Отправка запроса:
    - Нажмите «Создать запрос на слияние».

Ревью и слияние запроса

Слияние происходит после выполнения следующих условий:

- Запрос на слияние прошёл ревью и получил необходимое количество одобрений.
- Отсутствуют конфликты между ветками.
- Все CI/CD пайплайны успешно завершены.

Чтобы завершить процесс, нажмите кнопку «Слияние» на странице запроса.

Проекты и задачи

Группы и участники

Каждый участник группы или проекта имеет одну из предустановленных ролей: Гость, Наблюдатель, Разработчик, Мейнтейнер или Владелец. Роль определяет, что участник может видеть и изменять — как в самой группе, так и в каждом проекте внутри неё.

Группа также несёт настройки, которые применяются ко всем её проектам: правила защиты веток и тегов ограничивают, кто может отправлять изменения напрямую или удалять их, а правила отправки изменений и другие групповые политики задают значения по умолчанию, которые проект наследует, если группа не разрешила ему переопределять их.

1. Создайте группу:
    - Перейдите в раздел «Группы» → «Новая группа».
    - Укажите название группы (например, «Development Team»).
    - Настройте уровень видимости:
        - «Приватный» — виден только членам группы.
        - «Публичный» — виден всем без исключения.
        - «Внутренний» — виден всем зарегистрированным пользователям.

    Создание группы требует разрешения на создание групп, которое по умолчанию есть у каждого зарегистрированного пользователя. Администратор может ограничить это разрешение. Подробнее — в разделе Пользователи и доступ.
2. Пригласите пользователей:
    - Откройте созданную группу.

- Перейдите в раздел «Управлять доступом».
- Нажмите «Пригласить участников», укажите e-mail или имя пользователя и задайте роль (например, «Developer»).

> ⚠ При использовании SSO-интеграций процесс организации доступа может отличаться.

3. Назначьте пользователям группы роли:
   - Перейдите в нужную группу.
   - Выберите «Управление» → «Участники».
   - Назначьте каждому приглашённому пользователю необходимую роль.

В зависимости от назначенной роли пользователю доступен различный набор действий:

Гость:

- Группа: Может просматривать публичные элементы. Нет доступа к конфиденциальным данным.
- Проект: Может просматривать issues в публичных репозиториях, но без права комментирования и управления.

Наблюдатель:

- Группа: Видит всю информацию, включая issues и публичные проекты.
- Проект: Может читать код, CI/CD pipeline и отчёты без права внесения изменений.

Разработчик:

- Группа: Может создавать проекты и вносить изменения в репозитории.
- Проект: Доступ к веткам, коммитам, Merge Requests, CI/CD и задачам разработки.

Мейнтейнер:

- Группа: Полный контроль над проектами и управление членами группы.
- Проект: Управление настройками проекта, ветками, тегами и Merge Requests.

Владелец:

- Группа: Полный контроль, включая удаление и управление ролями.
- Проект: Доступ на уровне группы с полным управлением проектами.

Создание проекта

1. Перейдите в раздел «Проекты» в боковой панели.
2. Нажмите кнопку «Новый проект».

3. Выберите один из доступных способов создания:
   - «Создать пустой проект» — создаёт новый репозиторий с нуля.
   - «Импортировать проект» — перенос существующего проекта из другой системы. При использовании способа «Репозиторий по URL» можно установить флажок «Зеркалировать репозиторий», чтобы новый проект автоматически синхронизировался с исходным репозиторием в режиме pull-зеркалирования.
   - «Создать из шаблона» — проект на основе предварительно подготовленного шаблона.
4. В открывшейся форме укажите параметры проекта:
   - «Имя проекта» — задайте понятное и уникальное имя, например: `my-project`.
   - «Уровень доступа» — определите, кто сможет просматривать проект:
     - «Приватный» — доступ только у участников группы или проекта.
     - «Внутренний» — доступ для всех зарегистрированных пользователей.
5. Нажмите «Создать проект».
6. После создания:
   - Перейдите в созданный проект.
   - Нажмите кнопку «Код» в правом верхнем углу.
   - Скопируйте URL для клонирования — можно выбрать HTTPS или SSH.

Уровни доступа к проекту

Уровень	Описание
«Приватный»	Проект виден только участникам группы или самого проекта
«Внутренний»	Проект доступен всем зарегистрированным пользователям
«Публичный»	Проект доступен всем, включая незарегистрированных пользователей

Импорт проекта

Если исходный инстанс GitLab не имеет сетевого доступа к Deckhouse Code, импорт проекта по URL невозможен. Выгрузите проект в файл на исходном инстансе, а затем загрузите этот файл в Deckhouse Code.

Требования

- Версия GitLab на исходном инстансе совпадает с версией GitLab, на которой работает Deckhouse Code, либо старше её не более чем на две минорные версии. Например, если Deckhouse Code работает на GitLab 19.4, совместимы экспорты из GitLab 19.4, 19.3 и 19.2.
- У вас есть роль `Maintainer` или `Owner` на проекте, который нужно экспортировать.
- У вас есть роль `Maintainer` или `Owner` на целевой группе в Deckhouse Code.

Шаги импорта

Шаг 1. Включите экспорт проектов на исходном инстансе

Администратор исходного инстанса GitLab включает функцию экспорта:

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните «Настройки импорта и экспорта».
3. В разделе «Экспорт проекта» выберите «Включен».
4. Нажмите «Сохранить изменения».

> Если экспорт проектов на исходном инстансе уже включён, пропустите этот шаг.

Шаг 2. Экспортируйте проект

1. Откройте проект, который нужно перенести.
2. Перейдите в «Настройки» → «Основные».
3. Разверните «Расширенные».
4. Нажмите «Экспорт проекта».
5. После формирования экспорта скачайте файл (архив `.tar.gz`) по ссылке из письма или нажав «Скачать экспортируемые данные» на той же странице.

Сохраните экспортированный файл на рабочем компьютере: он понадобится на следующем шаге для загрузки в Deckhouse Code.

Шаг 3. Включите источник импорта «Экспорт в Deckhouse Code»

Администратор инстанса Deckhouse Code включает источник импорта:

1. Перейдите в «Admin» → «Настройки» → «Основные».
2. Разверните «Настройки импорта и экспорта».
3. В разделе «Источники импорта» установите флажок «Экспорт в Deckhouse Code».
4. Нажмите «Сохранить изменения».

Шаг 4. Импортируйте проект

1. В правом верхнем углу выберите «Создать новый» (+) → «Новый проект/репозиторий».
2. Выберите «Импортировать проект».
3. В поле «Импортировать проект из» выберите «Экспорт в Deckhouse Code».
4. Введите имя проекта и URL, затем выберите экспортированный файл.

5. Нажмите «Импортировать проект».

Проверить статус импорта можно через API импорта проекта.

> Для крупных проектов вместо веб-интерфейса используйте Rake-задачи для импорта проекта.

Что переносится, а что нет

Переносится: репозиторий, wiki, задачи, запросы на слияние, метки, вехи, сниппеты, релизы, прямые участники проекта, LFS-объекты, доски задач, архивные CI/CD-пайплайны, защищённые ветки и теги, правила отправки изменений.

Не переносится: CI/CD-переменные, вебхуки, логи и артефакты CI/CD-задач, образы package и container registry, зашифрованные токены, deploy keys, триггеры пайплайнов, политики безопасности.

Полный список экспортируемых и неэкспортируемых данных приведён в справке GitLab по экспорту и импорту проектов.

Управление задачами

Deckhouse Code предоставляет встроенные средства для управления задачами в рамках проекта.

Возможности:

- Создание задач с описанием, лейблами и назначением исполнителей.
- Установка приоритетов и сроков выполнения.
- Поддержка чек-листов (task lists) для разбивки задач на подэтапы.
- Автоматическое закрытие задач по коммитам (например, `Fixes #123`).

Этапы разработки (milestones)

Milestones позволяют объединять задачи и merge-запросы в релизы или спринты.

Возможности:

- Группировка задач и запросов по срокам реализации.
- Визуальное отображение прогресса выполнения.
- Автоматическое закрытие milestone при завершении всех связанных задач.

Kanban-доска (Issue board)

Инструмент для визуального управления задачами по методологии Kanban.

Возможности:

- Настраиваемые колонки (например, To Do, In Progress, Done).
- Перетаскивание задач между колонками для изменения статуса.

- Фильтрация по лейблам, исполнителям и состояниям задач.

Проектная документация (Wiki)

Встроенная Wiki с поддержкой Markdown для хранения и ведения документации проекта.

Возможности:

- Организация документации по страницам и разделам.
- Импорт и экспорт страниц для обмена и резервного копирования.

Wiki группы

Включение Wiki на уровне группы

Чтобы включить или настроить доступ к Wiki, откройте страницу нужной группы и перейдите в: «Настройки» → «Основные» → «Права доступа и возможности группы» → «Уровень доступа к Wiki».

Доступные варианты:

- Включено — Wiki доступна всем (для публичных групп) или только зарегистрированным пользователям (для внутренних групп).
- Приватные — Wiki доступна только участникам группы.
- Отключено — Wiki полностью отключена и недоступна.

Значение по умолчанию — Включено.

Доступ к Wiki

Чтобы открыть Wiki группы, на странице группы выберите «Планирование» → «Wiki».

Роли и Wiki

Уровень доступа определяется членством пользователя в группе:

Роль	Действия
Guest	Чтение Wiki
Reporter	Чтение, загрузка кода
Developer	Создание Wiki
Maintainer	Администрирование Wiki
Planner	Администрирование Wiki
Аноним / внешний пользователь	Доступ, если группа публичная или internal, и пользователь не external — только загрузка кода

Возможности

1. Структура и вложенность — создаёт древовидную навигацию и помогает организовать контент:
    - Создание иерархической структуры страниц с помощью символа `/` в названиях. Пример:

      ```
      devops/ci-pipelines
      devops/kubernetes
      product/design
      ```

2. Markdown, Rich Text, вложения и диаграммы:
    - Markdown (GitLab Flavored):
        - Упоминания (`@username`), ссылки на задачи/merge-реквесты (`#123` , `!456`), чек-листы (`- [ ]`), таблицы, блоки кода.
    - Rich Text Editor:
        - WYSIWYG-редактор для пользователей, предпочитающих визуальное форматирование.
        - Поддержка Mermaid и Draw.io без дополнительной настройки.
    - Вложения:
        - Загрузка и вставка изображений, PDF и других файлов.
        - Хранятся внутри Git-репозитория Wiki.

3. История изменений:
    - Полная история изменений для каждой страницы:
        - Кто вносил изменения.
        - Когда они были сделаны.
    - Просмотр diff-ов, откат изменений, отслеживание всех правок.
4. Обсуждения и комментарии на страницах.
5. Доступ через Git:
    - Каждая Wiki — это отдельный Git-репозиторий.
    - Клонирование через SSH или HTTPS:

      ```
      git clone git@code.example.com:groupname/wiki.git
      ```

    - Полный доступ к `.md`-файлам, веткам, истории для локального редактирования, резервных копий или автоматизации.
6. Экспорт в PDF:
    - Экспорт любой страницы Wiki в PDF через веб-интерфейс.
    - Полезно для офлайн-доступа, обмена или печати.
7. Шаблоны страниц:
    - Многоразовые шаблоны, хранящиеся в директории `templates/`.
    - Применяются при создании или редактировании страниц для стандартизации контента.
8. Настраиваемое боковое меню
    - Боковое меню показывает страницы в виде вложенного дерева.
    - Полностью настраивается с помощью специальной страницы `_sidebar`.
    - Можно добавлять ссылки, разделы и улучшать навигацию.

Аналитика задач

Отчёт «Аналитика задач» показывает, сколько задач открыто и закрыто в каждом месяце указанного периода, и перечисляет задачи этого периода в порядке убывания. Отчёт доступен для группы и для проекта.

Доступность отчёта

Отчёт доступен участникам группы или проекта с ролью «Гость» и выше, в том числе тем, кто получил роль в родительской группе, и пользователям с типом «Аудитор».

В группе или проекте с уровнем видимости «Публичный» отчёт открыт любому посетителю, включая не выполнившего вход.

С уровнем видимости «Внутренний» отчёт открыт любому пользователю, даже если он не состоит в этой группе или проекте.

Просмотр отчёта

Чтобы просмотреть отчёт группы:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика задач».

Чтобы просмотреть отчёт проекта:

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Аналитика задач».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/issues_analytics`
Проект	`https://<HOST>/<GROUP>/<PROJECT>/-/analytics/issues_analytics`

Отчёт группы охватывает задачи её проектов и проектов её подгрупп, кроме архивных.

Обзор в виде графика

График «Обзор» содержит по столбцу на каждый месяц периода, разделённому на серии:

- «Открыто» — задачи по месяцу создания;
- «Закрыто» — задачи по месяцу закрытия.

Задача, созданная в одном месяце и закрытая в другом, попадает в обе серии, поэтому в месяце может быть закрыто больше задач, чем открыто. Месяцы отсчитываются по UTC, а месяц без задач показан пустым столбцом.

Строка над графиком подводит итог периода: `Открыто: всего 128, 10.7 в месяц. Закрыто: всего 94, 7.8 в месяц.` Среднее — это итог, делённый на число месяцев периода.

Период отчёта

Период составляет 12 месяцев, включая текущий. Чтобы задать другую длительность, добавьте к адресу страницы параметр `months_back`:

```
https://<HOST>/groups/<GROUP>/-/issues_analytics?months_back=3
```

Параметр принимает значение от 1 до 36 месяцев, включая текущий. Значение вне диапазона приводится к ближайшей границе, а значение, не являющееся числом, — к 12.

Подпись под горизонтальной осью называет период: `За 3 месяца (Июл. 2026 – Сент. 2026)`, а для периода в один месяц — «Текущий месяц».

Панель фильтров

Панель фильтров над графиком ограничивает график и таблицу одновременно:

- «Автор» — один автор;
- «Ответственный» — один или несколько ответственных;
- «Этап» — один этап;
- «Метка» — одна или несколько меток;
- «Моя реакция» — эмодзи, которым отреагировал текущий пользователь;
- строка поиска — поиск по задачам.

Все фильтры, кроме «Моя реакция», принимают также оператор `!=`, который исключает выбранное значение из отчёта.

Фильтры и период сохраняются в адресе страницы, поэтому настроенный отчёт можно передать ссылкой.

Таблица задач

Таблица «Задачи» перечисляет до 100 задач, созданных в периоде, от новых к старым. Строка над таблицей показывает, сколько задач отображено и сколько соответствует фильтрам: `Самые новые задачи периода: 100 из 240`. Итог считается до 1000; далее строка принимает вид `Самые новые задачи периода: 100 из более чем 1000`.

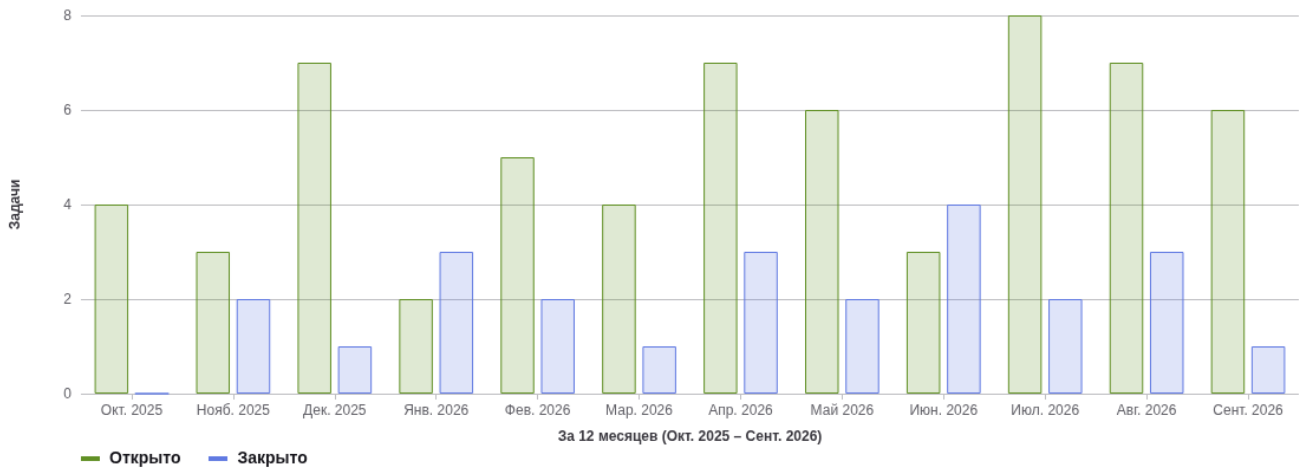
Аналитика задач

Сколько задач открыто и закрыто в каждом месяце, и самые новые задачи за этими числами.

Поиск или фильтрация результатов...

Обзор

Открыто: всего 62, 5.2 в месяц. Закрыто: всего 24, 2.0 в месяц.



Задачи

Самые новые задачи периода: 62

Задача	Возраст	Статус	Этап	Дата завершения	Ответственные	Создано
Archive inactive projects automatically #62 · 3	0 дней	Открыто	Release 2.4	17 окт. 2026 г.	IV	AK
Reduce N+1 queries on the members page #61 · 2	2 дня	Открыто	Release 2.5	13 окт. 2026 г.	MS	IV
Add a health check for the object storage connection #60 · 3	4 дня	Открыто	Release 2.3	9 окт. 2026 г.	DO	MS
Snippet clone URLs use the wrong port behind a proxy #59 · 2	6 дней	Открыто	Release 2.4	5 окт. 2026 г.	AK	DO

Колонка	Содержимое
«Задача»	Заголовок со ссылкой на задачу, её номер и число меток. При наведении на число меток показываются сами метки, а выбор метки открывает список задач с фильтром по ней
«Возраст»	Число дней с момента создания задачи
«Статус»	«Открыто» или «Закрыто»
«Этап»	Этап задачи со ссылкой на него
«Дата завершения»	Дата завершения задачи
«Ответственные»	Пользователи, назначенные на задачу
«Создано»	Автор задачи

Аналитика вкладов

Отчёт «Аналитика вкладов» показывает, сколько пушей, задач и запросов на слияние приходится на каждого участника группы за выбранный период. Отчёт охватывает группу, её подгруппы и все их проекты. Отчёт доступен для группы.

Доступность отчёта

Отчёт доступен участникам группы с ролью «Наблюдатель» и выше, а также пользователям с типом «Аудитор». Участники с ролью «Гость» или «Планировщик» не видят пункт «Аналитика вкладов» в меню группы, а прямой адрес отвечает кодом `404`.

Уровень видимости группы этот доступ не расширяет: с уровнем «Публичный» или «Внутренний» пользователь, не состоящий в группе, отчёт не видит.

Просмотр отчёта

Чтобы просмотреть отчёт:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика вкладов».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/contribution_analytics`

Период отчёта

Период задают вкладки над отчётом: «Прошлая неделя», «Прошлый месяц» и «Последние 3 месяца». Период заканчивается сегодняшним днём и начинается за неделю, месяц или три месяца до него; оба граничных дня входят в период целиком. Дни отсчитываются по часовому поясу сервера, по умолчанию UTC. Строка под вкладками называет дату начала: `Активность по задачам, запросам на слияние и пушам с 2026-09-21.`

Период хранится в адресе страницы как параметр `start_date`, поэтому настроенный отчёт можно передать ссылкой:

```
https://<HOST>/groups/<GROUP>/-/analytics/contribution_analytics?start_date=2026-09-01
```

Параметр принимает дату, например `2026-09-01`. Дата старше трёх месяцев приводится к дате три месяца назад, дата в будущем — к сегодняшнему дню, а значение, не являющееся датой, — к дате неделю назад. Больше трёх месяцев отчёт не охватывает.

Что считается вкладом

Отчёт считает события участников за период в группе, её подгруппах и всех их проектах, включая архивные:

- пуш — один `git push` даёт один пуш независимо от числа коммитов и веток в нём;
- задача — создание и закрытие, в том числе рабочих элементов, созданных непосредственно в группе или подгруппе;
- запрос на слияние — создание, закрытие, слияние и одобрение.

«Всего вкладов» — сумма этих событий участника. Остальные события не учитываются: комментарии, переоткрытие задачи или запроса на слияние, вступление в группу, создание проекта.

Считаются события, поэтому один объект может дать несколько вкладов: задача, созданная и закрытая в периоде, даёт одну открытую и одну закрытую задачу.

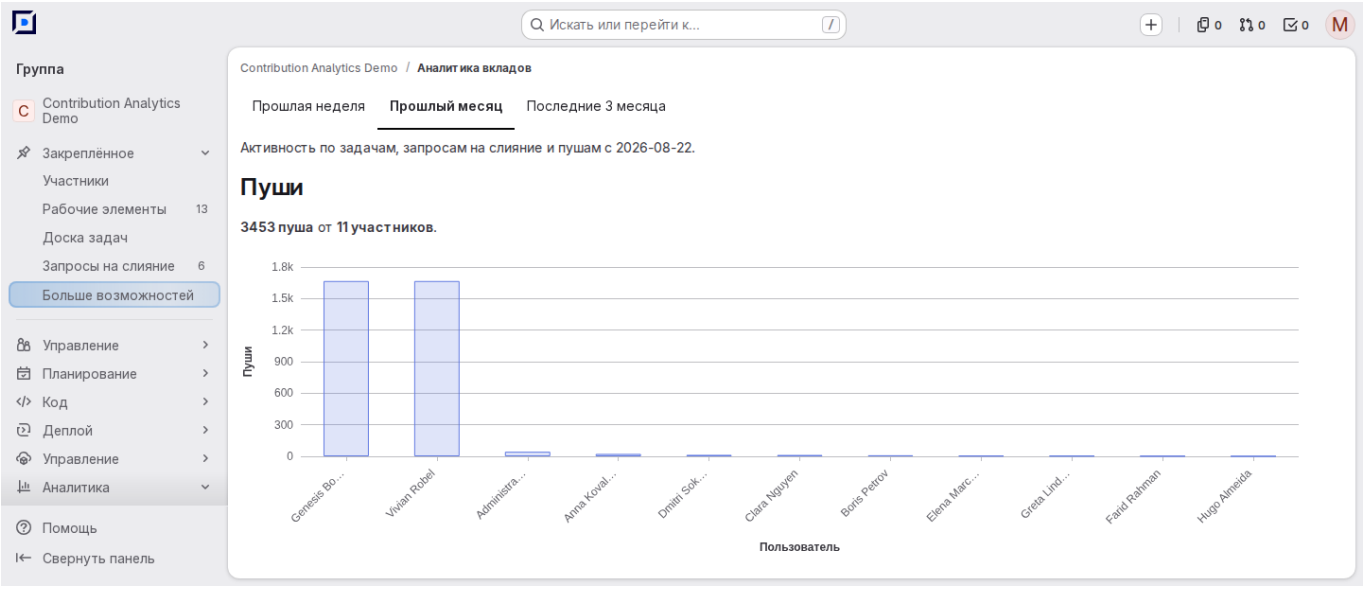
Пользователи с неактивной учётной записью (заблокированной, деактивированной или забаненной) и служебные учётные записи системы в отчёт не попадают вместе со своими событиями, поэтому итоги отчёта могут быть ниже активности группы. Боты токенов доступа проекта и группы перечисляются наравне с остальными участниками.

Новое событие появляется в отчёте в течение минуты: числа кешируются на минуту для каждой группы и периода.

Отчёт читает до 20 000 участников с событиями за период; дальше он показывает тех участников, которых успел прочитать.

График пушей

График «Пуши» содержит по столбцу на каждого участника, у которого за период есть хотя бы один пуш, от большего числа пушей к меньшему. Горизонтальная ось — «Пользователь», вертикальная — «Пуши». Строка над графиком подводит итог периода: `12 пушей от 3 участников.`



Если у двух участников совпадает отображаемое имя, подпись на оси дополняется именем пользователя: `Иван Петров (@ipetrov)`.

Без пушей за период вместо графика показывается текст «Нет пушей за выбранный период.»

График запросов на слияние

График «Запросы на слияние» содержит по группе столбцов на каждого участника, у которого за период есть хотя бы один созданный, слитый или закрытый запрос на слияние, в порядке убывания суммы этих трёх чисел, с разделением на серии:

- «Создано» — запросы на слияние, созданные участником;
- «Слито» — запросы на слияние, слитые участником;
- «Закрыто» — запросы на слияние, закрытые участником.

Одобрения учитываются только в таблице: участник, вся активность которого по запросам на слияние состоит из одобрений, на графике отсутствует.

Строка над графиком подводит итог периода: `Запросы на слияние: создано 8, слито 5, закрыто 2.` Без событий по запросам на слияние за период вместо графика показывается текст «Нет активности по запросам на слияние за выбранный период.»

График задач

График «Задачи» содержит по группе столбцов на каждого участника, у которого за период есть хотя бы одна созданная или закрытая задача, в порядке убывания суммы этих двух чисел, с разделением на серии:

- «Создано» — задачи, созданные участником;
- «Закрыто» — задачи, закрытые участником.

Строка над графиком подводит итог периода: `Задачи: создано 10, закрыто 6.` Без событий по задачам за период вместо графика показывается текст «Нет активности по задачам за выбранный период.»

Таблица вкладов

Таблица «Вклады участников группы» перечисляет всех участников, у которых за период есть хотя бы одно учтённое событие, с сортировкой по колонке «Участник». Нажатие на заголовок колонки сортирует таблицу по ней.

Колонка	Содержимое
«Участник»	Имя участника со ссылкой на его профиль
«Пуши в репозиторий»	Число пушей
«Открытые задачи»	Число задач, созданных участником
«Закрытые задачи»	Число задач, закрытых участником
«Открытые запросы на слияние»	Число запросов на слияние, созданных участником
«Одобренные запросы на слияние»	Число запросов на слияние, одобренных участником
«Слитые запросы на слияние»	Число запросов на слияние, слитых участником
«Закрытые запросы на слияние»	Число запросов на слияние, закрытых участником
«Всего вкладов»	Сумма остальных колонок

Аналитика потоков ценности

Отчёт «Аналитика потоков ценности» показывает, сколько времени задачи и запросы на слияние проводят на каждой стадии своего пути от создания до закрытия или слияния. Отчёт доступен для группы и для проекта. Отчёт группы охватывает группу, её подгруппы и все их проекты.

Доступность отчёта

Отчёт группы доступен участникам группы с ролью «Наблюдатель» и выше, а также пользователям с типом «Аудитор». Участники с ролью «Гость» или «Планировщик» не видят пункт «Аналитика потоков ценности» в меню группы, а прямой адрес отвечает кодом `404`. Уровень видимости группы этот доступ не расширяет: с уровнем «Публичный» или «Внутренний» пользователь, не состоящий в группе, отчёт не видит.

Отчёт проекта доступен участникам проекта с ролью «Гость» и выше, в том числе тем, кто получил роль в родительской группе, а также пользователям с типом «Аудитор». В проекте с уровнем видимости «Публичный» отчёт открыт любому посетителю, включая не выполнившего вход. С уровнем видимости «Внутренний» отчёт открыт любому пользователю, выполнившему вход. В проекте, где выключена функция «Аналитика» либо выключены одновременно задачи и запросы на слияние, пункт меню отсутствует, а прямой адрес отвечает кодом `404`.

Создание, изменение и удаление потоков ценности доступно участникам с ролью «Наблюдатель» и выше в группе или проекте. В архивной группе управлять потоками ценности нельзя.

В Standard Edition страница проекта открывается в виде, описанном в разделе Страница проекта в Standard Edition.

Просмотр отчёта

Чтобы просмотреть отчёт группы:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика потоков ценности».

Чтобы просмотреть отчёт проекта:

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Аналитика потоков ценности».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/value_stream_analytics`
Проект	`https://<HOST>/<GROUP>/<PROJECT>/-/value_stream_analytics`

Страница открывается на плитке «Обзор». Ссылка с параметром `stage_id` открывает отчёт на этой стадии. Выбранные поток ценности, стадия, период, фильтры и проекты сохраняются в адресе страницы, поэтому настроенный отчёт можно передать ссылкой.

Потоки ценности

Отчёт строится вокруг потока ценности — набора стадий, которыми измеряется путь задачи или запроса на слияние. Список потоков ценности расположен над фильтрами; пока поток не выбран, список показывает «Выберите поток создания ценности».

У группы нет встроенного потока ценности. Пока поток не создан, страница показывает «Создайте поток создания ценности, чтобы начать просмотр аналитики.»; участник, который может управлять потоками ценности, видит список с пунктом «Новый поток создания ценности».

Проект начинает со встроенного потока ценности с названием «default», который содержит встроенные стадии. Когда в проекте появляется сохранённый поток ценности, список содержит только сохранённые потоки. Встроенный поток нельзя изменить или удалить.

Список упорядочен по названию. Выбранный поток сохраняется в адресе страницы как параметр `value_stream_id`.

Стадии

Ряд плиток над отчётом — навигация по выбранному потоку: плитка «Обзор», затем по плитке на каждую стадию потока в порядке, заданном в потоке. У скрытой стадии плитки нет.

«Обзор» показывает плитки «Метрики жизненного цикла» и графики «Общее время» и «Задачи по типу». Стадия показывает график «Время стадии» и таблицу этой стадии.

Стадия измеряет время между своим событием начала и событием конца для одной задачи или одного запроса на слияние. Элемент попадает в стадию, когда достиг события конца; элемент, достигший только события начала, не учитывается. При наведении на плитку стадии показываются «Время этапа (медиана)», «Элементы стадии», примечание «Только элементы, достигшие стоп-события.» и события стадии под подписями «Запустить» и «Остановить».

Встроенные стадии и их события:

Стадия	Событие начала	Событие конца
«Задача»	«Создана задача»	«Проблема, впервые связанная с этапом или впервые добавленная на доску»
«Планирование»	«Проблема, впервые связанная с этапом или впервые добавленная на доску»	«Задача впервые упомянута в коммите»
«Написание кода»	«Задача впервые упомянута в коммите»	«Создан запрос на слияние»
«Тестирование»	«Сборка запроса на слияние начата»	«Последняя сборка запроса на слияние завершена»
«Контроль»	«Создан запрос на слияние»	«Запрос на слияние выполнен»
«Приёмка»	«Запрос на слияние выполнен»	«Запрос на слияние впервые развёрнут в продукцию»

Период отчёта отбирает элементы стадии по дате: для сохранённого потока ценности — по дате события конца, для встроенного потока — по дате создания задачи или запроса на слияние.

Метрики жизненного цикла

Ряд «Метрики жизненного цикла» показывается на «Обзоре» и ограничивается периодом и фильтрами страницы:

- «Время выполнения» — медианное время в днях от создания задачи до её закрытия;
- «Время цикла» — медианное время в днях от первого коммита, упоминающего задачу, до закрытия этой задачи;
- «Новые задачи» — число задач, созданных в периоде;
- «Коммиты» — число коммитов в ветке по умолчанию за период; плитка показывается только в отчёте проекта;
- «Деплои» — число успешных развёртываний в окружения с уровнем развёртывания `production`, завершённых в периоде; в отчёте группы — по проектам группы.

Медианы считаются в момент открытия страницы по самим задачам, какой бы поток ценности ни был выбран. Значение округляется до одного знака после запятой. Период плиток длится до конца своего последнего дня, поэтому работа текущего дня учитывается.

Графики

График «Общее время» показывается на «Обзоре»: по области на каждую стадию выбранного потока, сложенные друг на друга, с точкой на каждый день. Точка — средняя длительность элементов, завершивших стадию в этот день; подзаголовок графика — «Среднее время каждого этапа, сложенное в полный цикл». Скрытые стадии не учитываются. День, в который стадия ничего не завершила, рисуется пунктирной линией с названием «Нет данных» между соседними днями. Период, в котором ни одна стадия ничего не завершила, показывает «За выбранный период ни одна стадия ничего не завершила».

График «Время стадии: » заменяет его, пока выбрана стадия: те же точки с названием «Среднее время» только для этой стадии. Период, в котором стадия ничего не завершила, показывает «За выбранный период эта стадия ничего не завершила».

Оба графика показывают среднее, а плитки и таблица стадии — медиану: один элемент с необычно долгой длительностью сдвигает точку графика.

График «Задачи по типу» показывается на «Обзоре»: по столбцу на каждый день периода с разбивкой по меткам. Столбец считает задачи, закрытые в этот день, или запросы на слияние, слитые в этот день, по дате закрытия или слияния. График настраивается списком «Метки (не более 15)» в его заголовке:

- список предлагает метки группы, а в отчёте проекта — метки проекта и его групп; поле в начале списка ищет по названию;
- отчёт открывается с первыми 10 метками в алфавитном порядке; выбор более 15 меток отклоняется с сообщением «Одновременно можно показать не более 15 меток.»;
- переключатель «Показать» под списком выбирает «Задачи» или «Запросы на слияние».

Элемент с несколькими выбранными метками считается под каждой из них. Фильтр «Метка» страницы на этот график не действует. Период без закрытых или слитых элементов с выбранными метками показывает «За выбранный период не завершено ни одной работы с метками».

Таблица стадии

Таблица выбранной стадии показывается под графиком «Время стадии». Её колонки:

Колонка	Содержимое
«Задачи» или «Запросы на слияние»	Заголовок со ссылкой на элемент, его автор и дата создания; число элементов в стадии показано рядом с названием колонки
«Последнее событие»	Дата и время события конца
«Длительность»	Время между событием начала и событием конца

Таблица сортируется по «Последнее событие» или «Длительность» и разбита на страницы. В отчёте группы каждая строка называет также проект элемента.

Стадия без элементов в периоде при текущих фильтрах показывает «На этом этапе нет элементов для отображения с выбранными фильтрами и в указанном диапазоне времени.» Пользователь, который не может читать элементы стадии, например задачи стадии, измеряемой событиями сборки, видит «Вам нужны права.»

Период и фильтры

Список «Диапазон дат» над отчётом задаёт период: «Прошлая неделя», «Последние 30 дней», «Последние 90 дней», «Последние 180 дней» или «Настраиваемый», который открывает поля выбора даты «От» и «В». Отчёт открывается на последних 30 днях, включая сегодняшний. Период ограничен 180 днями. Период сохраняется в адресе как параметры `created_after` и `created_before`.

Панель «Фильтровать результаты» ограничивает плитки, графики и таблицу одновременно:

- «Этап» — один этап;
- «Метка» — одна или несколько меток;
- «Автор» — один автор;
- «Ответственный» — один или несколько ответственных.

Панель предлагает только совпадение по значению. Чтобы исключить значение, добавьте фильтр в адрес страницы в форме `not`, например `not[author_username]=<USER>`; отчёт тогда показывает всё, что не соответствует значению.

В отчёте группы список «Выбрать проекты» ограничивает отчёт одним или несколькими проектами группы и её подгрупп. Выбор сохраняется в адресе как параметры `project_ids[]`.

Свежесть данных

Сохранённый поток ценности измеряется по событиям стадий, которые фоновая задача записывает заранее; встроенный поток измеряется по самим задачам и запросам на слияние в момент запроса.

После сохранения потока ценности задача сразу обходит его стадии. Затем она запускается каждые 10 минут и читает события, появившиеся после предыдущего запуска. Пока идёт первый обход стадии, страница показывает «Данные отчёта ещё собираются, поэтому он охватывает пока лишь часть истории.» Когда первый обход завершён, страница показывает вместо этого «Данные на », а значок рядом — «Следующее обновление ожидается ». Ни одна из строк не показывается, пока обход ни разу не выполнялся и не идёт сейчас.

Создание потока ценности

Чтобы создать поток ценности:

1. Откройте отчёт группы или проекта.

2. В списке потоков ценности выберите «Новый поток создания ценности». Пункт расположен под разделителем в конце списка.
3. В поле «Название» введите название потока: от 3 до 100 символов, уникальное в группе или проекте.
4. Расставьте стадии в разделе «Стадии». Форма открывается со встроенными стадиями в их порядке. Встроенная стадия перемещается кнопками «Переместить вверх» и «Переместить вниз» и убирается из отчёта кнопкой «Скрыть стадию», которая помечает её как «Скрыта»; кнопка «Показать стадию» возвращает её. Встроенная стадия сохраняет свои события, и удалить её нельзя.
5. Чтобы добавить собственную стадию, нажмите «Добавить стадию» и заполните поля «Название стадии», «Событие начала» и «Событие конца». Для события по метке выберите метку в поле «Метка события начала» или «Метка события конца». Кнопка «Удалить стадию» убирает стадию из формы.
6. Нажмите «Сохранить поток создания ценности». Страница возвращается к отчёту, который открывается на первом потоке списка.

Кнопка «Вернуть набор по умолчанию» заменяет список стадий формы встроенными стадиями в их порядке.

Поток ценности содержит от 1 до 15 стадий: форма прекращает добавлять стадии с сообщением «Поток создания ценности вмещает не более 15 стадий.», а поток без стадий отклоняется с сообщением «В потоке создания ценности должна быть хотя бы одна стадия». Название собственной стадии не может повторять название встроенной стадии или другой стадии потока. Группа или проект содержит до 50 потоков ценности.

Форма также доступна по прямому адресу:

Форма	**Адрес**
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/value_stream_analytics/value_streams/new`
Проект	`https://<HOST>/<GROUP>/<PROJECT>/-/value_stream_analytics/value_streams/new`

События стадий

Собственная стадия открывается событием задачи или запроса на слияние и закрывается другим событием того же объекта; событие задачи и событие запроса на слияние соединить нельзя. Список «Событие конца» предлагает события, которые сочетаются с выбранным событием начала.

События задачи:

- «Создана задача»;

- «Задача впервые связана с этапом»;
- «Задача впервые добавлена на доску»;
- «Задача впервые назначена»;
- «Задача впервые упомянута в коммите»;
- «Задаче добавлена метка»;
- «У задачи снята метка»;
- «Задача последний раз изменена»;
- «Задача закрыта»;
- «Задача впервые развёрнута в рабочей среде».

События запроса на слияние:

- «Создан запрос на слияние»;
- «В запрос на слияние сделан первый коммит»;
- «Запрос на слияние впервые назначен»;
- «Запросу на слияние впервые назначен проверяющий»;
- «Запросу на слияние добавлена метка»;
- «У запроса на слияние снята метка»;
- «Сборка запроса на слияние начата»;
- «Последняя сборка запроса на слияние завершена»;
- «Запрос на слияние последний раз одобрен»;
- «Запрос на слияние последний раз изменён»;
- «Запрос на слияние выполнен»;
- «Запрос на слияние закрыт»;
- «Запрос на слияние впервые развёрнут в продукцию».

Пары ограничены теми, которые читаются вперёд во времени:

- «Создана задача» и «Создан запрос на слияние» только открывают стадию; «Задача впервые развёрнута в рабочей среде» и «Запрос на слияние впервые развёрнут в продукцию» только закрывают её.
- Событие о первом случае («Задача впервые назначена», «Задача впервые связана с этапом», «В запрос на слияние сделан первый коммит») не закрывает стадию, открытую событием о последнем случае («Задача закрыта», «Запрос на слияние последний раз одобрен», «У задачи снята метка»), потому что первый случай раньше последнего.
- «В запрос на слияние сделан первый коммит» не закрывает стадию, открытую событием «Создан запрос на слияние»: первый коммит запроса на слияние раньше самого запроса. «Сборка запроса на слияние начата» не закрывает стадию, открытую событием «Последняя сборка запроса на слияние завершена».
- «Задаче добавлена метка», «У задачи снята метка», «Запросу на слияние добавлена метка» и «У запроса на слияние снята метка» принимают по метке, поэтому событие по метке может открывать и закрывать одну и ту же стадию: стадия тогда измеряет время от одной метки до другой.

Изменение потока ценности

Чтобы изменить сохранённый поток ценности:

1. В списке потоков ценности выберите поток.
2. Нажмите «Редактировать» рядом со списком.
3. Измените название и стадии так, как они задаются в разделе Создание потока ценности. Стадия, убранная из формы, удаляется из потока при сохранении.
4. Нажмите «Сохранить поток создания ценности».

Кнопка «Редактировать» показывается участнику, который может управлять потоками ценности, и только пока выбран сохранённый поток.

Удаление потока ценности

Чтобы удалить сохранённый поток ценности:

1. В списке потоков ценности выберите поток.
2. В списке под разделителем выберите «Удалить », где `<NAME>` — название выбранного потока.
3. В диалоге «Удалить ?» нажмите «Удалить». Диалог сообщает: «Поток создания ценности и его стадии будут удалены. Измеренная ими активность — нет.»

После удаления список читается заново, и выбирается его первый поток. В проекте после удаления последнего сохранённого потока в список возвращается встроенный поток «default».

Страница проекта в Standard Edition

В Standard Edition страница проекта открывается без описанных выше дополнений:

- отчёт содержит встроенный поток «default» и его встроенные стадии; списка потоков ценности нет, а адреса формы отвечают кодом `404`;
- ряд «Метрики жизненного цикла» содержит «Новые задачи», «Коммиты» и «Деплои»; плиток «Время выполнения» и «Время цикла» нет;
- графиков «Общее время», «Время стадии» и «Задачи по типу» и строки о свежести данных нет;
- список «Диапазон дат» и панель «Фильтровать результаты» работают так, как описано в разделе Период и фильтры, а таблица стадии — как описано в разделе Таблица стадии.

Дашборд потоков ценности

«Дашборд потоков ценности» показывает счётчики обзора группы или проекта, их метрики жизненного цикла и DORA-метрики за последние полгода по месяцам, а для группы — как её проекты оцениваются по DORA-метрикам. Дашборд группы охватывает группу, её подгруппы и все их проекты. У личного проекта, расположенного в пространстве пользователя, дашборда нет.

Дашборд не принимает ни период, ни фильтры: таблицы охватывают пять прошедших календарных месяцев и текущий месяц, а счётчики показывают последний снимок.

Доступность отчёта

Дашборд группы доступен участникам группы с ролью «Наблюдатель» и выше, а также пользователям с типом «Аудитор». Участники с ролью «Гость» или «Планировщик» не видят пункт «Дашборд потоков ценности» в меню группы, а прямой адрес отвечает кодом `404`. Уровень видимости группы этот доступ не расширяет: с уровнем «Публичный» или «Внутренний» пользователь, не состоящий в группе, дашборд не видит.

Дашборд проекта доступен участникам проекта с ролью «Гость» и выше, в том числе тем, кто получил роль в родительской группе, а также пользователям с типом «Аудитор». В проекте с уровнем видимости «Публичный» дашборд открыт любому посетителю, включая не выполнившего вход. С уровнем видимости «Внутренний» дашборд открыт любому пользователю, выполнившему вход. В проекте, где выключена функция «Аналитика» либо выключены одновременно задачи и запросы на слияние, пункт меню отсутствует, а прямой адрес отвечает кодом `404`. У личного проекта нет ни пункта меню, ни страницы.

Панелям DORA нужно отдельное разрешение: таблице «DORA-метрики: », а на странице группы — также диаграмме «Оценка проектов по DORA: » и таблице «Проекты по DORA-метрике: ». Участники с ролью «Наблюдатель» и выше имеют его в группе и в проекте; пользователи с типом «Аудитор» — в группе. Без него панели остаются на странице и показывают вместо содержимого «У вас недостаточно разрешений для просмотра этой панели.» Так видят панели DORA участник с ролью «Гость», посетитель публичного проекта и пользователь с типом «Аудитор» на странице проекта.

Просмотр отчёта

Чтобы просмотреть дашборд группы:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Дашборд потоков ценности».

Чтобы просмотреть дашборд проекта:

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Дашборд потоков ценности».

Также дашборд можно открыть по прямому адресу:

Дашборд	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/dashboards/value_streams_dashboard`
Проект	`https://<HOST>/<GROUP>/<PROJECT>/-/analytics/dashboards/value_streams_dashboard`

Панели

Страница группы содержит сверху вниз:

- карточку «Обзор »;
- счётчики «Группы», «Проекты», «Пользователи», «Задачи», «Запросы на слияние» и «Пайплайны»;
- таблицу «Lifecycle-метрики: »;
- таблицу «DORA-метрики: »;
- диаграмму «Оценка проектов по DORA: »;
- таблицу «Проекты по DORA-метрике: ».

Страница проекта содержит карточку «Обзор », счётчики «Задачи», «Запросы на слияние» и «Пайплайны» и таблицы «Lifecycle-метрики: » и «DORA-метрики: ». `<NAMESPACE>` в названии панели — название группы или проекта.

Панель, данные которой не удалось загрузить, показывает «Произошла ошибка.» и предлагает «Повторить».

Карточка обзора и счётчики

Карточка «Обзор » показывает аватар, полное название со ссылкой на группу или проект, тип («Группа» или «Проект») и уровень видимости («Публичный», «Внутренний» или «Приватный»).

Счётчики — это снимки, которые снимаются в фоне раз в месяц, в последние три дня месяца, для иерархий, чья корневая группа включила подсчёт (см. раздел Включение подсчёта обзора). Счётчик читает самый свежий снимок, снятый за последний год, и показывает под значением «Последний подсчёт: ». Снимок охватывает каждую группу и каждый проект иерархии, поэтому дашборд подгруппы читает ту часть снимка, которая относится к подгруппе.

На странице группы счётчики считают:

- «Группы» — подгруппы группы на всех уровнях;
- «Проекты» — проекты группы и её подгрупп;
- «Пользователи» — прямых участников самой группы; участники подгрупп, неподтверждённые приглашения и запросы доступа не учитываются;

- «Задачи», «Запросы на слияние» и «Пайплайны» — по проектам группы и её подгрупп; запрос на слияние учитывается в целевом проекте.

На странице проекта «Задачи», «Запросы на слияние» и «Пайплайны» считают сам проект.

Пока корневая группа не включила подсчёт, счётчики показывают 0, а предупреждение в начале страницы сообщает: «Счётчики обзора — группы, проекты, пользователи, задачи, запросы на слияние и пайплайны — для этой иерархии не собираются, поэтому эти панели показывают ноль.» Пользователь, который может открыть настройки корневой группы, видит далее «Включите фоновый подсчёт в настройках аналитики корневой группы.», где «настройках аналитики» — ссылка на страницу; остальные видят «Владелец корневой группы может включить фоновый подсчёт в её настройках аналитики.» Каждая панель счётчика также показывает предупреждение «Этот счётчик снимается в фоне раз в месяц, а корневая группа не включила подсчёт.»

Когда подсчёт включён, предупреждение в начале страницы исчезает; до первого снимка счётчики показывают 0, а каждая панель счётчика показывает предупреждение «Снимок этого счётчика ещё не снят. Фоновый подсчёт выполняется раз в месяц.» Когда подсчёт выключен, счётчики снова показывают 0 с первым предупреждением, какие бы снимки ни были сняты.

Включение подсчёта обзора

Подсчёт включается в настройках корневой группы участником с ролью «Владелец». У подгруппы такой страницы нет: пункт «Аналитика» отсутствует в её меню «Настройки», а прямой адрес отвечает кодом `404`.

Чтобы включить подсчёт:

1. В верхней панели нажмите «Искать или перейти к…» и выберите корневую группу.
2. В меню группы перейдите в «Настройки» → «Аналитика».
3. В разделе «Дашборд потоков ценности» установите флажок «Включить фоновую агрегацию обзора». Раздел сообщает: «Счётчики обзора на дашборде снимаются в фоне раз в месяц для всей иерархии группы.», а флажок — «Без неё панели обзора не показывают счётчики.»
4. Нажмите «Сохранить изменения». Страница сообщает: «Настройки аналитики успешно обновлены.»

Страница также доступна по адресу `https://<HOST>/groups/<GROUP>/-/settings/analytics`.

Снимок иерархии снимается в последние три дня месяца, раз в месяц; иерархия, включившая подсчёт раньше, получает первый снимок в эти дни. Задание запускается ежечасно, и иерархия, слишком большая для одного запуска, досчитывается на следующем.

Чтобы выключить подсчёт, снимите флажок «Включить фоновую агрегацию обзора» и нажмите «Сохранить изменения».

Таблица метрик жизненного цикла

Таблица «Lifecycle-метрики: » содержит колонку «Метрика», шесть колонок месяцев, озаглавленных месяцем и годом, колонку «Изменение (%)» и колонку «Тренд»:

- колонки месяцев — пять прошедших календарных месяцев, начиная с самого раннего, и текущий месяц по сегодняшний день; границы месяцев считаются в часовом поясе браузера;
- «Изменение (%)» — изменение последнего полного месяца относительно первой колонки, месяца пятимесячной давности; текущий месяц не сравнивается. Значок зелёный, когда изменение улучшает показатель строки, и красный в остальных случаях: для количества — рост, для длительности — снижение. Значок показывает «н/д», когда у одного из двух месяцев нет значения или более ранний месяц равен нулю;
- «Тренд» — линия по пяти прошедшим месяцам без текущего.

Каждая ячейка вычисляется в момент открытия страницы. Ячейка без значения показывает «-». Длительности указаны в днях. Значок рядом с названием строки открывает её описание. Строка, данные которой не удалось загрузить, названа в предупреждении панели «Не удалось загрузить некоторые сравнения метрик: » с кнопкой «Повторить».

Строки таблицы:

Строка	Содержимое
«Время выполнения»	Медианное время от создания задачи до её закрытия по задачам, созданным в этом месяце и закрытым
«Время цикла»	Медианное время от первого коммита, упоминающего задачу, до закрытия этой задачи по задачам, созданным в этом месяце и закрытым
«Создано задач»	Задачи, созданные в этом месяце
«Закрыто задач»	Задачи, созданные в этом месяце и закрытые; свыше 10 000 ячейка показывает «10000+»
«Деплои»	Успешные развёртывания в production-окружения, завершённые в этом месяце
«Медианное время до слияния»	Медианное время от создания запроса на слияние до его слияния по запросам, созданным в этом месяце и слитым
«Пропускная способность запросов на слияние»	Запросы на слияние, слитые в этом месяце

Название каждой строки — ссылка на отчёт, который её раскрывает:

- «Время выполнения» и «Время цикла» — Аналитика потоков ценности;
- «Создано задач» и «Закрыто задач» — Аналитика задач;
- «Деплои», «Медианное время до слияния» и «Пропускная способность запросов на слияние» — на странице группы Аналитика продуктивности, на странице проекта Аналитика запросов на слияние.

Таблица DORA-метрик

Таблица «DORA-метрики: » содержит те же колонки, что и таблица метрик жизненного цикла. Её строки вычисляются по production-окружениям проекта, а на странице группы — всех проектов группы и её подгрупп, из дневных значений, описанных в разделе Запись DORA-метрик:

- «Частота деплоя» — успешные развёртывания в день: развёртывания месяца, делённые на число его календарных дней, для текущего месяца — на число дней по сегодняшний;
- «Время проведения изменений» — медианное время в днях от слияния запроса на слияние до окончания развёртывания, которое его доставило; медиана берётся по дневным медианам месяца;
- «Время восстановления сервиса» — медианное время в днях от создания инцидента до его закрытия; медиана берётся по дневным медианам месяца;
- «Доля неудачных изменений» — инциденты, созданные в этом месяце, на одно развёртывание месяца, в процентах; месяц без развёртываний показывает «-».

Значок «Изменение (%)» зелёный при росте «Частота деплоя» и при снижении трёх остальных строк. Названия строк этой таблицы не являются ссылками.

Оценка проектов по DORA

Диаграмма «Оценка проектов по DORA: » есть только на странице группы. Она оценивает каждый доступный вам проект группы и её подгрупп за последний полный календарный месяц по каждой DORA-метрике и показывает по столбцу на метрику: сегменты «Высокая», «Средняя» и «Низкая» считают проекты с такой оценкой, сегмент «Нет данных» — оценённые проекты без значения этой метрики. Оси подписаны «DORA-метрика» и «Проекты». Подпись под диаграммой: «Оценено проектов за последний полный месяц: ; без DORA-данных: .», где `<M>` — проекты без записанного DORA-дня в этом месяце; такие проекты ни в один столбец не входят. При наведении на столбец показываются пороги его метрики.

Проект получает оценку за месяц, когда в один из дней этого месяца записано событие в одном из его production-окружений. Оценка каждой метрики:

Метрика	«Высокая»	«Низкая»
«Частота деплоя»	не меньше одного развёртывания в день	меньше одного развёртывания за 30 дней
«Время проведения изменений»	не больше 7 дней	больше 30 дней
«Время восстановления сервиса»	не больше одного дня	больше 7 дней
«Доля неудачных изменений»	не больше 15 %	больше 45 %

Между двумя порогами оценка «Средняя». Когда ни один проект группы не оценён за месяц, каждый столбец относит все проекты к сегменту «Нет данных», а диаграмма добавляет «Ни один проект не получил оценку по: .»

Проекты по DORA-метрике

Таблица «Проекты по DORA-метрике: » есть только на странице группы. Она перечисляет доступные вам проекты группы и её подгрупп, у которых есть записанный DORA-день в последние два полных месяца: колонка «Проект» с аватаром и названием со ссылкой на проект, затем по колонке на каждую DORA-метрику со значением за последний полный месяц и значком его изменения относительно предыдущего месяца. Подпись над таблицей: «Показано из проектов; у нет DORA-данных за последние два месяца.»

Таблица открывается отсортированной по колонке «Проект»; чтобы отсортировать по другой колонке, нажмите её заголовок. Проекты без значения в колонке идут последними при любом направлении сортировки.

Запись DORA-метрик

Четыре метрики, их дневные графики и их REST API описаны на странице «DORA-метрики».

DORA-значения записываются по окружению и по дню в момент события:

- успешное развёртывание учитывается в день его окончания; время проведения изменений измеряется от слияния каждого запроса на слияние, связанного с развёртыванием, до окончания развёртывания, когда запросы на слияние связаны с ним, через пять минут после развёртывания;
- созданный, закрытый или переоткрытый инцидент записывается в самое старое production-окружение проекта: инцидент — в день создания, время его восстановления — в день закрытия; инцидент, помеченный как дубликат, не учитывается;
- задача, у которой изменён тип на «Инцидент» или обратно либо поставлена или снята пометка дубликата, пересчитывается в день её создания и, если она закрыта, в день закрытия.

День события берётся в часовом поясе сервера. Дашборд читает только production-окружения, поэтому развёртывание в окружение другого уровня не учитывается.

За время до обновления на версию с дашбордом ничего не записано, и история задним числом не восстанавливается. Удаление окружения удаляет его записанные дни; удаление самого старого production-окружения проекта удаляет вместе с ним историю инцидентов проекта.

Репозитории

Управление репозиториями Git

Хранение и версионирование кода

Deckhouse Code предоставляет инструменты для работы с репозиториями Git. Каждый проект включает репозиторий для хранения исходного кода и отслеживания истории изменений:

- Полноценная поддержка распределённой системы контроля версий Git.
- Выполнение стандартных команд: клонирование (`git clone`), коммит (`git commit`), отправка (`git push`), получение (`git pull`).
- Автоматическое отслеживание и сохранение истории изменений.

Поддержка форков (Forks)

Форк — это независимая копия репозитория, предназначенная для разработки без риска повлиять на основной код:

1. Разработчик создаёт форк основного репозитория.
2. Вносит изменения в своём форке.
3. Отправляет merge request (MR) в оригинальный репозиторий.
4. После проверки изменения могут быть слиты.

Подход широко используется в open source-проектах и при ограниченном доступе к основному репозиторию.

Работа с ветками (branching) и слиянием (merging)

Ветки позволяют разрабатывать новые функции и исправления изолированно от основной кодовой базы:

- Создание веток: `git branch <BRANCH_NAME>` , `git checkout -b <BRANCH_NAME>` .
- Переключение между ветками: `git checkout <BRANCH_NAME>` .
- Слияние изменений: `git merge <BRANCH_NAME>` .
- Поддержка защищённых веток (protected branches) — для внесения изменений требуется merge request.
- Автоматическое удаление веток после слияния (если включено).

Встроенный web-редактор

Позволяет вносить изменения в код прямо через браузер без необходимости клонировать репозиторий:

- Создание и редактирование файлов через веб-интерфейс.
- Автоматическое создание коммитов при сохранении изменений.
- Поддержка Markdown с возможностью предпросмотра.

Поддержка Git LFS (Large File Storage)

Git LFS позволяет эффективно управлять большими файлами, такими как:

- Изображения;
- Видео;
- Аудиофайлы;
- Датасеты для машинного обучения.

Git LFS заменяет содержимое больших файлов ссылками, а сами файлы хранятся вне Git-репозитория. Это снижает нагрузку на историю коммитов и ускоряет работу с репозиторием.

Защищённые ветки

Защищённые ветки (protected branches) ограничивают изменение важных веток репозитория: отправлять изменения (push) и выполнять слияние в такую ветку могут только те, кому это явно разрешено. Обычно защита используется для веток `main`, `master`, релизных и стабильных веток — чтобы изменения попадали в них только через merge request и код-ревью.

Ветка проекта, назначенная веткой по умолчанию, защищается автоматически при его создании.

Ограничения защищённых веток

Для защищённой ветки действуют следующие ограничения:

- Прямая отправка изменений (`git push`) запрещена для всех, кроме пользователей, указанных в списке «Разрешено выполнять push и слияние». Остальные участники вносят изменения через merge request.
- Принудительная отправка (`git push --force`) запрещена для всех, пока в правиле ветки не включена настройка «Разрешить force push».
- Ветку нельзя удалить командами Git. Удаление доступно только через веб-интерфейс пользователям с ролью `Maintainer` и выше.

Защита ветки по умолчанию

При создании проекта его ветка по умолчанию автоматически становится защищённой. Начальный уровень защиты задаётся настройкой «Защита начальной ветки по умолчанию»:

- **На уровне инстанса** — раздел «Область администратора» → «Настройки» → «Репозиторий», секция «Ветка по умолчанию». Настройка доступна администраторам инстанса.
- **На уровне группы** — на странице группы перейдите в раздел «Настройки» → «Репозиторий», секция «Ветка по умолчанию». Настройка доступна пользователям с ролью `Owner` в группе.

Настройка группы имеет приоритет над настройкой инстанса: если в группе она не задана, действует настройка инстанса. Для проектов в личном пространстве пользователя всегда действует настройка инстанса.

По умолчанию применяется полная защита: push и слияние разрешены только участникам с ролью `Maintainer` и выше, force push запрещён.

Правила веток

Защита настраивается через правила веток. Для этого откройте страницу проекта и перейдите в раздел «Настройки» → «Репозиторий» → «Правила веток». Настройка доступна пользователям с ролью `Maintainer` и `Owner`.

Чтобы создать правило:

1. Нажмите «Добавить правило ветки».
2. Выберите «Имя или шаблон ветки».
3. Укажите имя конкретной ветки (например, `main`) или шаблон с подстановочным символом `*` (например, `release/*`). Правило с шаблоном защищает все подходящие ветки, включая те, которые будут созданы позже.
4. Нажмите «Создать правило ветвления».

Чтобы открыть настройки существующего правила, нажмите «Просмотреть детали» напротив него в списке.

Настройка доступа к защищённой ветке

На странице правила в секции «Защитить ветку» настраиваются два списка доступа:

- **«Разрешено слияние»** — кто может выполнять слияние merge request в эту ветку.
- **«Разрешено выполнять push и слияние»** — кто может отправлять изменения в ветку напрямую (`git push`), а также выполнять слияние.

Чтобы изменить соответствующий список, нажмите «Изменить права на слияние» или «Изменить права на push и слияние».

В каждом списке можно комбинировать роли, отдельных пользователей и группы. В списке «Разрешено выполнять push и слияние» дополнительно доступны ключи развёртывания. Доступ получает каждый, кто подходит хотя бы под один из выбранных пунктов.

Доступные варианты:

Пункт списка	Кому предоставляет доступ
«Мейнтейнеры»	Участникам проекта с ролью `Maintainer` и выше
«Разработчики и Мейнтейнеры»	Участникам проекта с ролью `Developer` и выше
«Никто»	Никому. Доступ по ролям полностью закрыт
«Администраторы»	Администраторам инстанса
«Пользователи»	Перечисленным участникам проекта с правом записи в репозиторий (роль `Developer` и выше). Подробнее — в разделе «Доступ для отдельных пользователей и групп»
«Группы»	Прямым участникам перечисленных групп с ролью в группе `Developer` и выше. Группа должна быть приглашена в проект с ролью не ниже `Developer`. Подробнее — в разделе «Доступ для отдельных пользователей и групп»
«Ключи развёртывания»	Владельцу выбранного ключа развёртывания, в том числе при отправке изменений по этому ключу. Ключ должен быть включён в проекте с правом записи, а владелец ключа — быть участником проекта. Доступно только для списка «Разрешено выполнять push и слияние»

Доступ для отдельных пользователей и групп

Помимо ролей, доступ к защищённой ветке можно выдать точечно — конкретным пользователям и группам. Это позволяет, например, закрыть ветку для всех (выбрав «Никто») и разрешить слияние только релиз-менеджеру, не повышая его роль в проекте.

Пользователи

В селекторе «Пользователи» можно выбрать участников проекта с правом записи в репозиторий — пользователей с ролью `Developer` и выше, включая участников, унаследованных из родительской группы.

Добавление пользователя в список не повышает его права в проекте. Участник без роли `Developer` и выше не сможет отправлять изменения в защищённую ветку, даже если он указан в списке.

Эти списки позволяют сузить круг тех, кому право записи уже выдано ролью. Например, выберите «Никто» в ролях и перечислите конкретных пользователей — тогда работать с этой веткой смогут только они.

Группы

В селекторе «Группы» отображаются только группы, приглашённые в проект с ролью `Developer` и выше. Приглашение группы настраивается в разделе «Управление» → «Участники» проекта.

Родительские группы проекта и группы, приглашённые с меньшей ролью, выбрать нельзя.

Доступ через группу получают только прямые участники этой группы с ролью в ней `Developer` и выше. Участники вложенных подгрупп доступа не получают.

Прекращение действия доступа

Права проверяются в момент каждой операции, поэтому изменения членства применяются сразу:

- если пользователь перестал быть участником проекта, его доступ к защищённой ветке перестаёт действовать;
- если приглашение группы в проект отозвано или её роль понижена ниже `Developer`, доступ участников группы перестаёт действовать.

Запись при этом остаётся в списке доступа и снова начнёт действовать, если членство или приглашение будет восстановлено. Чтобы окончательно отозвать доступ, удалите пользователя или группу из списка в правиле ветки.

Разрешить force push

Переключатель «Разрешить force push» на странице правила разрешает принудительную отправку изменений (`git push --force`) в защищённую ветку. Пользователи, которым разрешён force push, определяются списком «Разрешено выполнять push и слияние». По умолчанию переключатель выключен, и force push запрещён для всех, включая мейнтейнеров и администраторов.

Особенности и ограничения

- Доступ, выданный на защищённую ветку, не меняет роль пользователя в проекте и не предоставляет ему дополнительные права. Он действует только на операции push и слияния в ветки, подпадающие под правило.
- Правило с шаблоном (например, `release/*`) применяется ко всем существующим и будущим веткам, имена которых соответствуют шаблону.
- Выбор пункта «Никто» отключает доступ по ролям, но не очищает списки пользователей, групп и ключей развёртывания. Перечисленные в них сущности сохраняют доступ.
- Настраивать правила веток могут только пользователи с ролью `Maintainer` и выше.

Правила отправки изменений

В Deckhouse Code представлен набор правил для контроля за соблюдением определённых политик при отправке кода в Git. Эти правила помогают поддерживать целостность репозитория, повышать качество кода и обеспечивать соответствие стандартам безопасности вашей организации.

Доступные правила

- ***Проверять email коммитера***

  Адрес электронной почты коммитера должен быть подтвержден в профиле Deckhouse Code.

- ***Запретить удаление тегов***

  Тег нельзя удалить через git push. Удаление через веб-интерфейс доступно.

- ***Проверка зарегистрированного пользователя***

  Автор коммита должен быть пользователем Deckhouse Code.

- ***Блокировать утечки секретов***

  Попытка закоммитить файлы, подпадающие под следующие шаблоны, будет отклонена:

  - `aws/credentials`,
  - `ssh/personal_rsa`, `ssh/personal_dsa`, `ssh/personal_ed25519`, `ssh/personal_ecdsa`, `ssh/personal_ecdsa_sk`, `ssh/personal_ed25519_sk`,
  - `ssh/server_rsa`, `ssh/server_dsa`, `ssh/server_ed25519`, `ssh/server_ecdsa`, `ssh/server_ecdsa_sk`, `ssh/server_ed25519_sk`,
  - `id_rsa`, `id_dsa`, `id_ed25519`, `id_ecdsa`, `id_ecdsa_sk`, `id_ed25519_sk`,
  - файлы с расширением `.pem` или `.key`,
  - файлы `.history` или `_history`,
  - файлы с расширением `.keystore` или `.jks`,
  - `keystore.p12`, `keystore.pfx`.

- ***Коммит должен быть подписан GPG***

Разрешены только коммиты, подписанные GPG-ключом. Изменения через веб-интерфейс будут отклоняться, если для веб-UI не настроена подпись GPG.

- ***В сообщении коммита должен присутствовать DCO (Сертификат происхождения разработчика)***

  Коммиты, включая merge-коммиты и squash-коммиты, должны содержать строку DCO «Signed-off-by». При включении этого правила в веб-интерфейсе будет недоступна опция отката запроса на слияние, поскольку автоматически сгенерированное сообщение коммита не содержит подписи DCO.

- ***Проверять имя коммитера***

  Имя коммитера должно совпадать с именем профиля в Deckhouse Code.

- ***Регулярное выражение для сообщения коммита***

  Коммиты должны соответствовать указанному шаблону регулярного выражения. Пустое поле отключает правило.

- ***Регулярное выражение для email автора коммита***

  Email автора должен совпадать с указанным регулярным выражением. Пустое поле отключает правило.

- ***Регулярное выражение для имени файла***

  Имена файлов должны соответствовать указанному шаблону регулярного выражения. Пустое поле отключает правило.

- ***Регулярное выражение для имени ветки***

  Имя ветки должно совпадать с указанным регулярным выражением. Пустое поле отключает правило.

- ***Регулярное выражение для недопустимых сообщений коммитов***

  Если сообщение соответствует указанному шаблону регулярного выражения, коммиты будут отклонены. Пустое поле отключает правило.

- ***Максимальный размер файла (МБ)***

  Файлы, размер которых превышает указанный предел (в мегабайтах), будут отклонены. Чтобы отключить правило, установите значение `0`.

Настройка правил отправки изменений

Чтобы настроить правила отправки изменений, откройте страницу проекта и перейдите в раздел «Настройки» → «Репозиторий» → «Правила отправки изменений».

Настройка правил доступна пользователям с ролью `Maintainer` и `Owner`.

Настройка правил на уровне группы или инстанса

Чтобы получить доступ к настройке правил на уровне группы, откройте страницу группы и выберите «Настройки» → «Репозиторий» → «Правила отправки изменений». Настройка на уровне группы доступна пользователям с ролью `Owner`.

Правила уровня инстанса применяются ко всем проектам инстанса и задаются администратором. Подробнее — в разделе Пользователи и доступ.

При изменении правила на уровне инстанса новые значения применяются ко всем группам и проектам.

При изменении правила на уровне группы новые значения будут применены ко всем нижестоящим группам и их проектам.

Правила, заданные на уровне группы или инстанса, используются как настройки по умолчанию для проектов. При создании проекта будут включены те правила, которые заданы в его группе или на уровне инстанса (если проект создан в пользовательском пространстве).

Запрет на переопределение правил

Можно запретить переопределение правил в нижестоящих проектах и группах. Для этого снимите галочку «Разрешить переопределять на уровне группы/проекта». Такое правило нельзя изменять в нижестоящих группах и проектах. На уровне инстанса эта настройка применяется ко всем группам и проектам.

Примеры:

1. Если на уровне группы включить правило *«Проверять email коммитера»* и запретить его переопределение, это правило будет включено во всех проектах группы и подгрупп, без возможности его отключить.
2. Если на уровне группы выключить это же правило и запретить его переопределение, оно будет отключено во всех проектах группы и подгрупп, также без возможности изменения.
3. Если разрешено переопределение правила, оно будет автоматически обновляться при изменениях на уровне предка, но в нижестоящих группах и проектах его можно будет изменить.

Иерархия применения выглядит так:
Инстанс → Группа → Подгруппа → Проект

Pull-зеркалирование

Чтобы настроить зеркалирование репозитория, на странице проекта перейдите в «Настройки» → «Репозиторий» → «Зеркалирование репозиториев».

Если репозиторий пуст, сначала выполните его импорт. При зеркалировании запускаются все хуки, и загрузка большого репозитория может существенно нагрузить систему.

Настройка pull-зеркалирования репозитория

> ⚠ Для одного проекта можно настроить только одну задачу pull-зеркалирования. Задач push-зеркалирования может быть несколько.

Чтобы настроить pull-зеркалирование репозитория, выполните следующие шаги:

1. Перейдите на страницу проекта:
   - Откройте проект в интерфейсе Deckhouse Code.
   - В левом меню выберите «Настройки» → «Репозиторий».
   - Прокрутите до секции «Зеркалирование репозиториев».
2. Укажите URL-адрес репозитория. Учётные данные в URL-адресе игнорируются. Для авторизации используйте поля в блоке «Метод аутентификации» ниже.
3. Настройте аутентификацию:
   - Если подключение осуществляется по HTTP(S), в поле «Метод аутентификации» выберите «Имя пользователя и пароль» и укажите:
     - «Имя пользователя» — имя пользователя;
     - «Пароль» — пароль или токен доступа.
   - Если используется SSH-зеркалирование, укажите имя пользователя (обычно `git`). После сохранения конфигурации Deckhouse Code сгенерирует SSH-ключ, который будет использоваться для доступа.

Фильтр веток

Блок «Фильтр веток» в настройках зеркалирования определяет, какие ветки загружаются из исходного репозитория:

- «Зеркалировать все ветки» — зеркалируются все ветки исходного репозитория.
- «Зеркалировать только защищённые ветки» — зеркалируются только ветки, защищённые в этом проекте. Параметр недоступен, пока в проекте нет ни одной защищённой ветки (учитываются также защищённые ветки родительской группы). Подробнее — в разделе «Защищённые ветки».
- «Зеркалировать совпадающие ветки» — зеркалируются только ветки, имена которых соответствуют указанному регулярному выражению (синтаксис RE2). Выражение сопоставляется с именем ветки целиком. При вводе выражения форма показывает, сколько веток этого репозитория ему соответствует.

Влияние фильтра веток на синхронизацию тегов

> ⚠ Фильтр веток действует и на теги. Если выбран параметр «Зеркалировать только защищённые ветки» или «Зеркалировать совпадающие ветки», тег исходного репозитория синхронизируется только в том случае, если его коммит есть в одной из веток этого репозитория. Остальные теги пропускаются.

Учитывайте следующие особенности:

- При выборе параметра «Зеркалировать все ветки» синхронизируются все теги исходного репозитория.
- Тег, пропущенный из-за фильтра, не теряется: он проверяется заново при каждой синхронизации, и как только его коммит попадает в одну из зеркалируемых веток, тег создаётся в зеркале.
- Проверка выполняется по всем веткам репозитория-зеркала, включая ветки, созданные непосредственно в Deckhouse Code, а не только по веткам, попадающим под фильтр.
- Уже синхронизированные теги остаются в репозитории, даже если фильтр веток впоследствии изменён так, что больше их не охватывает.

Импорт репозитория по URL с настройкой pull-зеркалирования

Вы можете настроить pull-зеркалирование во время импорта репозитория по URL: проект будет создан и сразу настроен как pull-зеркало исходного репозитория. При этом не требуется настраивать зеркалирование отдельно после импорта.

> ℹ Зеркалирование репозиториев должно быть включено администратором на уровне инстанса. В «Области администратора» перейдите в «Настройки» → «Репозиторий» → «Зеркалирование репозитория» (настройка `mirror_available`). Если зеркалирование не включено на уровне инстанса, флажок «Зеркалировать репозиторий» отображается неактивным (недоступен для выбора).

Чтобы импортировать репозиторий по URL с настройкой pull-зеркалирования, выполните следующие шаги:

1. Создайте новый проект:
   - Перейдите в «Новый проект» → «Импортировать проект» → «Репозиторий по URL».
   - Укажите URL-адрес исходного репозитория.
   - Если для доступа к исходному репозиторию требуется аутентификация, укажите учётные данные.
2. Установите флажок «Зеркалировать репозиторий».
3. Создайте проект.

При первоначальном импорте репозиторий создаётся и наполняется данными из источника. После этого проект автоматически поддерживается в актуальном состоянии по расписанию pull-зеркалирования (подробнее — в разделе «Работа расписания и обработка ошибок» ниже).

Чтобы ограничить зеркалирование защищёнными ветками или ветками, подходящими под шаблон, после импорта настройте фильтр веток на странице проекта «Настройки» → «Репозиторий» → «Зеркалирование репозиториев» (подробнее — в разделе «Фильтр веток» выше).

Также вы можете превратить существующий проект в pull-зеркало, выполнив в него импорт по URL; для этого требуется роль Maintainer. В обоих случаях зеркалирование выполняется от имени пользователя, который его настроил.

Работа расписания и обработка ошибок

- Назначение задач pull-зеркалирования выполняется один раз в час (`Projects::PullMirrorScheduleWorker`).
- Каждое зеркало обновляется не чаще одного раза в 6 часов.
- Если задача зеркалирования завершилась с ошибкой, следующая попытка будет выполнена при следующем запуске `Projects::PullMirrorScheduleWorker`. Максимальное количество попыток при ошибках — **5**. Нажатие кнопки «Обновить сейчас» сбрасывает счётчик неуспешных попыток.
- Если задача Sidekiq завершилась аварийно (например, при перезапуске Sidekiq), её статус будет обновлён через 3 часа, после чего будет добавлена новая попытка синхронизации.

Особенности работы с LFS (Large File Storage)

При pull-зеркалировании LFS-объекты загружаются **только в том случае**, если LFS включён в целевом проекте Deckhouse Code.

CODEOWNERS

Функция **CODEOWNERS** позволяет определить ответственных за отдельные части репозитория. Ответственными могут быть пользователи и группы. Изменения, затрагивающие файлы, указанные в `CODEOWNERS`, должны получить апрув от владельцев кода.

Использование CODEOWNERS помогает:

- требовать апрувы от доменных экспертов для важных директорий;
- упростить процесс поиска «ответственных» за конкретный участок кода.

CODEOWNERS дополняет механизм правил апрува, но не заменяет их. В отличие от правил апрува, которые задаются вручную в UI, CODEOWNERS работает через файл `CODEOWNERS` в репозитории.

Как работает CODEOWNERS

`CODEOWNERS` — текстовый файл в репозитории. Deckhouse Code использует его для определения владельцев файлов, участвующих в запросе на слияние.

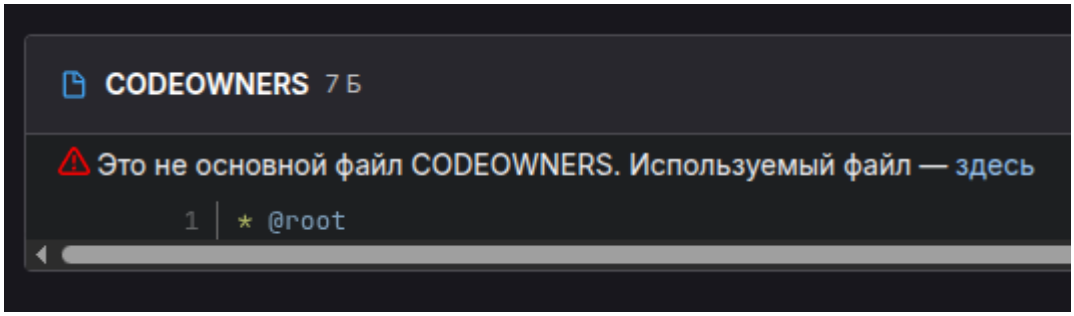
Сопоставляются те файлы, которые показывает вкладка «Изменения» запроса на слияние: итоговая разница между базой и вершиной запроса на слияние, а не файлы отдельных коммитов. Переименованный файл учитывается и по старому, и по новому пути, поэтому применяется запись, которая владеет любым из них.

Ищется в трёх местах (по приоритету):

1. `./CODEOWNERS`.
2. `./docs/CODEOWNERS`.
3. `./.gitlab/CODEOWNERS`.

Используется **только первый найденный файл**.

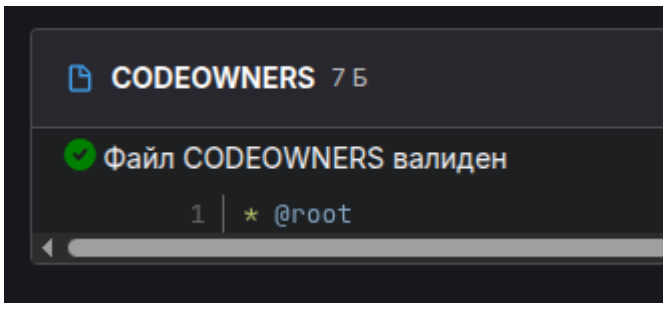
Если в проекте существует несколько файлов и вы сейчас просматриваете неприоритетный, то увидите соответствующее сообщение.



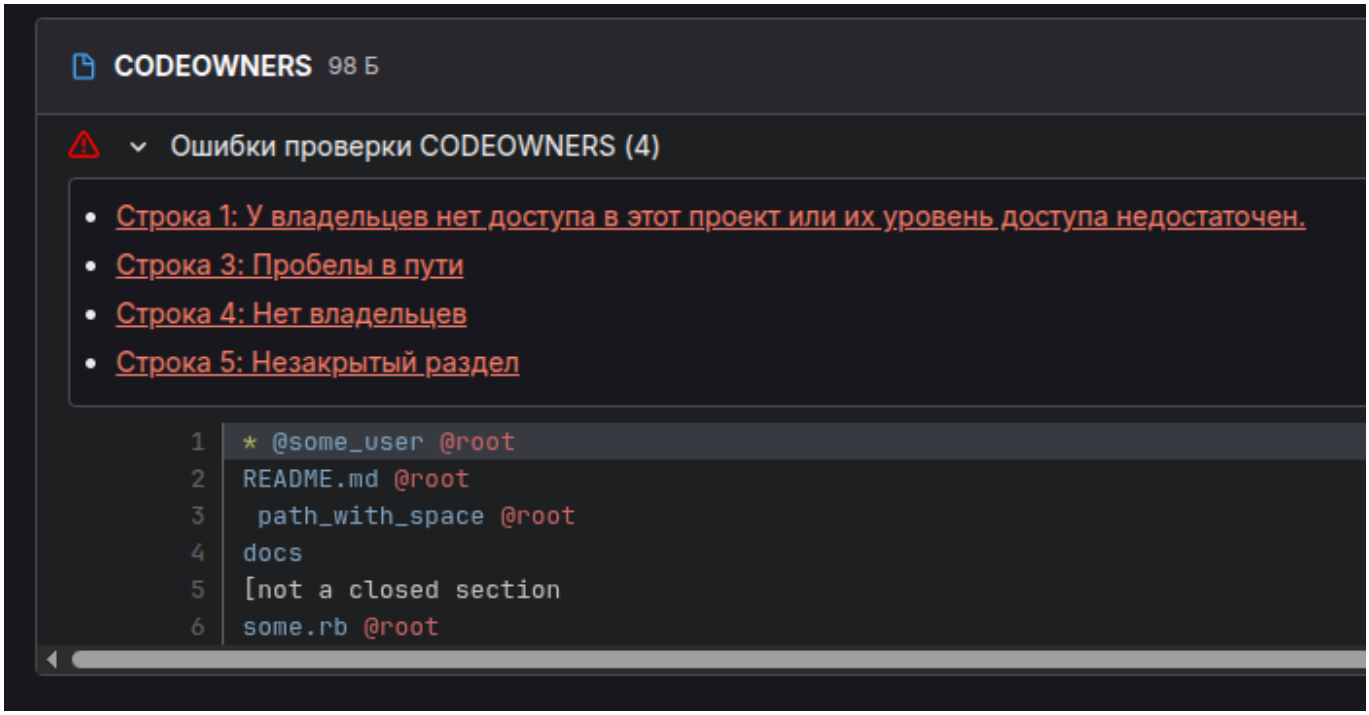
Валидация файла CODEOWNERS

Валидация помогает увидеть ошибки в файле на этапе редактирования.

Если файл валиден, отображается соответствующее сообщение:



Если файл невалиден, будут отображены проблемные строки и тип ошибки:



Валидация сообщает о следующих типах ошибок:

- *У владельцев нет доступа в этот проект или их уровень доступа недостаточен* — как минимум один из владельцев не имеет достаточных прав в проекте.
- *Пробелы в пути* — если путь содержит пробелы, нужно их экранировать символом `\`.
- *Нет владельцев* — не указано ни одного владельца.
- *Незакрытый раздел* — не хватает символа закрытия раздела `]`.

Где применяется CODEOWNERS

Если MR, изменяющий файл, попадает под правила CODEOWNERS, то необходимо получить апрувы от владельцев.

Формат CODEOWNERS

Строка правила:

```
<путь или паттерн> <список владельцев>
```

Владельцы:

- `@user`;
- `@group`;
- `@group/subgroup`;
- роли: `@@developer`, `@@maintainer`, `@@owner`.

Примеры

```
# Владелец всех файлов
* @default-team

# README.md только в корне
/README.md @docs-team

# Все Ruby-файлы
*.rb @backend-team

# Вся директория config
/config/ @devops

# README.md в любом месте
README.md @docs
```

Паттерны путей

Deckhouse Code использует подстановочные символы:

- `*` — любые символы кроме `/`.
- `**` — матч нескольких уровней директорий.
- `/dir/` — только указанная директория.
- `*.md` — все Markdown-файлы.
- `/config/**/*.rb` — все Ruby-файлы внутри `config/*` на всех уровнях.

Примеры паттернов

```
/docs/*       @docs-one      # один уровень
/docs/**      @docs-two      # рекурсивно
/app/**/*.rb  @ruby-team
```

Секции CODEOWNERS

В файле `CODEOWNERS` секции — это именованные области, которые анализируются отдельно и всегда применяются. Пока вы не определите ни одной секции, Deckhouse Code рассматривает весь файл `CODEOWNERS` как одну единственную секцию.

Особенности использования `CODEOWNERS`:

- Deckhouse Code рассматривает записи без секций, включая правила, определённые до первой секции, как отдельную, безымянную секцию.
- Каждая секция обрабатывает свои правила отдельно.
- Если путь к файлу соответствует нескольким записям внутри одной секции, используется только последняя подходящая запись в этой секции.

- Если путь к файлу соответствует записям в нескольких секциях, используется последняя подходящая запись в каждой секции.

Например, в файле `CODEOWNERS` со следующими секциями, определяющими владельцев для README:

```
* @root

[README Owners]
README.md @user1 @user2
internal/README.md @user4

[README other owners]
README.md @user3
```

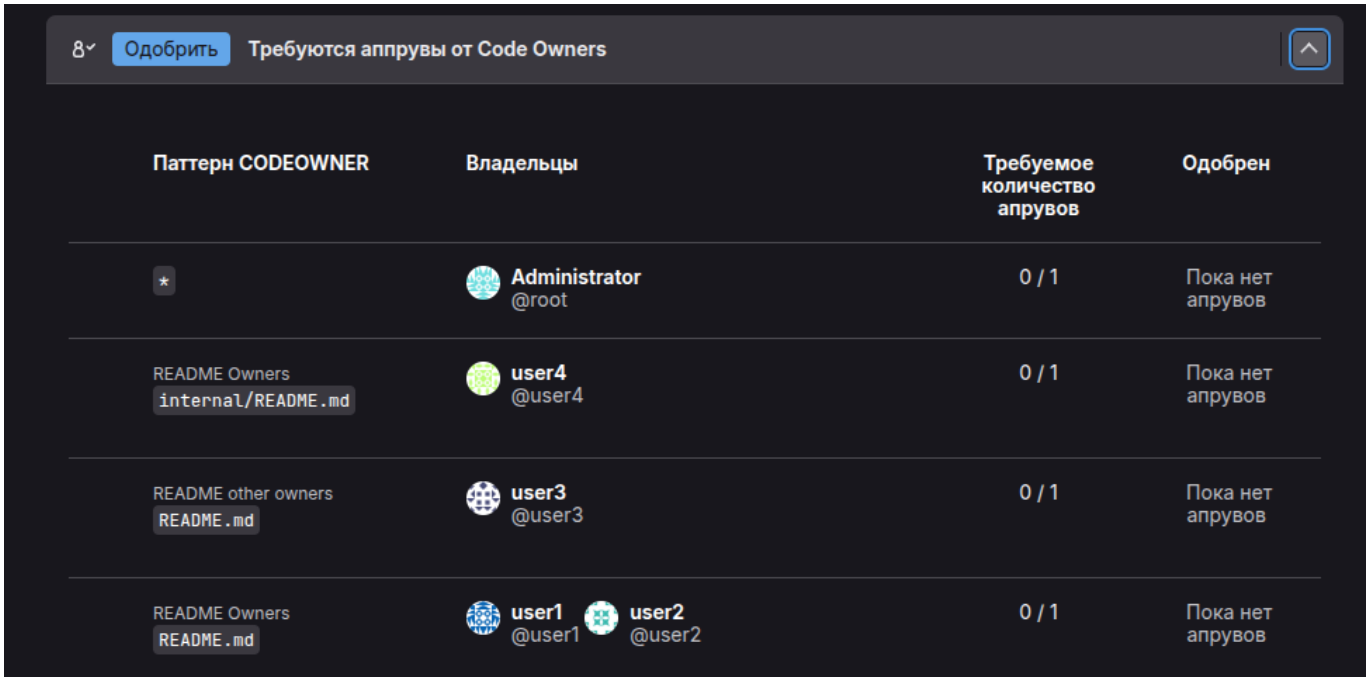
Владельцы для `README.md` в корневой директории:

- `@root` — из безымянной секции.
- `@user1` и `@user2` — из секции `[README Owners]`.
- `@user3` — из секции `[README other owners]`.

Владельцы для `internal/README.md`:

- `@root` — из безымянной секции.
- `@user4` — из секции `[README Owners]`. (И `README.md`, и `internal/README.md` подходят под правила секции, но используется только последняя подходящая запись в этой секции).
- `@user3` — из секции `[README other owners]`.

В виджете запроса на слияние каждый владелец кода отображается под своим ярлыком.



Детальный пример: сложная комбинация секций

Ниже — пример реального промышленного файла, показывающий:

- секции с разным количеством апрувов;
- переопределение владельцев;
- опциональные секции;
- исключения;
- наследование секций с одинаковыми именами.

```
# Глобальные настройки
* @fallback-team
!*.lock                    # lock-файлы не требуют апрувов
!**/generated/**           # автоматическая генерация

[Backend][2] @backend-core
# Требуются 2 апрува от backend-core для любого Ruby-кода
app/**/*.rb

# Но для критических моделей определяем другой порядок
app/models/**/*.rb @backend-core @security-team

# Эта модель — исключение, владелец другой
app/models/legacy/**/*.rb @migration-team

[Backend]
# Дополнение секции: теперь Backend включает также SQL
db/**/*.sql @db-team

[Ruby Optional]
# Дополнительная подсветка владельцев, апрувы НЕ обязательны
^[Ruby Optional]
*.rb @ruby-advisors

[Frontend][3]
# Для UI требуется 3 апрува
*.vue @frontend-team
*.js  @frontend-team

# переопределение: конкретный файл → конкретный человек
frontend/critical_entry.vue @frontend-lead

[Docs]
*.md @technical-writers

[Docs]
# переопределение: README всегда принадлежит docs-lead
README.md @docs-lead
```

Особенности примера:

- секция `[Backend]` объявлена дважды → Deckhouse Code объединит её;
- `[Backend][2] @backend-core` задаёт **обязательные 2 апрува**;
- `[Frontend][3]` требует **3 апрува** от фронтенда;
- `[Ruby Optional]` отмечена как опциональная — владельцы видны, но их апрув не обязателен;
- исключения `!*.lock` и `!**/generated/**` означают, что эти файлы вообще не требуют апрувов.

Исключения (!)

Используются для исключения файлов внутри секции:

```
* @default
!package-lock.json
!**/generated/**
```

После исключения файл **не может быть снова включён** в ту же секцию.

Порядок обработки правил

Правила обрабатываются в соответствии со следующим порядком:

- Deckhouse Code читает файл сверху вниз.
- В пределах одной секции правила с одинаковым путём **переопределяют** предыдущие.
- Если файл подходит под несколько секций — владельцы из всех этих секций добавляются.
- Секции не переопределяют друг друга — они *независимы*.

Кто может быть владельцем (eligible owners)

1. Пользователи (через @username)

Пользователь считается валидным владельцем, если:

- Имеет **прямое** членство в проекте (Developer, Maintainer или Owner).
- Имеет членство в группе проекта (включая наследуемое: группа проекта, родительские группы, предки родительских групп).
- Является прямым или унаследованным участником группы, которая напрямую приглашена в проект.
- Является прямым но не унаследованным участником группы которая приглашена в группу проекта.
- Является прямым но не унаследованным участником группы которая приглашена в предка группы проекта.

Если подытожить, то подходят все пользователи которые отображаются на странице участников проекта с ролью developer или выше.

Не считается владельцем, если:

- заблокирован,
- доступ отозван,
- роль ниже Developer.

2. Группы (через @group или @group/subgroup)

Группа может быть владельцем **только если:**

- Она напрямую приглашена в проект с ролью Developer+ (через «Пригласить группу»).
- Владельцами считаются только *прямые участники* этой группы.

Наследуемые участники родительских групп владельцами не считаются.

3. Роли (через @@role)

Формат:

```
@@developer
@@maintainer
@@owner
```

Особенности:

- учитываются **только прямые участники проекта**;
- члены групп в расчёт не входят;
- роли не наследуют друг друга (`@@developer` НЕ включает maintainer'ов).

Можно указывать несколько ролей в одной строке:

```
file.md @@developer @@maintainer
```

Взаимодействие с правилами апрува

CODEOWNERS и правила апрувов работают независимо, но их требования суммируются.

Пример:

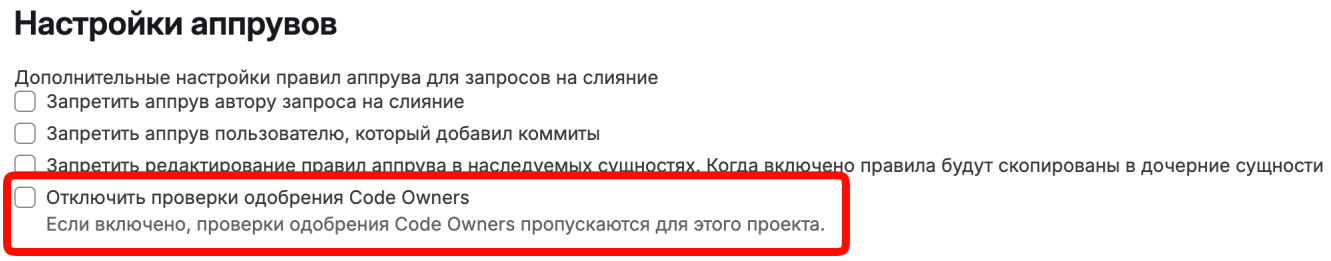
- правило `Security` требует 1 апрув;
- секция `[Backend][2]` требует 2 апрува;
- MR меняет backend-файл.

Итог: необходимо **3 апрува**.

Отключение проверок CODEOWNERS

Проверки CODEOWNERS, которые включены в Deckhouse Code по умолчанию, можно отключить на уровне проекта. Это удобно, когда в проекте есть файл `CODEOWNERS`, но требовать апрувы от владельцев кода не нужно.

Чтобы отключить проверки, перейдите в раздел «Проект» → «Настройки» → «Запросы на слияние» → раздел дополнительных настроек правил апрува и выберите настройку «Отключить проверки одобрения Code Owners».



При отключении проверки CODEOWNERS пропускаются во всём проекте:

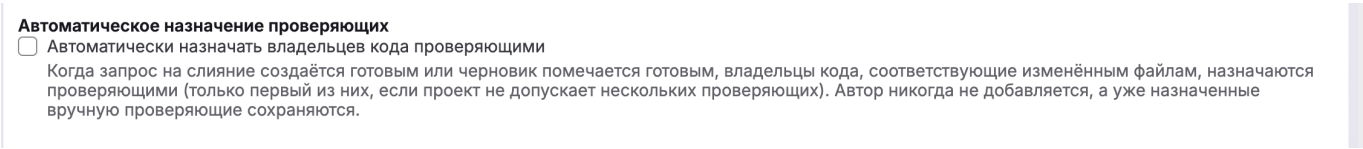
- запросы на слияние не блокируются из-за отсутствующих апрувов от владельцев кода;
- владельцы кода не определяются для изменённых файлов — список применимых записей пуст;
- изменение настройки сразу применяется к уже открытым запросам на слияние.

Проверка синтаксиса файла `CODEOWNERS` при этом продолжает работать. Файл можно валидировать даже при отключённых проверках.

Автоматическое назначение владельцев кода проверяющими

Владельцев кода, соответствующих изменённым файлам, можно автоматически назначать проверяющими запроса на слияние.

Для этого перейдите в раздел «Проект» → «Настройки» → «Запросы на слияние» → «Автоматическое назначение проверяющих» и выберите настройку «Автоматически назначать владельцев кода проверяющими».



По умолчанию настройка выключена. При её включении владельцы кода назначаются проверяющими:

- при создании запроса на слияние сразу в готовом статусе (не в статусе черновика);
- при снятии статуса черновика с запроса на слияние.

При автоматическом назначении действуют следующие ограничения:

- для черновиков назначение не выполняется;
- если у запроса на слияние уже есть проверяющие, автоматическое назначение не выполняется — назначенные вручную проверяющие сохраняются;
- автор запроса на слияние не может быть назначен проверяющим;
- назначаются только пользователи, у которых есть доступ на чтение запроса на слияние;

- если в проекте запрещено назначать нескольких проверяющих, назначается только первый владелец кода.

Рекомендации по оформлению

При оформлении файла `CODEOWNERS` следуйте рекомендациям:

- Сначала задавайте широкие правила (`* @team`). Далее уточняющие (`app/**/*.rb @backend-team`).
- Используйте секции для лучшей структуры и наглядности.
- Избегайте дубликатов — последнее правило в секции имеет приоритет.
- Опциональные секции полезны для подсветки экспертов без обязательных апрувов.

Пример финального файла

```
# Глобальные владельцы
* @default-team

# Исключения
!yarn.lock
!**/generated/**

[Backend][2]
app/**/*.rb @backend-core

[Frontend][3]
*.vue @frontend-team
*.js @frontend-team

^[Docs]
*.md @docs-team

[Critical Overrides]
/config/production.yml @ops-lead
```

Групповая аналитика репозитория

Отчёт «Аналитика репозитория» доступен на уровне группы. Он показывает покрытие кода тестами в проектах группы и её подгрупп. Данные появляются после завершения в ветке по умолчанию пайплайна, который публикует покрытие.

Доступность отчёта

Отчёт доступен участникам группы с ролью «Наблюдатель» и выше, а также пользователям с типом «Аудитор». Гости и пользователи, не выполнившие вход, не могут открыть отчёт, включая отчёт публичной группы.

Просмотр отчёта

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика репозитория».

Также отчёт можно открыть по прямому адресу:

```
https://<HOST>/groups/<GROUP>/-/analytics/repository_analytics
```

Покрытие кода тестами

Раздел «Покрытие кода тестами» показывает данные за последний день, в котором есть данные о покрытии:

- «Проекты с покрытием» — число проектов с данными о покрытии;
- «Среднее покрытие по заданиям» — среднее покрытие по задачам, которые его публикуют;
- «Задания с покрытием» — число задач, публикующих покрытие.

Подпись «Обновлено» указывает этот день.

Среднее покрытие тестами

График «Среднее покрытие тестами» показывает среднее покрытие всех задач по дням за последние 30 дней. Дни без данных о покрытии на графике не отмечены.

Наведите указатель на точку графика, чтобы увидеть дату, процент покрытия и число задач с покрытием.

Последние результаты покрытия тестами

Таблица «Последние результаты покрытия тестами» по умолчанию показывает до 20 проектов с покрытием. Строки отсортированы в алфавитном порядке. Чтобы заменить этот набор, выберите один или несколько проектов в списке «Проекты». В список входят проекты подгрупп.

Колонка	Содержимое
«Проект»	Название проекта со ссылкой на его график покрытия
«Покрытие»	Среднее покрытие по задачам проекта
«Задания с покрытием»	Число задач, публикующих покрытие
«Последнее обновление»	Дата последних данных о покрытии

Скачивание данных о покрытии

Отчёт позволяет скачать исторические данные о покрытии для всех проектов группы и её подгрупп или для выбранных проектов. CSV-файл содержит до 1000 последних записей в порядке от новых к старым. Каждая строка содержит дату, имя группы задач, название проекта и процент покрытия.

Чтобы скачать CSV-файл:

1. Нажмите «Скачать исторические данные покрытия тестами (.csv)».
2. В списке «Проекты» выберите проекты или оставьте «Все проекты».
3. В поле «Диапазон дат» выберите 7, 14, 30, 60 или 90 дней. По умолчанию выбрано 30 дней.
4. Нажмите «Скачать данные покрытия тестами (.csv)».

Аналитика репозитория проекта

Отчёт «Аналитика репозитория» доступен на уровне проекта. Он показывает языки программирования в ветке репозитория по умолчанию, а также покрытие кода тестами и статистику коммитов для выбранной ветки или тега.

Доступность отчёта

Отчёт доступен пользователям, которые могут скачать код проекта. В проекте должен быть инициализированный репозиторий и хотя бы один коммит в ветке по умолчанию. Коммиты вики в отчёт не входят.

Просмотр отчёта

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Аналитика репозитория».

Также отчёт можно открыть по прямому адресу:

```
https://<HOST>/<GROUP>/<PROJECT>/-/graphs/<REF>/charts
```

Чтобы посмотреть покрытие кода тестами и статистику коммитов для другой ветки или тега, выберите её или его в списке рядом с «Статистика коммитов». График языков программирования всегда использует ветку по умолчанию.

Языки программирования

График «Языки программирования, используемые в этом репозитории» показывает долю кода каждого языка программирования в ветке проекта по умолчанию. Расчёт использует размер кода в байтах и исключает сгенерированный и сторонний код.

Покрытие кода тестами

График «Статистика покрытия кода» показывает данные о покрытии выбранной ветки или тега за последние 90 дней. Данные появляются после завершения пайплайна с данными о покрытии.

Если покрытие публикуют несколько групп задач, выберите их название в списке над графиком. Наведите указатель на точку графика, чтобы увидеть дату и процент покрытия. Если на графике есть данные, нажмите «Скачать исходные данные (.csv)», чтобы скачать их.

Данные о покрытии доступны только пользователям, которые могут просматривать пайплайны и результаты задач проекта. Для публичного проекта пользователь, не выполнивший вход, может просматривать покрытие при включённых публичных пайплайнах.

Статистика коммитов

Раздел «Статистика коммитов» использует до 2000 коммитов без коммитов слияния из выбранной ветки или тега. Он показывает общее число коммитов, среднее число за день и число авторов, а затем графики коммитов:

- по дням месяца;
- по дням недели;
- по часам в UTC.

Аналитика участников

Отчёт «Аналитика участников» доступен на уровне проекта. Он показывает, как коммиты в выбранную ветку или тег распределены по времени, суммарно и по каждому автору.

Доступность отчёта

Отчёт доступен пользователям, которые могут скачать код проекта: участникам с ролью «Планировщик» или «Наблюдатель» и выше, в том числе тем, кто получил одну из них в родительской группе, и пользователям с типом «Аудитор». В публичном проекте отчёт открыт любому посетителю, включая не выполнившего вход, а во внутреннем — любому вошедшему пользователю.

У проекта должен быть репозиторий хотя бы с одним коммитом. Открытие отчёта проекта с пустым репозиторием перенаправляет на страницу проекта.

Выключение функции «Репозиторий» или «Аналитика» в настройках проекта закрывает отчёт и убирает пункт «Аналитика участников» из меню «Аналитика».

Просмотр отчёта

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Аналитика участников».

Также отчёт можно открыть по прямому адресу:

```
https://<HOST>/<GROUP>/<PROJECT>/-/graphs/<REF>
```

Отчёт открывается для ветки проекта по умолчанию или для ветки, выбранной в проекте в момент перехода. Чтобы посмотреть другую ветку или тег, выберите их в списке над графиком. Кнопка «История» рядом со списком открывает список коммитов выбранной ветки или тега.

График коммитов

График «Коммитов в» содержит по точке на каждый день от первого до последнего из загруженных коммитов и показывает, сколько коммитов сделано в этот день. Вертикальная ось — «Количество коммитов». При наведении на график показываются дата и число коммитов.

Подпись над графиком: «За исключением merge-коммитов. Ограничено 6000 коммитами.» Отчёт загружает 6000 последних коммитов выбранной ветки или тега и пропускает коммиты слияния. В репозитории с большим числом коммитов график начинается с самого старого из загруженных.

Ползунок под графиком сужает период. Суженный период применяется и к графикам авторов, и к их счётчикам коммитов.

Графики авторов

Под графиком «Коммитов в» отчёт показывает по графику на каждого автора, до 100 авторов, в порядке убывания числа коммитов. Каждый график озаглавлен именем автора и строкой `128 коммитов (ivan.petrov@example.com)` и показывает коммиты этого автора по дням на той же временной оси, что и график «Коммитов в».

Автор определяется по адресу электронной почты в коммитах без учёта регистра. Коммиты с одним адресом и разными именами показаны под именем из самого свежего коммита, а коммиты одного человека с разных адресов — как разные авторы. Адрес с учётными записями пользователей установки отчёт не сопоставляет.

Как считаются коммиты

Коммит относится к дню своей даты коммита в часовом поясе, записанном в коммите. Коммит без часового пояса отсчитывается по UTC. Дата коммита — это дата последнего изменения коммита, поэтому коммит после rebase или amend относится к дню этого изменения.

Запросы на слияние

Ревью кода

Merge Requests (MR) — запросы на слияние

Ключевой инструмент обеспечения качества кода перед его объединением в основную ветку.

Основные возможности:

- Создание MR после коммита в функциональную ветку.
- Автоматическое сравнение изменений с целевой веткой.

- Встроенные обсуждения и ревью кода.
- Поддержка авто-слияния после успешных проверок (CI/CD, ревью).
- Визуальное отображение конфликтов и средства для их разрешения.

Просмотр изменений (Diff View)

Позволяет визуально сравнивать версии кода.

Поддерживаемые режимы:

- Стандартный diff — отображение изменений по коммитам.
- Side-by-side diff — сравнение строк кода построчно.
- Inline diff — подсветка изменений внутри строк..

Комментарии и обсуждения

Инструменты совместной работы над изменениями в коде.

Возможности:

- Комментирование отдельных строк кода.
- Ведение обсуждений с упоминанием участников.
- Пометка обсуждений как разрешённых (Mark as resolved).
- Автоматическое формирование задач по комментариям.

Уведомления о событиях

Система оповещений о событиях в проекте.

Поддержка:

- Email-уведомлений о новых коммитах, MR и комментариях.
- Интеграция с внешними мессенджерами: Slack, Mattermost, Telegram (настраиваемые уведомления).

Подтверждение Merge Request’ов (Approvals)

Механизм контроля качества через обязательное одобрение MR перед слиянием.

Возможности:

- Настройка минимального числа одобрений.
- Назначение обязательных ревьюеров.
- Поддержка auto-approval по заданным условиям.
- Блокировка слияния при отсутствии требуемых подтверждений.
- Интеграция с механизмом Codeowners для обязательного ревью владельцами кода.
- Автоматический сброс подтверждений при изменении содержимого MR.
- Логирование всех действий, связанных с процессом одобрения.

Codeowners

Механизм назначения ответственных за участки кодовой базы.

Функциональность:

- Определение владельцев для файлов и директорий проекта.
- Использование файла CODEOWNERS для описания ответственности.
- Автоматическое назначение владельцев в ревьюеры для MR.
- Интеграция с системой Approvals: владельцы становятся обязательными ревьюерами.
- Поддержка шаблонов путей (`*` , `**`) и групп владельцев.
- Возможность ограничить изменения в защищённых ветках только для владельцев.

Правила апрувов запросов на слияние

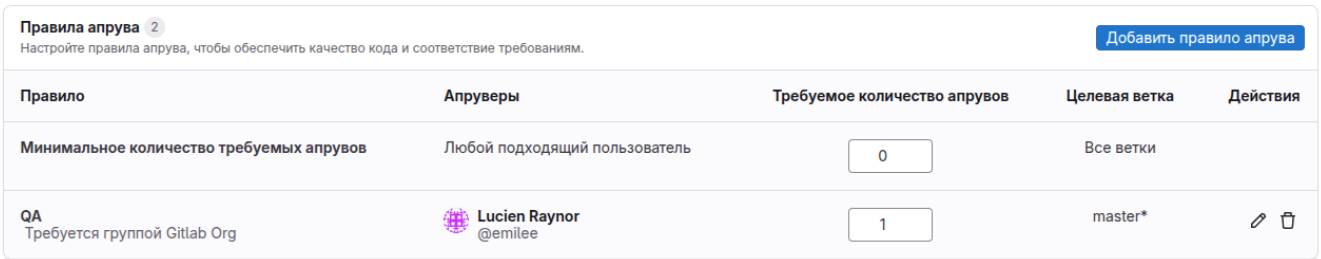
Правила апрувов помогают контролировать качество кода и обеспечивать обязательное ревью перед слиянием. С их помощью можно определить, сколько апрувов нужно получить, кто именно должен участвовать в проверке, и в каких случаях правило должно срабатывать.

Эти правила работают на уровнях группы, проекта и запросов на слияние, наследуются вниз по иерархии и позволяют централизованно управлять процессом ревью. Вместе с CODEOWNERS они формируют гибкую и надёжную систему контроля изменений, защищая важные части репозитория от нежелательных или непросмотренных правок.

Определение правил

Правила можно определить на уровне:

- Группы (доступно для роли **Maintainer**). Чтобы определить правила для группы, перейдите в «Группа» → «Настройки» → «Запросы на слияние» → «Правила апрува».
- Проекта (доступно для роли **Maintainer**). Чтобы определить правила для проекта, перейдите в «Настройки» → «Запросы на слияние» → «Правила апрува».
- Запроса на слияние (доступно для роли **Developer**). Чтобы определить правила для запроса на слияние, на странице создания или редактирования MR раскройте секцию **«Правила апрува»**.



Описание столбцов таблицы правил апрува

- **Правило** — содержит название правила и признак наследования. Запись «Требуется группой Example Org» означает, что правило создано в группе *Example Org*.
- **Апруверы** — содержит список пользователей и групп, чьи апрувы учитываются при проверке выполнения правила.
- **Требуемое количество апрувов** — количество апрувов, которое необходимо получить от пользователей или участников групп, указанных в столбце «Апруверы». Значение `Любой подходящий пользователь` означает, что будет засчитан апрув от любого пользователя, который имеет право его выполнить.
- **Целевая ветка** — правило применяется только если запрос на слияние направлен в указанную ветку. В качестве названия ветки можно использовать wildcard (`*`).

Наследование правил

Правила апрувов наследуются каскадно: группа → подгруппа → проект → запрос на слияние. Механизм наследования имеет следующие особенности:

- Когда создаётся подгруппа, проект или запрос на слияние, правила родительской сущности автоматически появляются в новой сущности как унаследованные. В таблице правил рядом с названием отображается, на каком уровне правило было изначально создано.
- При создании новых правил на уровне группы или проекта они автоматически добавляются во все существующие нижестоящие подгруппы и проекты. **В уже созданные запросы на слияние новые правила не добавляются**.
- Если изменить правило на родительском уровне, изменения автоматически применяются ко всем унаследованным правилам на нижестоящих уровнях. Например, если проект унаследовал правило из группы и в группе изменить количество требуемых апрувов, то это изменение отразится и в проекте.
- Если же изменить унаследованное правило в дочерней сущности, **оно превращается в самостоятельное правило**. То есть правило перестаёт быть связанным с родительским: на дочернем уровне остаётся собственная копия с новыми параметрами, и дальнейшие изменения правила в группе уже не затронут это правило в проекте.

Правила для группы

Правила, созданные на уровне группы, автоматически добавляются в новые проекты и подгруппы как унаследованные. Чтобы открыть их, перейдите в «Группа» → «Настройки» → «Запросы на слияние» → «Правила апрува».

Управлять правилами группы могут участники группы с ролью **Maintainer**.

В разделе «Правила апрува» отображается список правил, настроенных на уровне группы.

Создание правила в группе

Создать новое правило можно, нажав кнопку **«Добавить правило апрува»** и указав:

- Название правила.
- Целевую ветку (доступны пресеты «Все ветки», «Все защищённые ветки», «Ввести название ветки», либо wildcard `*`).
- Требуемое количество апрувов.
- Апруверов — прямых участников группы или группы на инстансе. При добавлении группы учитываются апрувы только от прямых участников добавленной группы.

Правила для проекта

Правила, созданные на уровне проекта, автоматически добавляются в новые запросы на слияние как унаследованные. Чтобы открыть их, перейдите в «Проект» → «Настройки» → «Запросы на слияние» → «Правила апрува».

Управлять правилами проекта могут участники проекта с ролью **Maintainer**.

Создание правила в проекте

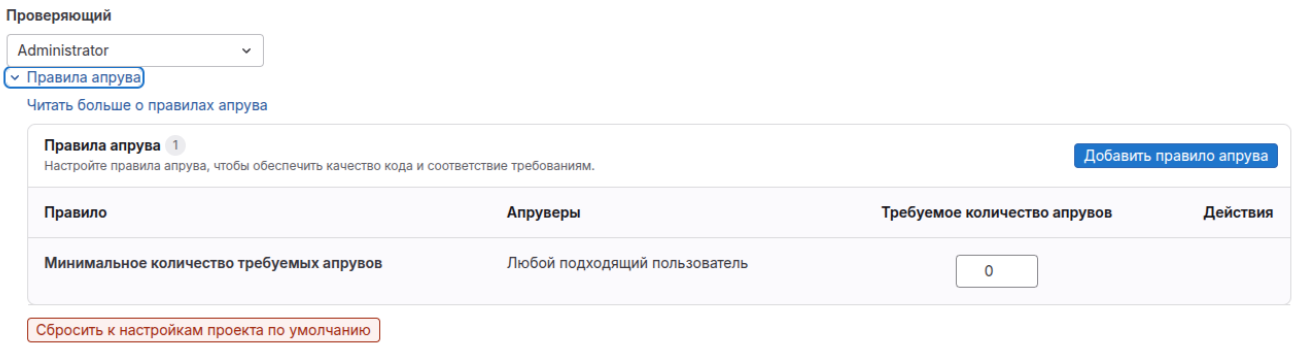
При создании нового правила можно указать:

- Название правила.
- Целевую ветку (можно выбрать пресет, указать название ветки или регулярное выражение; также можно выбрать защищённую ветку проекта).
- Требуемое количество апрувов.
- Апруверов — прямых или унаследованных участников проекта, а также группы, приглашённые в проект с ролью **Reporter**. Со стороны добавленной группы учитываются апрувы только от прямых участников группы.

Правила для запроса на слияние

Чтобы открыть правила запроса на слияние, на странице его создания или редактирования раскройте секцию **«Правила апрува»**.

Правилами запроса на слияние могут управлять пользователи с ролью **Developer**.



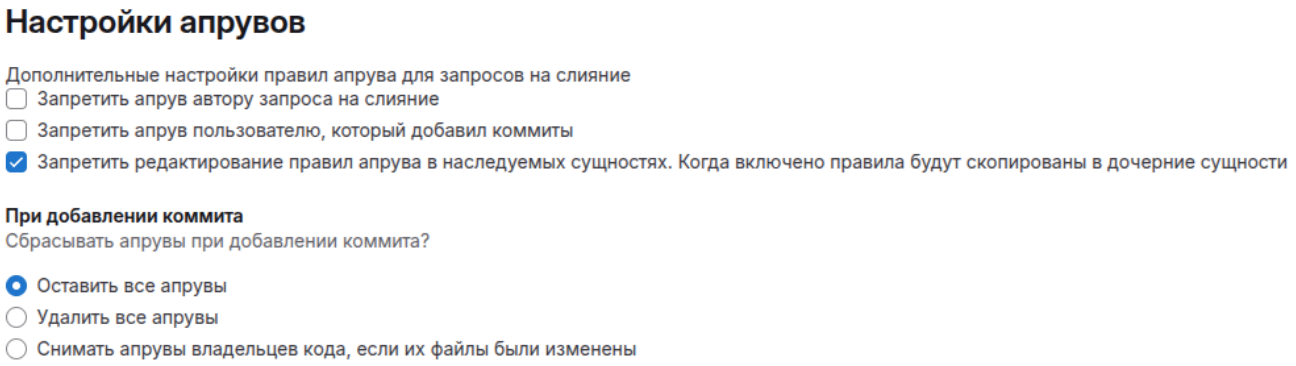
На странице MR доступна кнопка **«Сбросить к настройкам проекта по умолчанию»** — она удаляет текущие правила и добавляет все правила проекта, которые применимы для целевой ветки MR.

Создание правила в MR

При создании правила можно указать:

- Название правила.
- Требуемое количество апрувов.
- Апруверов — прямых или унаследованных участников проекта, а также группы с ролью **Reporter** или выше. Со стороны группы учитываются апрувы только от прямых участников группы.

Дополнительные настройки апрувов



На уровне проекта или группы можно настроить дополнительные параметры:

- **Запретить апрув автору запроса на слияние** — автор MR не может делать апрув.
- **Запретить апрув пользователю, который добавил коммиты** — коммиттеры не могут делать апрувы.
- **Запретить редактирование правил в наследуемых сущностях** — при включении:
  - правила копируются в дочерние сущности и становятся недоступны для редактирования или удаления;
  - нижестоящие группы и проекты могут создавать только собственные правила;
  - в существующие MR правила автоматически не добавляются;
  - текущие настройки апрувов также распространяются на все дочерние сущности.
- **При добавлении коммита** — секция определяет, должны ли сбрасываться существующие апрувы при появлении нового коммита.

Кто может делать апрувы

Если настройки апрувов этого не запрещают, апрувы могут делать:

- все участники проекта с ролью **Developer** и выше;

- участники с ролью **Planner** и выше, если они назначены ответственными в запросе на слияние.

Внешние проверки статуса в запросах на слияние

Внешние проверки статуса позволяют мейнтейнерам проекта отправлять данные запроса на слияние во внешний сервис и использовать ответ сервиса как дополнительное условие слияния. Используйте их, если перед слиянием запрос должен пройти внешнюю проверку соответствия требованиям, контроль качества или проверку безопасности.

Внешние проверки статуса настраиваются отдельно для каждого проекта и не наследуются другими проектами.

Как работают внешние проверки статуса

Когда происходит событие запроса на слияние, Deckhouse Code отправляет информацию о запросе во все внешние сервисы, проверки статуса через которые применяются к целевой ветке. Внешний сервис проверяет запрос на слияние и отправляет результат обратно в Deckhouse Code.

Результат проверки всегда относится к текущему `HEAD` SHA исходной ветки. Если в запрос на слияние добавлены новые коммиты, предыдущий результат больше не относится к новому состоянию, и Deckhouse Code ожидает новый результат.

Внешние проверки статуса влияют на возможность слияния только если в проекте включена настройка **Проверки статуса должны быть успешными**. Если настройка выключена, виджет продолжает показывать результаты проверок, но проверки в состоянии `с ошибкой` или `ожидает выполнения` не блокируют слияние.

Настройка внешних проверок статуса

Чтобы настроить внешние проверки статуса, откройте проект и перейдите в «Настройки» → «Запросы на слияние».

На странице используются две связанные области настроек:

- «Проверки слияния» — настройки проекта, влияющие на возможность слияния.
- «Внешние проверки статуса» — таблица для создания, обновления и удаления внешних сервисов проверки статуса.

Пользователю нужно право на управление настройками запросов на слияние в проекте. В стандартных ролях оно доступно пользователям с ролью **Мейнтейнер** или **Владелец**.

Настройки проверок слияния

Проверки статуса должны быть успешными

Флажок «Проверки статуса должны быть успешными» блокирует запросы на слияние до тех пор, пока все применимые внешние проверки статуса не перейдут в состояние `пройдено`.

Если флажок выключен, внешние проверки статуса продолжают отображаться в запросах на слияние, но не блокируют слияние.

Тайм-аут проверок статуса

Поле «Тайм-аут проверок статуса» задает тайм-аут по умолчанию для проверок статуса в проекте.

Поведение:

- Значение по умолчанию — `5` минут.
- Значение должно быть `1` минута или больше.
- Значение применяется к проверкам без индивидуального тайм-аута.
- Если внешний сервис не ответил до истечения тайм-аута, ответ проверки переходит в состояние `с ошибкой`.

Таблица внешних проверок статуса

Таблица «Внешние проверки статуса» показывает все сервисы проверки статуса, настроенные в проекте.

В таблице отображаются:

- Название проверки статуса.
- URL внешнего сервиса.
- Область применения по целевым веткам.
- Тайм-аут проверки или тайм-аут проекта по умолчанию.
- Состояние общего секрета HMAC.
- Действия «Обновить» и «Удалить».

Если у проверки не задан индивидуальный тайм-аут, в таблице отображается тайм-аут проекта с пометкой `(по умолчанию)`.

Добавление внешней проверки статуса

Чтобы добавить внешнюю проверку статуса:

1. Откройте проект.
2. Перейдите в «Настройки» → «Запросы на слияние».
3. В блоке «Внешние проверки статуса» нажмите «Добавить внешнюю проверку статуса».
4. Заполните поля.
5. Нажмите «Новая внешняя проверка статуса».

Поле	Описание
Название сервиса	Название внешнего сервиса. Значение обязательно, должно быть уникальным в проекте и не должно превышать 255 символов
API для проверки	URL эндпоинта внешнего сервиса. Значение обязательно, должно быть уникальным в проекте и должно использовать протокол `http` или `https`
Целевая ветка	Область, определяющая целевые ветки запросов на слияние, для которых применяется проверка
Тайм-аут в минутах	Индивидуальный тайм-аут проверки. Если задан, переопределяет значение проекта **Тайм-аут проверок статуса**
Общий секрет HMAC	Необязательный секрет для подписи запросов, отправляемых из Deckhouse Code во внешний сервис

После создания проверки Deckhouse Code создает ответы в состоянии `ожидает выполнения` для подходящих открытых запросов на слияние. Запросы во внешний сервис для таких уже существующих запросов на слияние не отправляются. Эти ответы перейдут в состояние `с ошибкой` после истечения тайм-аута, если пользователь не перезапустит проверку.

Для новых событий запросов на слияние Deckhouse Code автоматически отправляет запросы в подходящие внешние сервисы проверки статуса.

Область применения по веткам

Поле «Целевая ветка» определяет, для каких запросов на слияние используется проверка статуса.

Значение поля «Целевая ветка»	Описание
Все ветки	Проверка применяется к запросам на слияние в любую целевую ветку
Все защищённые ветки	Проверка применяется к запросам на слияние в защищённые ветки
Выбранные защищённые ветки	Проверка применяется только к запросам на слияние в выбранные защищённые ветки. Поддерживаются защищённые ветки с wildcard-шаблонами

Если целевая ветка запроса на слияние не соответствует области применения проверки, проверка не отображается в виджете запроса на слияние.

Когда целевая ветка меняется, Deckhouse Code пересчитывает применимые проверки. Проверки, которые больше не применяются, удаляются из запроса на слияние, а новые применимые проверки добавляются в состоянии `ожидает выполнения`.

Общий секрет HMAC

Если задан **Общий секрет HMAC**, Deckhouse Code добавляет заголовок `X-Gitlab-Signature` к запросам во внешний сервис.

Значение заголовка — HMAC-SHA256 от тела запроса, сформированный с использованием общего секрета.

Секрет хранится в зашифрованном виде. После сохранения интерфейс показывает только факт наличия секрета. Чтобы заменить секрет:

1. В строке проверки нажмите «Обновить».
2. Нажмите «Изменить секрет».
3. Введите новое значение.
4. Нажмите «Обновить проверку статуса».

Жизненный цикл проверки

Когда происходит событие запроса на слияние, Deckhouse Code отправляет данные запроса на слияние во все применимые внешние сервисы проверки статуса. Данные содержат объект `external_approval_rule` с `id`, `name` и `external_url` проверки.

Ответ проверки может иметь один из следующих статусов:

Статус	Описание
`ожидает выполнения`	Deckhouse Code ожидает ответ внешнего сервиса
`пройдено`	Внешний сервис подтвердил состояние запроса на слияние
`с ошибкой`	Внешний сервис отклонил состояние запроса на слияние, запрос к внешнему URL завершился ошибкой или истек тайм-аут

Внешний сервис обновляет ответ через API-эндпоинт:

```
POST /projects/:id/merge_requests/:merge_request_iid/status_check_responses
```

Запрос должен содержать:

- `external_status_check_id` — идентификатор проверки статуса.
- `sha` — текущий `HEAD` SHA исходной ветки запроса на слияние.
- `status` — `пройдено` или `с ошибкой`.

Deckhouse Code не обновляет ответ, если:

- `sha` не совпадает с текущим `HEAD` исходной ветки.
- Ответ больше не находится в состоянии `ожидает выполнения`.
- Тайм-аут уже истек.

Тайм-ауты

Отсчёт тайм-аута начинается в момент отправки запроса во внешний сервис. Если пользователь перезапускает проверку в состоянии `с ошибкой`, отсчёт тайм-аута начинается с момента перезапуска.

Значение тайм-аута выбирается в следующем порядке:

1. Значение «Тайм-аут в минутах» у проверки статуса.
2. Значение проекта «Тайм-аут проверок статуса».

Фоновый обработчик проверяет ответы в состоянии `ожидает выполнения` каждую минуту. Когда ответ превышает свой тайм-аут, Deckhouse Code переводит его в состояние `с ошибкой` и записывает причину `Automatically closed after timeout`.

Если запрос во внешний сервис завершился ошибкой, Deckhouse Code переводит ответ в состояние `с ошибкой` и сохраняет причину ошибки. Пользователи могут увидеть причину ошибки в виджете запроса на слияние.

Виджет внешних проверок статуса

Виджет «Внешние проверки статуса» отображается на странице запроса на слияние, если для запроса есть применимые внешние проверки.

Виджет показывает:

- Название проверки.
- Статус проверки.
- URL внешнего сервиса, если ваша роль позволяет его видеть.
- Подробности ошибки для проверок в состоянии `с ошибкой`, если Deckhouse Code сохранил причину ошибки.
- Действие «Повторить» для проверок в состоянии `с ошибкой`, если ваша роль позволяет их перезапускать.

Виджет помогает авторам и ревьюерам понять, какие внешние системы ещё должны ответить до слияния запроса.

Проверки в состоянии «ожидает выполнения»

Проверка остаётся в состоянии `ожидает выполнения`, пока Deckhouse Code ожидает ответ внешнего сервиса.

Пока хотя бы одна проверка находится в состоянии `ожидает выполнения`, виджет запроса на слияние обновляет внешние проверки статуса примерно каждые 10 секунд. Опрос останавливается, когда проверок в состоянии `ожидает выполнения` больше нет.

Проверка в состоянии `ожидает выполнения` может автоматически перейти в состояние `с ошибкой`, если внешний сервис не ответит до истечения настроенного тайм-аута. Тайм-аут настраивается на уровне проекта или для конкретной внешней проверки статуса.

Проверки в состоянии «с ошибкой»

Проверка может перейти в состояние `с ошибкой`, если:

- Внешний сервис вернул `с ошибкой` для текущего `HEAD` SHA запроса на слияние.
- Deckhouse Code не смог отправить запрос во внешний сервис.
- URL внешнего сервиса заблокирован или недоступен.
- Внешний сервис не ответил до истечения тайм-аута.

Если в проекте включена настройка **Проверки статуса должны быть успешными**, проверка в состоянии `с ошибкой` блокирует слияние до тех пор, пока проверка не перейдет в `пройдено` или пока настройки проекта не будут изменены.

Перезапуск проверки в состоянии «с ошибкой»

Вы можете перезапустить проверку в состоянии `с ошибкой`, если у вас есть роль **Разработчик** или выше в проекте.

Перезапуск доступен только для проверок в состоянии `с ошибкой`, которые относятся к текущему `HEAD` SHA исходной ветки запроса на слияние.

Чтобы перезапустить проверку в состоянии `с ошибкой`:

1. Откройте запрос на слияние.
2. Найдите виджет «Внешние проверки статуса».
3. В строке проверки в состоянии `с ошибкой` нажмите «Повторить».

После перезапуска Deckhouse Code переводит проверку обратно в состояние `ожидает выполнения` и повторно отправляет текущие данные запроса на слияние во внешний сервис.

Используйте перезапуск после исправления проблемы во внешнем сервисе или если сбой был временным.

URL проверок

Виджет может показывать URL внешнего сервиса для проверки. URL виден только пользователям, роль которых позволяет просматривать URL внешних проверок статуса. В стандартных ролях проекта это доступно пользователям с ролью **Разработчик** или выше.

Если вы не видите URL, вы все равно можете видеть статус проверки, если ваша роль позволяет смотреть ответы проверок статуса.

Доступ к результатам проверок

Доступ к информации о внешних проверках статуса зависит от роли в проекте:

Действие	Минимальная роль
Создать, обновить, удалить или посмотреть сервисы проверки статуса проекта	**Мейнтейнер** или **Владелец**
Отправить обратный вызов со статусом проверки	**Разработчик**
Смотреть ответы проверок в виджете	**Наблюдатель**
Смотреть URL внешнего сервиса	**Разработчик**
Перезапустить проверку в состоянии `с ошибкой`	**Разработчик**

Для проектов с видимостью `внутренний` авторизованный пользователь без признака внешнего пользователя может читать ответы внешних проверок статуса в виджете на странице запроса на слияние, если у него есть доступ на чтение данного запроса.

URL внешней проверки при этом отображается только пользователям с правом просмотра URL внешних проверок (в стандартных ролях — `Разработчик` или выше), поэтому не отображается обычному пользователю, который не является участником проекта.

Удаление внешней проверки статуса

Чтобы удалить проверку:

1. Откройте проект.
2. Перейдите в «Настройки» → «Запросы на слияние».
3. В блоке «Внешние проверки статуса» нажмите «Удалить» для нужной проверки.
4. Подтвердите удаление.

После удаления проверка больше не применяется к запросам на слияние проекта.

События аудита

Deckhouse Code записывает события аудита для управления проверками статуса и изменения ответов.

Событие аудита	Описание
`external_status_check_created`	Создана внешняя проверка статуса
`external_status_check_updated`	Обновлена внешняя проверка статуса
`external_status_check_destroyed`	Удалена внешняя проверка статуса
`external_status_check_response_updated`	Ответ изменен через обратный вызов внешнего сервиса, перезапуск, ошибку запроса или тайм-аут

Устранение неполадок

Запрос на слияние заблокирован внешней проверкой статуса

Проверьте:

- Включен ли флажок «Проверки статуса должны быть успешными».
- Применяется ли проверка к целевой ветке запроса на слияние.
- Отправил ли внешний сервис `пройдено` для текущего `HEAD` SHA.
- Не истек ли тайм-аут до обратного вызова внешнего сервиса.

Проверка завершилась в статусе «с ошибкой» по тайм-ауту

Проверьте:

- Значение проекта «Тайм-аут проверок статуса».
- Индивидуальное значение «Тайм-аут в минутах» у проверки.

- Время ответа внешнего сервиса.
- Нужно ли перезапустить проверку после исправления внешнего сервиса.

Проверка остаётся в состоянии «ожидает выполнения»

Проверка может оставаться в состоянии `ожидает выполнения`, пока Deckhouse Code ожидает внешний сервис.

Проверьте:

- Доступен ли внешний сервис.
- Может ли сервис обработать данные запроса на слияние.
- Отправляет ли сервис ответ для текущего `HEAD` SHA.
- Достаточен ли настроенный тайм-аут для завершения обработки.

Если тайм-аут истечет, проверка перейдет в состояние `с ошибкой`.

Запрос во внешний сервис завершился ошибкой

Проверьте:

- URL в поле «API для проверки».
- Сетевой доступ от Deckhouse Code до внешнего сервиса.
- Использует ли URL протокол `http` или `https`.
- Текст ошибки в виджете запроса на слияние.

Действие Повторить недоступно

Действие «Повторить» отображается только если выполняются все условия:

- Проверка находится в состоянии `с ошибкой`.
- Проверка относится к текущему `HEAD` SHA запроса на слияние.
- У вас есть роль **Разработчик** или выше в проекте.

Если действие «Повторить» недоступно, попросите мейнтейнера проекта проверить настройки проекта и вашу роль.

URL внешнего сервиса не отображается

URL скрыт, если ваша роль не позволяет просматривать URL внешних проверок статуса. Если вам нужен доступ к URL внешнего сервиса, обратитесь к мейнтейнеру проекта.

Ошибка дублирования имени или URL

В проекте не может быть двух внешних проверок статуса с одинаковым «Название сервиса» или «API для проверки». Используйте уникальные значения для каждой проверки.

Цепочки слияния

Цепочка слияния (merge train) формирует очередь из запросов на слияние в целевую ветку. Перед слиянием каждый запрос проверяется с учётом запросов, стоящих перед ним в очереди. Благодаря этому в целевую ветку попадают только изменения, проверенные с учётом порядка их слияния.

Для проверки используются пайплайны для результатов слияния (merged results pipelines). Для первого запроса в очереди такой пайплайн проверяет его изменения вместе с текущим состоянием целевой ветки. Для каждого следующего запроса также учитываются изменения из предыдущих запросов в очереди. Поэтому если запросы проходят проверки по отдельности, но их изменения несовместимы, проблема будет обнаружена до слияния в целевую ветку.

Цепочки слияния настраиваются отдельно для каждого проекта. Для каждой целевой ветки проекта используется отдельная цепочка слияния.

Принцип работы цепочки слияния

Запросы на слияние располагаются в очереди в порядке добавления. Добавленные первыми запросы находятся в начале очереди.

Обработка запросов в очереди выполняется следующим образом:

1. Запрос на слияние, для которого включено автослияние (auto-merge), добавляется в конец очереди.
2. Deckhouse Code создаёт для запроса временную Git-ссылку (Git-ref) и запускает для него пайплайн. Git-ref содержит состояние целевой ветки, изменения из предыдущих запросов в очереди и изменения из текущего запроса.
3. Если пайплайн первого запроса в очереди завершается успешно, запрос сливается в целевую ветку и покидает очередь.
4. Первым в очереди становится следующий запрос. Если после слияния предыдущего запроса состояние, для которого запускался пайплайн, стало неактуальным, Deckhouse Code пересоздаёт пайплайн для нового состояния. Пайплайны последующих запросов также пересоздаются.

Цепочка не хранится как отдельный объект. Она формируется из запросов на слияние, добавленных в очередь для одной целевой ветки проекта. Когда запрос сливается или удаляется из очереди, позиции остальных запросов пересчитываются.

Предварительные требования

Перед настройкой цепочек слияния убедитесь, что выполнены следующие требования:

- в проекте включён механизм CI/CD;
- конфигурационный файл CI/CD настроен на создание пайплайнов для запросов на слияние;
- в проекте включены пайплайны для результатов слияния;
- для изменения настроек проекта у пользователя есть роль **Мейнтейнер** или **Владелец**.

Включение цепочек слияния

Чтобы включить цепочки слияния:

1. Откройте проект.
2. Перейдите в «Настройки» → «Запросы на слияние».
3. В разделе «Параметры слияния» установите флажок «Включить пайплайны с результатами слияния».
4. Установите флажок «Включить цепочки слияния».
5. Нажмите «Сохранить изменения».

После включения цепочек слияния становятся доступны дополнительные настройки:

Настройка	Описание
Сливать немедленно без перезапуска цепочки слияния	Добавляет варианты немедленного слияния, описанные в разделе Немедленное слияние. Настройка недоступна, если проект требует слияния в режиме fast-forward или semi-linear, а также пока установлено требование слияния через цепочку
Требовать слияния запросов только через цепочку слияния	Отклоняет все прямые слияния в проекте. Подробнее — в разделе Обязательное слияние через цепочку
Максимальное количество параллельных пайплайнов на цепочку слияния	Ограничивает количество одновременно выполняемых пайплайнов цепочки. Подробнее — в разделе Ограничение параллельных пайплайнов

Снятие флажка «Включить цепочки слияния» не очищает существующую очередь сразу. Очередь освобождается по мере того, как цикл обновления доходит до каждого запроса, поэтому некоторое время после сохранения настройки они могут по-прежнему отображаться в очереди.

Работа с очередью

Запросы на слияние можно добавлять в очередь, просматривать, удалять из неё или сливать немедленно.

Добавление запроса на слияние в цепочку

Добавить запрос на слияние в цепочку можно через веб-интерфейс или API.

Добавление через веб-интерфейс

Чтобы добавить запрос на слияние в цепочку, нажмите «Установить автоматическое слияние» на странице запроса. Если в проекте включены цепочки слияния, запрос будет добавлен в цепочку вместо прямого слияния.

В зависимости от состояния запроса доступен один из следующих вариантов:

Вариант	Описание
Добавить в цепочку слияния	Запрос сразу добавляется в очередь. Вариант доступен, если запрос готов к слиянию и пайплайн для текущего `HEAD` SHA завершён
Добавить в цепочку слияния после прохождения всех проверок слияния	Запрос добавляется в очередь после успешного прохождения проверок

Вариант, который будет использован, отображается рядом с кнопкой слияния. Если запрос уже проверялся пайплайном цепочки слияния, отображается текст «Повторно добавить в цепочку слияния».

Если последний пайплайн запроса завершился с ошибкой, перед добавлением запроса в очередь Deckhouse Code запрашивает подтверждение.

Добавление через API

Чтобы добавить запрос на слияние в цепочку через API, выполните запрос:

```
curl --request POST --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/merge_trains/merge_requests/
<MERGE_REQUEST_IID>"
```

Где:

- `<TOKEN>` — токен для аутентификации в Deckhouse Code;
- `<HOST>` — адрес инстанса Deckhouse Code;
- `<PROJECT_ID>` — идентификатор проекта;
- `<MERGE_REQUEST_IID>` — внутренний идентификатор запроса на слияние в проекте.

В запросе можно указать следующие параметры:

Параметр	Описание
`sha`	SHA, который вы ожидаете увидеть в исходной ветке. Если текущий SHA не совпадает с указанным, запрос отклоняется
`squash`	Объединяет коммиты исходной ветки в один при слиянии запроса. При слиянии через цепочку также учитывается настройка squash самого запроса на слияние
`auto_merge`	Добавляет запрос в очередь только после успешного прохождения проверок слияния

Возможные коды ответа:

Код	Описание
`201`	Запрос на слияние добавлен в очередь
`202`	Запрос ожидает прохождения проверок слияния и ещё не добавлен в очередь
`400`	Цепочки слияния отключены в проекте или текущее состояние запроса не позволяет добавить его в очередь
`401`	Запрос выполнен без аутентификации
`403`	Роль пользователя не позволяет просматривать цепочку или сливать этот запрос
`404`	Проект или запрос на слияние не найден
`409`	Для запроса на слияние уже включено автослияние

Просмотр цепочки слияния

Получить информацию о цепочке можно через веб-интерфейс или API.

Просмотр через веб-интерфейс

Открыть страницу цепочек слияния проекта можно одним из следующих способов:

- на странице запроса на слияние, добавленного в очередь, перейдите по ссылке «Просмотреть цепочку слияния»;
- на странице пайплайна цепочки слияния перейдите по ссылке «Просмотр деталей цепочки слияния» в заголовке страницы;
- откройте страницу напрямую по адресу `https://<HOST>/<NAMESPACE>/<PROJECT>/-/merge_trains`.

На странице цепочек слияния отображаются:

- фильтр для выбора целевой ветки. Изначально выбрана ветка проекта по умолчанию.
- вкладка «Активные» с запросами на слияние, которые находятся в очереди, и вкладка «Слитые» с уже слитыми запросами. На каждой вкладке указано количество запросов;
- информация о каждом запросе на слияние: статус пайплайна, заголовок, время добавления в очередь или слияния, а также пользователь, который добавил или слил запрос.

Пока запрос на слияние находится в очереди, его позиция отображается на странице запроса:

- для первого запроса — «Запущена новая цепочка слияния, и этот запрос на слияние стоит первым в очереди»;
- в остальных случаях отображается позиция в очереди, например, «Этот запрос на слияние занимает позицию 2 из 5 в очереди».

Просмотр через API

Чтобы получить информацию о цепочках слияния через API, выполните один из следующих запросов:

```
# Все цепочки проекта.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/merge_trains"

# Цепочка одной целевой ветки.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/merge_trains/<TARGET_BRANCH>"

# Статус одного запроса на слияние в очереди.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/merge_trains/merge_requests/
<MERGE_REQUEST_IID>"
```

Где:

- `<TOKEN>` — токен для аутентификации в Deckhouse Code;
- `<HOST>` — адрес инстанса Deckhouse Code;
- `<PROJECT_ID>` — идентификатор проекта;
- `<TARGET_BRANCH>` — имя целевой ветки;
- `<MERGE_REQUEST_IID>` — внутренний идентификатор запроса на слияние в проекте.

Первые два эндпоинта принимают следующие параметры:

Параметр	Описание
`scope`	Ограничивает ответ запросами, которые ещё стоят в очереди (`active`), или уже покинули её (`complete`)
`sort`	Задаёт порядок запросов по их позиции в очереди: `asc` — от самых старых к новым, `desc` — от самых новых к старым. Значение по умолчанию — `desc`

Оба эндпоинта поддерживают постраничный вывод с помощью параметров `page` и `per_page`. Первый эндпоинт возвращает запросы на слияние для всех целевых веток. Чтобы получить очередь только для определённой ветки, используйте второй эндпоинт и укажите её имя.

Получить информацию о цепочках слияния и очередях проекта также можно через GraphQL API.

Удаление запроса на слияние из цепочки

Чтобы удалить запрос на слияние из цепочки, используйте один из следующих способов:

- отмените автослияние на странице запроса;
- на странице цепочек слияния нажмите значок удаления в строке запроса, затем нажмите «Удалить из цепочки слияния».

При удалении запроса из очереди пайплайны для следующих за ним запросов пересоздаются, поскольку состояние цепочки изменилось.

Запрос нельзя удалить из цепочки, если его слияние уже началось. В этом случае Deckhouse Code отклоняет удаление и завершает слияние запроса.

Deckhouse Code также автоматически удаляет запрос из цепочки, если он больше не может быть слит. Подробнее — в разделе Запрос на слияние удалён из цепочки.

Немедленное слияние

Запрос на слияние можно слить немедленно, не дожидаясь его очереди в цепочке. Немедленное слияние доступно через веб-интерфейс и API.

Слияние через веб-интерфейс

Если в проекте включена настройка «Сливать немедленно без перезапуска цепочки слияния», на странице запроса в очереди рядом с кнопкой слияния доступны два дополнительных варианта:

Вариант	Описание
Слить сейчас и перезапустить цепочку	Запрос сливается немедленно, а пайплайны следующих за ним запросов пересоздаются с учётом слитых изменений
Слить сейчас без перезапуска цепочки	Запрос сливается немедленно, а существующие пайплайны продолжают выполняться без перезапуска. В результате запросы, уже находящиеся в очереди, не проверяются вместе со слитыми изменениями

⚠ Используйте вариант «Слить сейчас без перезапуска цепочки» только если слитые изменения не могут повлиять на запросы, находящиеся в очереди.

Оба варианта требуют подтверждения и позволяют выполнить слияние вне установленного порядка очереди.

Варианты немедленного слияния недоступны, если в проекте включено обязательное слияние через цепочку. Настройка «Сливать немедленно без перезапуска цепочки слияния» также недоступна, если для проекта требуется слияние в режиме fast-forward или semi-linear.

Результат немедленного слияния зависит от выбранного варианта:

- при слиянии без перезапуска цепочки запрос отображается на вкладке «Слитые» со статусом `skip_merged`;
- при слиянии с перезапуском цепочки запрос сливается напрямую и не отображается на вкладке «Слитые». После слияния он удаляется из очереди, а удаление записывается в историю активности запроса.

Слияние через API

Для немедленного слияния через API используйте эндпоинт слияния с параметром `skip_merge_train`:

```
curl --request PUT --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/merge_requests/<MERGE_REQUEST_IID>/
merge?skip_merge_train=true"
```

Где:

- `<TOKEN>` — токен для аутентификации в Deckhouse Code;
- `<HOST>` — адрес инстанса Deckhouse Code;
- `<PROJECT_ID>` — идентификатор проекта;
- `<MERGE_REQUEST_IID>` — внутренний идентификатор запроса на слияние в проекте.

Параметр `skip_merge_train` принимает следующие значения:

Значение	Описание
`true`	Запрос сливается немедленно без перезапуска цепочки
`false`	Значение по умолчанию. Запрос сливается немедленно, а пайплайны следующих за ним запросов пересоздаются

Параметр действует только при включённой настройке «Сливать немедленно без перезапуска цепочки слияния». В остальных случаях параметр игнорируется. Если в проекте включено обязательное слияние через цепочку, запрос на слияние отклоняется.

Настройка цепочки слияния

В настройках проекта можно сделать слияние через цепочку обязательным и ограничить количество одновременно выполняемых пайплайнов.

Обязательное слияние через цепочку

Чтобы запретить прямое слияние запросов в проекте, установите флажок «Требовать слияния запросов только через цепочку слияния».

После включения этой настройки:

- варианты немедленного слияния становятся недоступны;
- запрос на слияние через API отклоняется, если для него не включено автослияние и он не добавлен в цепочку;

- слияния, которые выполняет сама цепочка, продолжают работать без изменений.

При включении обязательного слияния настройка «Сливать немедленно без перезапуска цепочки слияния» автоматически отключается и становится недоступна. После сохранения настроек с включённым обязательным слиянием эта настройка остаётся отключённой, в том числе после отключения обязательного слияния. Чтобы снова использовать её, включите настройку заново.

Обязательное слияние действует только при включённых цепочках слияния. Если цепочки отключены, запросы можно сливать напрямую.

Ограничение распространяется на всех пользователей, включая владельцев проекта и администраторов.

Ограничение параллельных пайплайнов

Для нескольких запросов в цепочке пайплайны могут выполняться одновременно. Количество одновременно выполняемых пайплайнов определяется двумя ограничениями:

- ограничением инстанса Deckhouse Code, которое задаёт администратор. По умолчанию — `20`;
- настройкой проекта «Максимальное количество параллельных пайплайнов на цепочку слияния».

Используется меньшее из этих двух значений. Например, если для инстанса установлено ограничение `20`, а для проекта — `10`, одновременно могут выполняться не более 10 пайплайнов.

Чтобы использовать ограничение инстанса, оставьте настройку проекта пустой. Для проекта можно указать значение от `1` до `500`, но оно не может увеличить ограничение, установленное для инстанса.

Пайплайны цепочки слияния

Для каждого запроса в очереди Deckhouse Code запускает пайплайн для Git ref. Такой пайплайн выполняется не для исходной ветки запроса. В списке пайплайнов он помечается как `merge train`, что позволяет отличить его от обычного пайплайна для результатов слияния.

Завершённый пайплайн цепочки слияния нельзя перезапустить, поскольку состояние репозитория, для которого он выполнялся, могло измениться. Чтобы запустить новый пайплайн, повторно добавьте запрос на слияние в цепочку.

Пайплайн пересоздаётся в следующих случаях:

- запрос, находящийся перед ним в очереди, слит, удалён из очереди или получил новый пайплайн;
- пайплайн предыдущего запроса в очереди завершился с ошибкой;
- изменения отправлены напрямую в целевую ветку. В этом случае пересоздаются пайплайны для всех запросов в очереди, начиная с первого.

Запуск задач только в пайплайнах цепочки слияния

В пайплайне цепочки слияния переменная `CI_MERGE_REQUEST_EVENT_TYPE` имеет значение `merge_train`. С её помощью можно запускать задачу только в пайплайнах цепочки слияния:

```
integration-tests:
  script: ./run-integration-tests.sh
  rules:
    - if: $CI_MERGE_REQUEST_EVENT_TYPE == "merge_train"
```

Чтобы, наоборот, пропускать задачу в пайплайнах цепочки, используйте обратное условие:

```
quick-lint:
  script: ./lint.sh
  rules:
    - if: $CI_MERGE_REQUEST_EVENT_TYPE != "merge_train"
```

Статусы запросов в очереди

Для каждого запроса на слияние в цепочке определяется статус, который можно получить через REST API или GraphQL API. В веб-интерфейсе статусы напрямую не отображаются: запросы распределяются между вкладками «Активные» и «Слитые».

Статус	Описание
`idle`	Запрос находится в очереди, пайплайн ещё не запущен
`fresh`	Пайплайн соответствует текущему результату слияния
`stale`	Состояние цепочки изменилось, пайплайн пересоздаётся
`merging`	Выполняется слияние запроса
`merged`	Запрос слит через цепочку и удалён из очереди
`skip_merged`	Запрос слит немедленно без перезапуска цепочки и удалён из очереди

Запросы на слияние со статусами `idle`, `fresh` и `stale` отображаются на вкладке «Активные», их можно удалить из очереди. Запросы со статусами `merged` и `skip_merged` отображаются на вкладке «Слитые». Запрос со статусом `merging` не отображается ни на одной из вкладок до завершения слияния.

Активность запроса на слияние

Deckhouse Code записывает, что происходит с запросом на слияние в цепочке, в виде системных заметок в разделе «Активность» запроса. Фиксируются следующие события:

- запрос запустил новую цепочку или был добавлен в существующую цепочку на определённую позицию в очереди;
- запрос удалён из цепочки пользователем;
- запрос удалён из цепочки системой с указанием причины;
- автоматическое добавление в цепочку после прохождения проверок включено, отменено или прервано с указанием причины.

Если запрос на слияние удалён из цепочки системой, его участники также получают соответствующую задачу в списке задач.

Доступ к цепочкам слияния

Доступные действия с цепочками слияния зависят от роли пользователя и настроек проекта:

Действие	Требования
Просмотр страницы цепочек слияния и просмотр информации через API	В проекте включены цепочки слияния, и у пользователя есть доступ на чтение запросов на слияние и пайплайнов. В стандартных ролях это доступно пользователям с ролью **Наблюдатель** или выше
Получение статуса цепочки для отдельного запроса на слияние	Те же требования, что и для просмотра цепочки
Добавление запроса на слияние в цепочку	У пользователя есть право на слияние этого запроса. В стандартных ролях это доступно пользователям с ролью **Разработчик** или выше
Удаление запроса на слияние из очереди на странице цепочек	У пользователя есть право на отмену пайплайнов, изменение запроса на слияние и слияние в целевую ветку. В стандартных ролях это доступно пользователям с ролью **Разработчик** или выше
Изменение настроек цепочек слияния в проекте	Роль **Мейнтейнер** или **Владелец**

Для просмотра цепочек слияния через веб-интерфейс или API требуется аутентификация. Анонимным пользователям цепочки недоступны даже в публичных проектах.

События аудита

Deckhouse Code записывает следующие события аудита при изменении настроек цепочек слияния в проекте:

Событие аудита	Описание
`project_cicd_merge_trains_enabled_updated`	Изменена настройка включения цепочек слияния в проекте
`project_cicd_merge_train_enforced_updated`	Изменена настройка обязательного слияния через цепочку

Устранение проблем

Ниже описаны распространённые проблемы при работе с цепочками слияния и способы их устранения.

Запрос на слияние удалён из цепочки

Если во время выполнения пайплайна запрос больше не может быть слит, Deckhouse Code удаляет его из очереди, чтобы не блокировать остальные запросы. Причина удаления указывается в системной заметке в разделе «Активность».

Наиболее частые причины удаления:

- цепочки слияния отключены в проекте;
- в исходную ветку запроса на слияние добавлены новые коммиты;
- запрос на слияние закрыт;
- запрос на слияние помечен как черновик;
- изменена целевая ветка запроса на слияние;
- запрос на слияние стал непригодным для слияния, например, из-за конфликта;
- для запроса на слияние отменено автослияние;
- пайплайн цепочки слияния для этого запроса завершился с ошибкой;
- удалена учётная запись пользователя, добавившего запрос в очередь.

Если запрос ожидает прохождения проверок перед добавлением в цепочку, процесс также прерывается, если запрос становится черновиком или его пайплайн завершается с ошибкой.

Дискуссии и апрувы проверяются при добавлении запроса в очередь. Новая дискуссия, открытая после этого, не блокирует слияние, даже если в проекте требуется закрытие всех дискуссий. Это позволяет сохранить уже проверенное пайплайном состояние и не пересоздавать пайплайны последующих запросов.

Если обязательный апрув отозван, слияние блокируется. Запрос сохраняет позицию в очереди, но при достижении её начала удаляется из цепочки.

Устраните причину, указанную в системной заметке, и повторно добавьте запрос в цепочку.

Кнопка слияния не предлагает варианты цепочки слияния

Проверьте следующее:

- в настройках проекта включены пайплайны с результатами слияния и цепочки слияния;
- конфигурационный файл CI/CD создаёт пайплайны для запросов на слияние;
- у запроса на слияние есть пайплайн для текущего `HEAD` SHA исходной ветки, и этот пайплайн завершён;
- роль пользователя позволяет слить этот запрос.

Слияние отклонено в проекте, требующем цепочку слияния

Если включена настройка «Требовать слияния запросов только через цепочку слияния», прямое слияние отклоняется, в том числе через API. Вместо этого добавьте запрос на слияние в цепочку.

Ограничение действует для всех пользователей, включая владельца проекта. Чтобы снова разрешить прямые слияния, отключите эту настройку.

Пайплайн цепочки слияния нельзя перезапустить

Завершённый пайплайн цепочки слияния нельзя перезапустить, поскольку состояние, для которого он выполнялся, могло измениться. Чтобы создать пайплайн для текущего состояния, повторно добавьте запрос на слияние в цепочку.

Слияние выглядит зависшим

Если процесс слияния прерывается, Deckhouse Code автоматически возвращает запрос в очередь и продолжает работу цепочки. Дополнительные действия не требуются.

Цепочки слияния отключены, но очередь всё ещё отображается

После отключения цепочек запросы удаляются из существующей очереди не сразу. Некоторое время они могут по-прежнему отображаться на странице, пока Deckhouse Code постепенно очищает очередь.

Страница цепочек слияния пуста

Проверьте следующее:

- в фильтре выбрана целевая ветка, для которой нужно просмотреть очередь;
- для запросов на слияние включено автослияние;
- в проекте включены цепочки слияния. Если они отключены, страница недоступна, даже если очередь осталась с прежнего состояния.

Действие удаления недоступно

Удалить запрос из цепочки можно только на вкладке «Активные» и при наличии прав на отмену пайплайнов, изменение запроса на слияние и слияние в целевую ветку.

Запрос нельзя удалить из цепочки, если его слияние уже началось.

Аналитика ревью кода

Отчёт показывает, сколько времени открытые запросы на слияние проекта ждут ревью.

Что попадает в отчёт

В отчёт попадают открытые запросы на слияние проекта, у которых есть хотя бы один комментарий от ревьювера запроса (любого пользователя, кроме автора).

Слитые, закрытые и запросы на слияние без комментариев в отчёте отсутствуют.

Комментарии ботов и системные заметки (смена метки, назначение проверяющего) не считаются началом ревью, поэтому запрос на слияние, у которого есть только такие комментарии, в отчёт не попадает.

Просмотр отчёта

Откройте проект и выберите в боковом меню «Аналитика» → «Аналитика ревью кода».

Подсчёт времени, затраченного на ревью

Время ревью отсчитывается от первого комментария ревьювера до текущего момента.

Значение приблизительное и отображается в минутах, часах или днях. Время начала ревью фиксируется в момент сохранения комментария. Запрос, который ждал ревью восемь месяцев, имел системные комментарии и получил комментарий ревьювера час назад, покажет время ревью «примерно час».

Фильтры

Панель фильтров над таблицей позволяет настроить фильтрацию отчёта по этапу и по метке. Фильтр «Этап» принимает одно значение, фильтр «Метка» принимает несколько значений.

Перечень запросов на слияние

Таблица «Запросы на слияние на ревью» содержит запросы, которые находятся на ревью, начиная с самого раннего по дате создания. Счётчик рядом с заголовком таблицы показывает, сколько запросов на слияние остаётся на ревью с учётом фильтров.

Одна страница таблицы содержит не более 20 запросов на слиянии, при превышении общего количества под таблицей отображаются элементы постраничной навигации. Смена фильтра возвращает таблицу на первую страницу.

Таблица содержит следующие колонки:

Колонка	Содержимое
«Запрос на слияние»	Заголовок со ссылкой на запрос на слияние, его номер, количество меток и комментариев, а также количество одобрений, если они есть
«Время ревью»	Время, прошедшее с первого комментария ревьювера
«Автор»	Автор запроса на слияние
«Апруверы»	Пользователи, одобрившие запрос на слияние
«Комментарии»	Количество комментариев
«Коммиты»	Количество коммитов
«Изменения строк»	Количество добавленных и удалённых строк

Если с учётом фильтров на ревью нет ни одного запроса на слияние, вместо таблицы отображается сообщение «Нет запросов на слияние на ревью».

Доступ к отчёту

Отчёт доступен пользователям с ролью «Наблюдатель» или выше. В публичном проекте он доступен всем, кто видит проект, включая анонимных пользователей.

Если в настройках проекта отключена функция «Аналитика», отчёт и пункт меню недоступны.

Аналитика запросов на слияние

Отчёт показывает, сколько запросов на слияние было слито за выбранный период и сколько времени занимало слияние. Информация представлена в карточке «Среднее время до слияния», графике «Пропускная способность» и таблице «Запросы на слияние».

Что попадает в отчёт

В отчёт попадают слитые запросы на слияние проекта в пределах выбранного диапазона дат. Открытые запросы и запросы, закрытые без слияния, в отчёте отсутствуют.

Запрос, созданный до начала диапазона и слитый внутри него, учитывается полностью: в среднее время до слияния попадает весь интервал от создания до слияния.

Просмотр отчёта

Откройте проект и выберите в боковом меню «Аналитика» → «Аналитика запросов на слияние».

Диапазоны дат

Для просмотра можно выбрать один из диапазонов дат, начиная с текущего момента:

- «Последние 7 дней»;
- «Последние 30 дней»;
- «Последние 90 дней»;
- «Последние 180 дней»;
- «Последние 365 дней» (вариант по умолчанию);

В варианте «Произвольный диапазон» даты задаются в полях «От» и «До». Максимальная ширина диапазона ограничена 365 днями.

Выбранный диапазон отображается рядом с заголовками «Среднее время до слияния», «Пропускная способность» и «Запросы на слияние».

График пропускной способности

График «Пропускная способность» показывает количество слитых запросов на слияние по месяцам. В первом и последнем месяце диапазона учитываются только дни, которые входят в диапазон: например, у диапазона, начинающегося 15 марта, первая точка графика охватывает 15–31 марта.

Если в диапазоне с учётом фильтров нет слитых запросов на слияние, вместо графика отображается сообщение «В этом диапазоне дат не слито ни одного запроса на слияние».

Среднее время до слияния

Карточка «Среднее время до слияния» показывает среднее время от создания запроса на слияние до его слияния за весь диапазон в днях.

В среднее значение попадают запросы, у которых момент слияния позже момента создания. Запрос, у которого момент слияния совпадает с моментом создания или предшествует ему, например импортированный с уже заполненным `merged_at`, учитывается на графике и в таблице. В расчёт среднего он не входит. Если в диапазоне нет ни одного запроса с моментом слияния позже создания, в карточке отображается прочерк (`-`).

Фильтры

Панель фильтров позволяет настроить фильтрацию для всех элементов отчёта одновременно. Доступны следующие фильтры:

- «Исходная ветка»;
- «Целевая ветка»;
- «Этап»;
- «Метка»;
- «Автор»;
- «Ответственный».

Фильтр «Метка» принимает несколько значений, и в отчёте остаются запросы на слияние, у которых есть все выбранные метки.

Таблица запросов на слияние

Таблица «Запросы на слияние» содержит слитые запросы на слияние, входящие в диапазон дат, начиная с последнего слитого. Одна страница таблицы содержит не более 20 запросов на слияние, при превышении общего количества под таблицей отображаются элементы постраничной навигации. Смена фильтра возвращает таблицу на первую страницу.

Таблица содержит следующие колонки:

Колонка	Содержимое
«Запрос на слияние»	Заголовок со ссылкой на запрос на слияние, его номер, статус пайплайна, количество меток и комментариев, а также количество одобрений, если они есть
«Дата слияния»	Дата, когда запрос на слияние был слит
«Время до слияния»	Приблизительный интервал между созданием запроса на слияние и его слиянием, в минутах, часах или днях
«Этап»	Этап запроса на слияние со ссылкой на него
«Коммиты»	Количество коммитов в запросе на слияние
«Пайплайны»	Количество пайплайнов запроса на слияние
«Изменения строк»	Количество добавленных и удалённых строк
«Ответственные»	Аватары ответственных

Если в диапазоне с учётом фильтров нет слитых запросов на слияние, вместо таблицы отображается сообщение «В этом диапазоне дат не слито ни одного запроса на слияние» и предложение расширить диапазон или очистить фильтры.

Доступ к отчёту

Отчёт доступен пользователям с ролью «Наблюдатель» или выше. Пользователям с ролью «Гость» и анонимным пользователям отчёт недоступен даже в публичном проекте.

Если в настройках проекта отключена функция «Аналитика», отчёт и пункт меню недоступны.

Аналитика продуктивности

Отчёт «Аналитика продуктивности» показывает, сколько времени слитые запросы на слияние группы шли до слияния, на какие этапы их жизни ушло это время и насколько велики изменения. Страница содержит гистограмму запросов на слияние по выбранной метрике, точечный график тех же запросов по дням слияния со скользящей медианой и таблицу слитых запросов на слияние периода. Отчёт доступен для группы.

Доступность отчёта

Отчёт доступен участникам группы с ролью «Наблюдатель» и выше, в том числе тем, кто получил роль в родительской группе, и пользователям с типом «Аудитор». Участники с ролью «Гость» или «Планировщик» и посетители публичной группы не видят пункт меню, а адрес отвечает им `404`.

В отчёт входят только проекты, запросы на слияние которых читатель может видеть: проект, к которому у читателя нет доступа, или проект, где запросы на слияние скрыты от него, в расчёт не попадает.

Просмотр отчёта

Чтобы просмотреть отчёт:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика продуктивности».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/productivity_analytics`

Что попадает в отчёт

Отчёт учитывает слитые запросы на слияние проектов группы и проектов её подгрупп, включая архивные проекты, дата слияния которых входит в период отчёта. Открытые запросы на слияние и запросы, закрытые без слияния, в отчёте отсутствуют.

Дата слияния берётся по UTC, и дни на графиках — это дни по UTC.

Метрики времени отсчитываются от открытия запроса на слияние: запрос, открытый до периода и слитый внутри него, учитывается полностью. Запрос, у которого момент слияния совпадает с моментом открытия или предшествует ему, в отчёт не попадает.

Даты первого и последнего коммита, число коммитов, число изменённых строк и число изменённых файлов записываются из изменений запроса на слияние при его открытии и после каждой отправки изменений в его исходную ветку. У запроса на слияние, изменения которого были отправлены до того, как инстанс начал записывать эти колонки, они пусты, пока администратор не

заполнит их так, как описано в разделе Заполнение колонок для существующих запросов на слияние.

Метрики

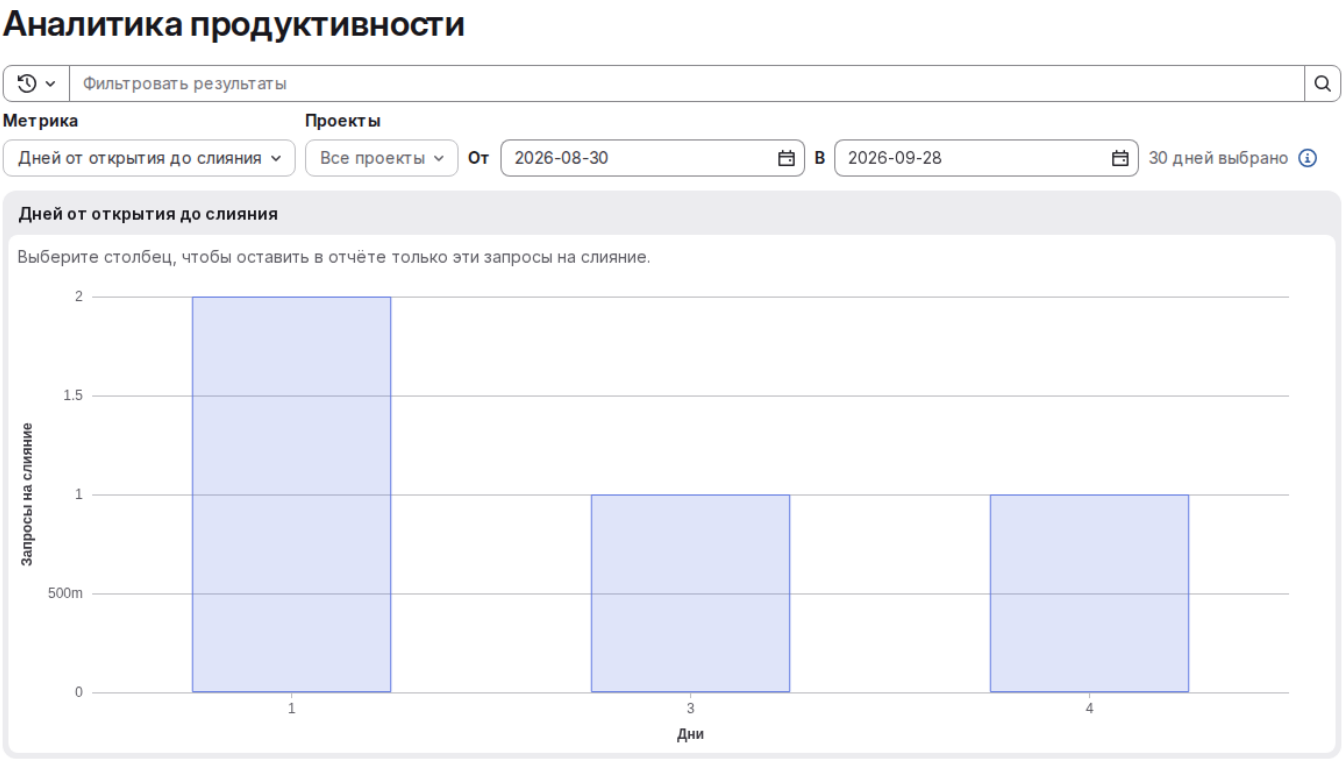
Список «Метрика» задаёт метрику гистограммы и вертикальной оси точечного графика. Список открывается на «Дней от открытия до слияния».

Метрика	Содержимое	Столбцы гистограммы
«Дней от открытия до слияния»	Полных дней от открытия запроса на слияние до его слияния	По столбцу на каждое число дней, в котором есть запросы на слияние; общего столбца для хвоста нет, поэтому запрос, слитый через год, — отдельный столбец на `365`
«От открытия до первого комментария»	Время от открытия запроса на слияние до первого комментария	`<1h`, `1-3h`, `3-6h`, `6-12h`, `12-24h`, `24-48h`, `48h+`
«От первого комментария до последнего коммита»	Время от первого комментария до последнего коммита	Те же интервалы в часах
«От последнего коммита до слияния»	Время от последнего коммита до слияния	Те же интервалы в часах
«Коммитов в запросе на слияние»	Число коммитов	От `0` до `19`, затем `20+`
«Изменённых строк на коммит»	Изменённые строки, делённые на число коммитов	`<10`, `10-25`, `25-50`, `50-100`, `100-250`, `250-500`, `500+`
«Изменённых файлов»	Число изменённых файлов	От `0` до `19`, затем `20+`

Запрос на слияние, о котором метрика ничего не говорит, в гистограмму и на точечный график не попадает: ни разу не прокомментированный — для «От открытия до первого комментария» и «От первого комментария до последнего коммита»; тот, у которого последний коммит предшествует первому комментарию, — для «От первого комментария до последнего коммита»; тот, у которого колонки коммитов пусты, — для всех метрик, читающих коммиты; без коммитов — для «Изменённых строк на коммит».

Гистограмма

Заголовок гистограммы — выбранная метрика. Каждый столбец содержит запросы на слияние периода, которые в него попали; горизонтальная ось названа единицей метрики («Дни», «Часы», «Коммиты», «Строки» или «Файлы»), вертикальная — «Запросы на слияние».

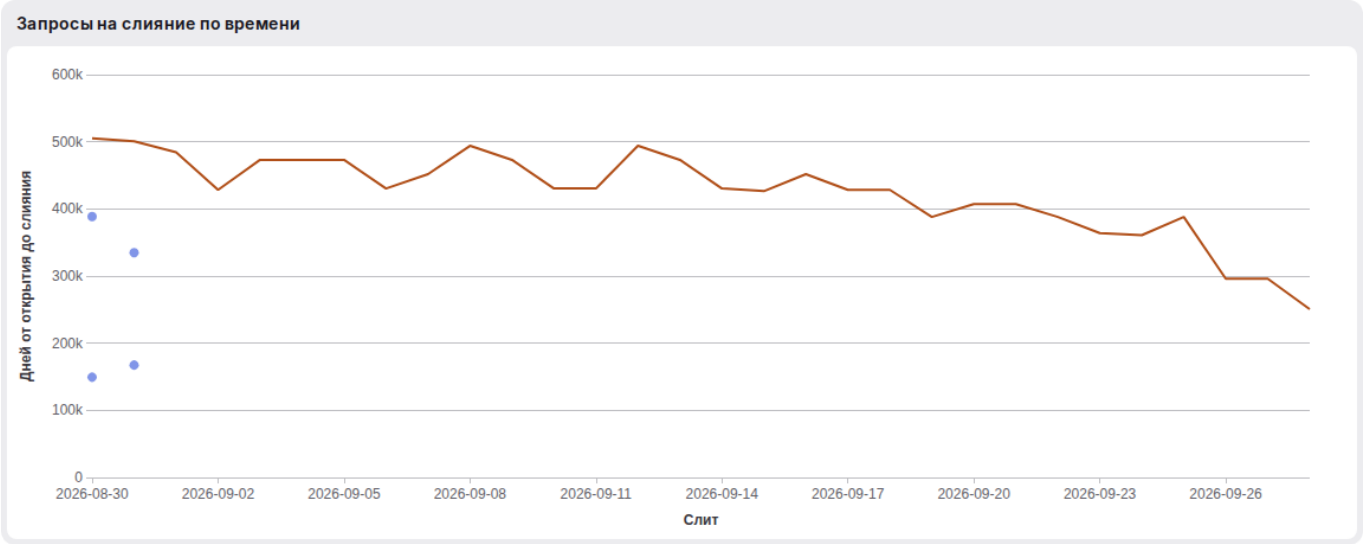


Выбор столбца ограничивает точечный график, его медиану и таблицу запросами на слияние этого столбца; остальные столбцы отображаются полупрозрачными. Можно выбрать несколько столбцов; повторный выбор столбца снимает его, а кнопка «Снять выбор столбцов» над графиком снимает все. Строка под графиком называет выбор: `Ниже — запросы на слияние из 2 выбранных столбцов.` Смена метрики снимает выбор.

Если в периоде с учётом фильтров нет слитых запросов на слияние, вместо графика отображается сообщение «За этот период не было слито ни одного запроса на слияние» и подсказка «Расширьте диапазон дат или выберите другие проекты, чтобы увидеть разброс времени слияния.»

Запросы на слияние по времени

График «Запросы на слияние по времени» показывает по точке на каждый слитый запрос на слияние: горизонтальная ось «Слит» содержит все дни периода, включая дни без слияний, а вертикальная — выбранную метрику.



Вертикальная ось метрики времени отсчитывает секунды, какая бы единица ни стояла в названии метрики: запрос, слитый через четыре дня, метрика «Дней от открытия до слияния» помещает около `350k`. Гистограмма и таблица той же метрики считают дни.

Линия «<метрика>, медиана за 30 дней» — медиана метрики за 30 дней до каждого дня периода включительно. Она учитывает дни до начала периода, поэтому идёт с его первого дня. Медиана считается по сетке значений: для метрики времени она отличается от точной менее чем на полпроцента, а длительность больше пяти лет считается как пять лет.

На графике не больше 2000 точек, начиная с последних слитых. Если запросов на слияние в периоде больше, строка под графиком принимает вид `Показаны последние 2000 из 3125 запросов на слияние.`; гистограмма и таблица учитывают все.

Период отчёта

Период открывается на последних 30 днях, включая текущий, и охватывает не больше 90 дней, считая оба конца. Период задаётся в полях «От» и «В»; дата окончания не позже текущего дня, а более широкий диапазон выбрать нельзя. Строка рядом с полями считает дни: `30 дней выбрано`.

Период в адресе страницы не сохраняется: ссылка открывается на последних 30 днях.

Фильтры

Список «Проекты» ограничивает отчёт выбранными проектами группы и её подгрупп. Он открывается на «Все проекты», принимает несколько проектов, содержит поиск, а «Очистить всё» возвращает его ко всем проектам. С выбранными проектами список принимает вид `Выбрано 3 проекта`. Выбранные проекты в адресе страницы не сохраняются.

Панель фильтров над графиками ограничивает гистограмму, точечный график и таблицу одновременно:

- «Автор» — один автор;

- «Метка» — одна или несколько меток; в отчёте остаются запросы на слияние со всеми выбранными метками;
- «Этап» — один этап; «Нет» оставляет запросы на слияние без этапа, «Любой» — запросы с любым этапом.

Фильтры принимают только оператор `=`.

Метрика, порядок сортировки, выбранные столбцы и фильтры сохраняются в адресе страницы, поэтому настроенный отчёт можно передать ссылкой:

```
https://<HOST>/groups/<GROUP>/-/analytics/productivity_analytics?
metric=TIME_TO_MERGE&sort=DAYS_TO_MERGE_DESC&buckets[]=2
```

Ссылка открывается на последних 30 днях и всех проектах, а значение, которого отчёт не предлагает, приводится к значению по умолчанию.

Таблица слитых запросов на слияние

Таблица «Слитые запросы на слияние» перечисляет слитые запросы на слияние периода, по строке на запрос со всеми метриками, начиная с последнего слитого. Таблица показывает 20 строк; кнопка «Развернуть» под ней добавляет следующие 20.

Слитые запросы на слияние

Запрос на слияние	Автор	Слит ↓	Дней до слияния	Коммиты	Изменённых строк	Изменённых файлов	Строк на коммит	Коммит → комментарий	Комментарий → коммит	Коммит → слияние
Change 1 in API	Administrator	31 авг. 2026 г.	1	1	5	1	5	1.5h	9h	1.5d
Change 2 in Web	Naomi Leffler	31 авг. 2026 г.	3	2	30	2	15	3h	18h	3d
Change 3 in Runner	Alla Steuber	30 авг. 2026 г.	1	3	105	3	35	5h	1.5d	30min
Change 4 in API	Susannah Steuber	30 авг. 2026 г.	4	5	350	4	70	1.4d	3d	2h

Выбор заголовка измеряемой колонки сортирует таблицу по ней в любом направлении; колонки «Запрос на слияние» и «Автор» не сортируются. Запрос на слияние, о котором колонка ничего не говорит, имеет пустую ячейку и в любом направлении оказывается в конце.

Колонка	Содержимое
«Запрос на слияние»	Заголовок со ссылкой на запрос на слияние
«Автор»	Автор запроса на слияние со ссылкой на профиль
«Слит»	Дата слияния
«Дней до слияния»	Полных дней от открытия запроса на слияние до его слияния
«Коммиты»	Число коммитов
«Изменённых строк»	Число изменённых строк
«Изменённых файлов»	Число изменённых файлов
«Строк на коммит»	Изменённые строки, делённые на число коммитов, с точностью до двух знаков
«Коммит → комментарий»	Время от открытия запроса на слияние до первого комментария
«Комментарий → коммит»	Время от первого комментария до последнего коммита; пусто, если после первого комментария коммитов не было
«Коммит → слияние»	Время от последнего коммита до слияния

Колонки времени показывают сокращённое значение: `30s`, `45min`, `2.5h`, `3d`.

Заполнение колонок для существующих запросов на слияние

Колонки, которые читаются из изменений запроса на слияние, — «Коммиты», «Изменённых строк», «Изменённых файлов», «Строк на коммит», а также даты первого и последнего коммита, стоящие за «Комментарий → коммит» и «Коммит → слияние», — для существующих запросов на слияние заполняет задача, запускаемая из консоли инстанса. Задача берёт каждый запрос на слияние, у которого есть изменения, и читает их последнюю версию; запрос без изменений пропускается. `GROUP=<GROUP>` ограничивает задачу проектами одной группы и её подгрупп, где `<GROUP>` — полный путь группы; без `GROUP` задача берёт все запросы на слияние инстанса. Путь, по которому группы нет, останавливает задачу с сообщением `No group at <GROUP>`.

Пакет для ОС Linux

```
sudo gitlab-rake fe:analytics:backfill_merge_request_metrics GROUP=<GROUP>
```

Omnibus Docker

```
docker exec code gitlab-rake fe:analytics:backfill_merge_request_metrics GROUP=<GR
OUP>
```

Helm-чарт

```
d8 k -n code exec deploy/code-toolbox -- gitlab-rake
fe:analytics:backfill_merge_request_metrics GROUP=<GROUP>
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rake
fe:analytics:backfill_merge_request_metrics GROUP=<GROUP>
```

Команда ставит в очередь фоновую задачу на каждый запрос на слияние и печатает `Enqueued a metrics refresh for 1250 merge requests`; колонки заполняются по мере выполнения задач. Задача для запроса на слияние, в который за это время была выполнена отправка изменений, ничего не делает: эта отправка сама записала колонки. Повторный запуск перезаписывает колонки по последним изменениям.

CI/CD

Непрерывная сборка и поставка ПО

Deckhouse Code использует файл `.gitlab-ci.yml` для автоматизации процессов тестирования, сборки и деплоя с помощью CI/CD-пайплайнов. Это позволяет реализовать следующий функционал:

- Автоматическую проверку кода при каждом коммите.

- Гибкую настройку этапов выполнения (stages) и задач (jobs).
- Параллельное выполнение задач для ускорения сборки и тестирования.
- Добавление произвольных шагов тестирования с учётом специфики проекта.
- Запуск пайплайнов по различным событиям: коммиты, merge requests, создание тегов и др.

Защищённые окружения

Защищённые окружения ограничивают развёртывание. Задачу развёртывания могут запускать только те, кому это явно разрешено, а для самого развёртывания могут дополнительно требоваться одобрения. Обычно защита используется для окружений, которые обслуживают пользователей, например `production`.

Защита настраивается в проекте для имени окружения и в группе для уровня развёртывания. Правило группы действует на все проекты этой группы и её подгрупп.

Принцип работы защищённых окружений

Список «Разрешено развёртывать» в правиле задаёт, кто может запускать задачу развёртывания в это окружение. Он же определяет, кто может изменять, останавливать и удалять само окружение. Список «Правила одобрения» задаёт, чьё одобрение требуется для развёртывания и сколько одобрений должен дать каждый субъект.

Значение имени правила зависит от того, кому правило принадлежит:

Владелец правила	Имя правила
Проект	Точное имя окружения проекта. Имя чувствительно к регистру, поэтому правило с именем `Production` не защищает окружение `production`
Группа	Один уровень развёртывания: `production`, `staging`, `testing`, `development` или `other`. Правило действует на все окружения этого уровня в проектах группы и её подгрупп

Уровень развёртывания окружения задаётся параметром `deployment_tier` в конфигурационном файле CI/CD. Если параметр не указан, Deckhouse Code определяет уровень по наличию в имени окружения следующих подстрок без учёта регистра:

Уровень	Подстроки в имени окружения
`development`	`dev`, `review`, `trunk`
`testing`	`test`, `tst`, `int`, `acpt`, `accept`, `qa`, `qc`, `control`, `quality`
`staging`	`stg`, `stag`, `modl`, `model`, `pre`, `demo`, `non`
`production`	`prod`, `prd`, `live`

Строки проверяются в этом порядке, и уровень определяет первое совпадение. Окружение `dev-live` получает уровень `development`. Имя, в котором нет ни одной из этих подстрок, получает уровень `other`.

Развёртывание в окружение регулируется правилом проекта для его имени вместе с правилами групп для его уровня, и все эти правила действуют одновременно. Для запуска задачи развёртывания пользователь должен быть указан в списке «Разрешено развёртывать» каждого действующего правила, а для выполнения развёртывания должны быть собраны все требуемые одобрения по правилам одобрения этих правил.

Администраторы инстанса могут развёртывать по любому правилу и одобрять по любому правилу одобрения. Эти права, как и остальные исключения для администратора на этой странице, действуют независимо от режима администратора.

Администратор, который не состоит в проекте, получает доступ к проекту и его задачам после переключения в режим администратора, если он включён в инстансе Deckhouse Code. Исключение — развёртывание, созданное собственной задачей администратора. Оно подчиняется разделу «Одобрение собственного развёртывания» наравне с остальными.

Предварительные требования

Перед настройкой защищённых окружений убедитесь, что выполнены следующие требования:

- в проекте включён механизм CI/CD;
- в конфигурационном файле CI/CD есть задача с параметром `environment`;
- для настройки правил проекта у пользователя есть роль **Мейнтейнер** или **Владелец**, а для настройки правил группы — роль **Владелец** в группе.

Правило проекта можно настроить до создания окружения. После создания окружение сразу будет защищено, если его имя совпадает с именем правила.

Защита окружения в проекте

Защитить окружение проекта можно через веб-интерфейс или API.

Защита через веб-интерфейс

Чтобы защитить окружение:

1. Откройте проект.
2. Перейдите в «Настройки» → «CI/CD».
3. Разверните раздел «Защищённые окружения».
4. Нажмите «Добавить защищённое окружение».
5. В поле «Имя окружения» укажите точное имя окружения.
6. В списке «Разрешено развёртывать» выберите роли, пользователей или группы. Подробнее — в разделе «Разрешения на развёртывание».
7. В списке «Правила одобрения» выберите субъектов, чьё одобрение требуется для развёртывания, и задайте значение «Требуемые одобрения» для каждой роли и группы. Подробнее — в разделе «Правила одобрения».
8. Нажмите «Создать защищённое окружение».

Каждый субъект может быть указан в списке «Разрешено развёртывать» и в списке «Правила одобрения» не более одного раза в каждом. При этом один и тот же субъект может одновременно присутствовать в обоих списках.

Правило появляется в таблице «Настроенные защищённые окружения». Правила, унаследованные от групп, перечислены отдельно в таблице «Унаследованные защищённые окружения».

Чтобы изменить правило, нажмите «Редактировать» в его строке. Имя окружения задаётся при создании правила и позже не меняется. Чтобы защитить другое имя, создайте новое правило.

Чтобы снять защиту с окружения, нажмите «Удалить» в его строке и подтвердите удаление в диалоге «Удалить защищённое окружение?». Вместе с правилом удаляются его записи доступа к развёртыванию и правила одобрения.

Защита через API

Чтобы защитить окружение через API, выполните запрос:

```
curl --request POST --header "PRIVATE-TOKEN: <TOKEN>" \
  --header "Content-Type: application/json" \
  --data '{
    "name": "production",
    "deploy_access_levels": [{ "access_level": 40 }],
    "approval_rules": [{ "access_level": 40, "required_approvals": 2 }]
  }' \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments"
```

Где:

- `<TOKEN>` — токен для аутентификации в Deckhouse Code;
- `<HOST>` — адрес инстанса Deckhouse Code;
- `<PROJECT_ID>` — идентификатор проекта.

В запросе можно указать следующие параметры:

Параметр	Обязателен при создании	Описание
`name`	Да	Точное имя защищаемого окружения. Сопоставляется с учётом регистра и позже не меняется
`deploy_access_levels`	Да	Записи списка «Разрешено развёртывать». Каждая запись задаёт субъекта через `access_level`, `user_id` или `group_id`, а запись с группой дополнительно принимает `group_inheritance_type`
`approval_rules`	Нет	Правила одобрения. Каждая запись задаёт субъекта так же и добавляет `required_approvals`
`required_approval_count`	Нет	Устаревший параметр, оставленный для совместимости. Принимает только значение `0`; количество одобрений задаётся в каждом правиле через `approval_rules`

Параметр `access_level` принимает следующие значения:

- `20` — Наблюдатель;
- `30` — Разработчик;
- `40` — Мейнтейнер;
- `60` — Администраторы инстанса.

Поле `access_level_description` в ответе содержит:

- для `30` — `Developers + Maintainers`;
- для `40` — `Maintainers`;
- для `20` и `60` — пустое значение;
- для записи с пользователем или группой — имя пользователя или группы.

Запись, переданная с `id`, изменяется, а без `id` — создаётся. Чтобы удалить запись, передайте её `id` вместе с `"_destroy": true`. Чтобы сохранить текущего субъекта записи и

изменить только `group_inheritance_type` или `required_approvals`, передайте `id` без `access_level`, `user_id` и `group_id`.

> Если в запросе указан субъект, не подходящий для правила, запрос отклоняется целиком. Частично применённая конфигурация не сохраняется.

Получение конфигурации:

```
# Все правила проекта, отсортированные по имени.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments"

# Правила, унаследованные от групп проекта, вместе с группой-владельцем.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments/inherited"

# Одно правило.
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments/
<ENVIRONMENT_NAME>"
```

Изменение конфигурации:

```
# Изменение списков существующего правила.
curl --request PUT --header "PRIVATE-TOKEN: <TOKEN>" \
  --header "Content-Type: application/json" \
  --data '{ "approval_rules": [{ "access_level": 40, "required_approvals": 1 }] }'
\
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments/
<ENVIRONMENT_NAME>"

# Снятие защиты.
curl --request DELETE --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/protected_environments/
<ENVIRONMENT_NAME>"
```

Оба эндпоинта со списками правил поддерживают постраничный вывод. Номер страницы задаётся параметром `page`, а количество записей на странице — `per_page`.

Если имя окружения содержит слеш, закодируйте его в пути запроса как `%2F`. Например, для окружения `review/eu` используйте `review%2Feu`. Незакодированный слеш воспринимается как разделитель сегментов пути, поэтому запрос с `review/eu` не соответствует маршруту и возвращает `404`.

Возможные коды ответа:

Код	Описание
`200`	Конфигурация возвращена или изменена
`201`	Защищённое окружение создано
`204`	Защита снята
`400`	Запрос отклонён
`401`	Запрос выполнен без аутентификации
`403`	Роль пользователя не позволяет выполнить операцию
`404`	Проект или защищённое окружение не найдены (имя окружения сопоставляется с учётом регистра)
`409`	Защищённое окружение с таким именем уже есть в проекте, это же правило изменяется другим запросом либо параллельный запрос только что добавил того же субъекта
`422`	У субъекта записи уже есть запись в этом же списке либо в параметре `required_approval_count` передано положительное значение

Код `400` возвращается, если:

- субъект не подходит для правила;
- значение `required_approvals` находится вне допустимого диапазона;
- в `required_approval_count` передано отрицательное значение;
- вложенная запись с указанным `id` не принадлежит правилу.

Защита окружений в группе

Правило группы защищает окружения по уровню развёртывания, поэтому одно правило охватывает окружения этого уровня во всех проектах группы и её подгрупп.

Чтобы создать правило группы:

1. Откройте группу.
2. Перейдите в «Настройки» → «CI/CD».
3. Разверните раздел «Защищённые окружения».

4. Нажмите «Добавить защищённое окружение».
5. В поле «Имя окружения» выберите уровень развёртывания:
    - «Рабочая среда»;
    - «Предпродакшен»;
    - «Тестирование»;
    - «Разработка»;
    - «Другое».
6. В списках «Разрешено развёртывать» и «Правила одобрения» выберите субъектов.
7. Нажмите «Создать защищённое окружение».

Субъекты правила группы ограничены её иерархией:

Субъект	Требования
Роль	В форме доступен только вариант «Мейнтейнер». Через API правило группы принимает те же значения `access_level`, что и правило проекта
Пользователь	Имеет в группе роль **Мейнтейнер** или **Владелец**, в том числе унаследованную из родительской группы
Группа	Сама группа, одна из её подгрупп или группа, которой предоставлен доступ к ней

В настройках проекта правила групп отображаются в таблице «Унаследованные защищённые окружения» со ссылкой на группу-владельца в столбце «Унаследовано от» и с отметкой «Уровень деплоя» рядом с именем. Такое правило нельзя изменить или удалить в проекте. Чтобы изменить его, откройте настройки группы-владельца.

Чтобы создать правило группы через API, используйте `/groups/:id/protected_environments` с теми же параметрами, что и для правила проекта. При этом параметр `name` должен содержать только уровень развёртывания; для любого другого значения API возвращает `400`.

Разрешения на развёртывание

Список «Разрешено развёртывать» определяет, кто может запускать задачу развёртывания в окружение. В списке можно комбинировать роли, отдельных пользователей и группы. Доступ получает каждый, кто подходит хотя бы под один из пунктов.

Доступные варианты:

Пункт списка	Кому предоставляет доступ	Условия и ограничения
«Разработчик»	Участникам проекта с ролью **Разработчик** и выше	Доступен для правил проекта
«Мейнтейнер»	Участникам проекта с ролью **Мейнтейнер** и выше	—
«Администраторы инстанса»	Администраторам инстанса	Задаётся только через API значением `"access_level": 60`. В форме защищённого окружения для роли доступны только «Разработчик» и «Мейнтейнер»
Пользователь	Указанному пользователю	Доступ пользователя в проекте должен быть **Разработчик** или выше
Группа	Участникам указанной группы	Группе должен быть предоставлен доступ к проекту. Доступ получают только участники с ролью **Наблюдатель** или выше в проекте. Состав учитываемых участников определяется селектором «Членство в группе»

Селектор «Членство в группе» принимает следующие значения:

Значение	Описание
«Прямые участники»	Только прямые участники выбранной группы
«Унаследованные участники»	Участники выбранной группы, её родительских групп и групп, которым предоставлен доступ к ней

Если список «Разрешено развёртывать» пуст, в таблице отображается «Никто». При этом:

- участники проекта не могут запускать задачу развёртывания в это окружение независимо от роли;
- развёртывание остаётся доступно администраторам инстанса;

- пустой список любого действующего правила запрещает развёртывание в окружение, даже если другое действующее правило его разрешает;
- пустой список правила группы отменяет разрешения правила проекта.

Изменение, после которого список останется без записей, отклоняется. При этом список может опустеть, если указанные в нём участники или группы потеряют доступ к проекту, например при удалении участника из проекта или отзыве доступа у группы.

Пользователь без разрешения на развёртывание в защищённое окружение не может выполнять следующие действия с задачей развёртывания:

- запускать;
- отменять;
- перезапускать;
- очищать.

Изменения роли или членства в группе учитываются сразу при проверке прав.

> Добавление пользователя в список «Разрешено развёртывать» не повышает его роль в проекте. Для доступа к окружению требуется роль **Наблюдатель** или выше.

Правила одобрения

В правиле одобрения задаются субъект и необходимое количество одобрений. Поле «Требуемые одобрения» принимает значения от `1` до `5`. Для субъекта-пользователя оно всегда равно `1`, поскольку один пользователь принимает по правилу одно решение.

Субъекты выбираются так же, как в списке «Разрешено развёртывать». Пользователь, который подходит под правило одобрения, может одобрить развёртывание без права запускать его задачу.

Решение относится к одному правилу, а одобрения всех правил учитываются вместе.

- Развёртывание считается одобренным, когда по каждому правилу всех действующих защищённых окружений получено требуемое количество одобрений.
- Пользователь, который подходит под несколько правил, принимает решение по каждому из них отдельно. Если в правиле проекта и правиле группы указан один и тот же субъект, одобрения по этим правилам учитываются независимо, поэтому один человек принимает решение по каждому из них.
- Одобрение учитывается, пока его автор подходит под то правило, для которого оно принято. Если автор теряет роль или членство в группе, названные правилом, одобрение перестаёт учитываться, и развёртывание ожидает оставшиеся одобрения.
- Отклонение продолжает действовать и после того, как его автор перестал подходить под правило.
- После первого решения об отклонении развёртывание считается отклонённым независимо от уже полученных одобрений.

При изменении правил защищённого окружения уже принятые решения сохраняются, но учитываются по новым правилам.

Управление защищённым окружением

Изменение любых настроек защищённого окружения, а также его остановка и удаление доступны пользователям из списка «Разрешено развёртывать». Одной роли **Мейнтейнер** для этого недостаточно. На администраторов инстанса ограничение не распространяется.

Действие «Очистить окружения» останавливает устаревшие окружения, кроме защищённых. Запущенными остаются окружения, имя которых указано в правиле проекта, а также окружения с уровнем, указанным в унаследованном правиле группы.

Одобрение развёртывания

Пока развёртывание ожидает одобрений, его задача остаётся ручной и не запускается. На странице развёртывания в разделе «Одобрения развёртывания» отображается прогресс по каждому действующему правилу одобрения.

Чтобы открыть страницу развёртывания, перейдите в «Деплои» → «Окружения», откройте окружение и выберите нужное развёртывание в истории. Если развёртывание ожидает одобрений, в истории отображаются статус «Ожидание» и кнопка «Посмотреть одобрения развёртывания», а в списке «Окружения» и в шапке страницы — отметка «Требуется одобрение».

Одобрение через веб-интерфейс

В разделе «Одобрения развёртывания» отображаются следующие элементы:

- Сводное состояние развёртывания показывает количество недостающих одобрений, например «Ожидается 2 из 3 одобрений». Когда все правила выполнены, вместо него отображается текст «Все необходимые одобрения получены», а после отклонения — текст «Развёртывание отклонено и не будет выполнено».
- Прогресс по каждому правилу показывает число собранных одобрений, например «Мейнтейнер: 1 из 2 одобрений», и статус «Одобрено», «Ожидает одобрения» или «Отклонено». Если два правила называют одного субъекта, отметка «Проект» или «Группа» показывает, к какому из них относится строка.
- Ссылка «Открыть настройки защищённого окружения» ведёт на настройки CI/CD проекта.

Ниже раздела расположен список «История одобрений» с принятыми решениями, начиная с последнего: автор, статус, время, строка «Правило» с субъектом правила, для которого решение принято, и комментарий.

Чтобы принять решение:

1. В поле «Правило для принятия решения» выберите правило, за которое голосуете. В списке перечислены правила, под которые вы подходите, и правила, по которым вы уже голосовали.
2. При необходимости укажите комментарий в поле «Комментарий (необязательно)». Поле принимает до 250 символов.
3. Нажмите «Одобрить» или «Отклонить».

Deckhouse Code выводит сообщение «Ваше решение сохранено.» и обновляет прогресс по правилам.

Кнопки «Одобрить» и «Отклонить» доступны отдельно для каждого правила, поэтому автор развёртывания, которому недоступно одобрение, может принять решение об отклонении по этому правилу.

Пока развёртывание ожидает одобрений, ранее принятое решение по правилу можно изменить с одобрения на отклонение. Повтор уже принятого решения сохраняет исходную запись вместе с её комментарием. Форма не отображается после получения всех необходимых одобрений или после отклонения развёртывания.

Одобрение через API

Чтобы принять решение через API, выполните запрос:

```
curl --request POST --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<PROJECT_ID>/deployments/<DEPLOYMENT_ID>/
approval?status=approved"
```

Где:

- `<TOKEN>` — токен для аутентификации в Deckhouse Code;
- `<HOST>` — адрес инстанса Deckhouse Code;
- `<PROJECT_ID>` — идентификатор проекта;
- `<DEPLOYMENT_ID>` — идентификатор развёртывания.

В запросе можно указать следующие параметры:

Параметр	Обязателен	Описание
`status`	Да	Решение: `approved` или `rejected`
`comment`	Нет	Комментарий к решению, до 255 символов
`represented_as`	Нет	Имя субъекта правила, которому засчитывается решение. Для роли используется `Developers + Maintainers` или `Maintainers`, а для пользователя или группы — их имя. Если параметр не указан, решение засчитывается самому раннему из подходящих правил. Значение сопоставляется с учётом регистра как подстрока имени субъекта

В ответе возвращается принятое решение с автором, статусом, комментарием и временем. Запрос на принятие решения, которое недоступно текущему пользователю, возвращает `403`. Некорректный запрос или запрос по развёртыванию, которое больше не ожидает одобрений, возвращает `400`.

Эндпоинты развёртываний также возвращают состояние одобрения:

- `approval_summary` — прогресс по каждому правилу;
- `approvals` — принятые решения;
- `pending_approval_count` — количество недостающих одобрений.

Эти поля возвращаются пользователям, которым доступно чтение защищённых окружений проекта. У остальных пользователей они в ответе отсутствуют.

Те же данные доступны через GraphQL API в полях `approvalSummary`, `approvals` и `pendingApprovalCount` типа `Deployment`. Каждое правило в `approvalSummary` дополнительно содержит `canApprove` и `canReject` для текущего пользователя.

В одном GraphQL-запросе эти три поля можно получить только для одного развёртывания. Запрос по списку развёртываний завершается ошибкой `"approvalSummary" field can be requested only for 1 Deployment(s) at a time.`. Для принятия решения используется мутация `approveDeployment`, которая получает глобальный идентификатор правила в поле `approvalRuleId`.

Одобрение собственного развёртывания

Если задача пользователя создала развёртывание и пользователь подходит под правило одобрения, он может отклонить это развёртывание, но по умолчанию не может его одобрить.

Чтобы разрешить такому пользователю одобрять развёртывание, перейдите в «Настройки» → «CI/CD» → «Защищённые окружения». В разделе «Параметры одобрения» установите флажок «Разрешить инициатору pipeline одобрять развёртывания» и нажмите «Сохранить изменения». Настройка действует для всего проекта и по умолчанию выключена.

Автором развёртывания считается пользователь его задачи, а для задачи без пользователя — пользователь развёртывания.

Поведение задачи развёртывания

Перед назначением раннера для задачи развёртывания в защищённое окружение проверяются права пользователя задачи, а если у задачи нет пользователя — права пользователя пайплайна.

Результат проверки определяет, что происходит с задачей.

- Если действующие правила содержат правила одобрения, Deckhouse Code создаёт развёртывание и переводит задачу в ручной режим независимо от прав автора пайплайна. Задачу нельзя запустить, пока одобрения не собраны или развёртывание отклонено.
- После того как каждое правило собрало свои одобрения, задача не запускается автоматически. Пользователь запускает её вручную. Право на запуск задачи и списки «Разрешено развёртывать» проверяются для того, кто её запускает.

- Если действующие правила не содержат правил одобрения, а пользователю развёртывание не разрешено, задача завершается с причиной «ошибка доступа к защищённому окружению» до создания развёртывания и назначения раннера.
- Задача, которая выполняет развёртывание в незащищённое окружение, выполняется как обычно.

Перед назначением раннера проверки выполняются повторно. Если в это время для задачи стали требоваться одобрения, она возвращается в ручной режим и ожидает их.

Отклонённое развёртывание

Первое принятое решение об отклонении завершает задачу развёртывания с причиной «развёртывание отклонено», а развёртывание переходит в состояние ошибки. Состояние развёртывания обновляет фоновая задача, поэтому оно меняется с задержкой.

Параметр `retry` не перезапускает отклонённую задачу развёртывания; её может перезапустить только пользователь. Отклонённое развёртывание сохраняет этот статус до перезапуска задачи. При перезапуске создаются новая задача и новое развёртывание, которое собирает одобрения заново без переноса решений по предыдущему развёртыванию.

Доступ к защищённым окружениям

Доступные действия зависят от роли пользователя и правил окружения:

Действие	Требования
Чтение защищённых окружений проекта через API	Для чтения правил пользователь должен иметь доступ к окружениям проекта, а аудитору для чтения участие в проекте не требуется. В приватном или внутреннем проекте для этого требуется роль **Наблюдатель** или выше, а в публичном — любая роль (если чтение окружений входит в общедоступный набор прав)
Чтение защищённых окружений группы через API	Пользователь входит в группу либо является аудитором
Открытие раздела «Защищённые окружения» в настройках проекта или группы	Та же роль, что и для создания правил: **Мейнтейнер** или **Владелец** в проекте, **Владелец** в группе
Создание, изменение и удаление правил проекта	Роль **Мейнтейнер** или **Владелец**
Создание, изменение и удаление правил группы	Роль **Владелец** в группе
Запуск задачи развёртывания в защищённое окружение	Пользователь указан в списке «Разрешено развёртывать» каждого действующего правила
Изменение, остановка и удаление защищённого окружения	Пользователь указан в списке «Разрешено развёртывать» каждого действующего правила
Одобрение и отклонение развёртывания	Пользователь подходит под правило одобрения этого окружения и имеет доступ на чтение развёртывания. Об ограничении на одобрение своего развёртывания — в разделе «Одобрение собственного развёртывания»

За исключением аудиторов, членство требуется и в публичном проекте. Пользователь, который не состоит в проекте и не является аудитором, не читает ни его защищённые окружения, ни состояние одобрения его развёртываний.

Экспорт и импорт защищённых окружений

Экспорт проекта содержит защищённые окружения проекта вместе с записями списка «Разрешено развёртывать» и правилами одобрения. Принятые решения не экспортируются, поскольку относятся к конкретному развёртыванию, а экспорт переносит только конфигурацию проекта.

Для поиска субъектов на целевом инстансе Deckhouse Code экспорт сохраняет:

- полный путь группы-субъекта;
- полный путь группы, которой принадлежит проект;
- имя пользователя-субъекта.

При импорте группа-субъект ищется по полному пути с учётом переноса относительного пути внутрь целевой группы, а пользователь-субъект — сначала среди импортированных участников, затем по имени пользователя.

Если субъекта сопоставить нельзя или устаревшее поле `required_approval_count` в экспорте содержит любое значение, кроме `0`, импорт проекта отклоняется целиком. Импорт с потерянным субъектом дал бы более слабую защиту, чем исходная конфигурация.

События аудита

Deckhouse Code записывает следующие события аудита для защищённых окружений:

Событие аудита	Описание
`environment_protected`	Создано защищённое окружение
`environment_unprotected`	Удалено защищённое окружение
`protected_environment_deploy_access_level_added`	В список «Разрешено развёртывать» добавлена запись
`protected_environment_deploy_access_level_updated`	Изменена запись списка «Разрешено развёртывать»
`protected_environment_deploy_access_level_deleted`	Из списка «Разрешено развёртывать» удалена запись
`protected_environment_approval_rule_added`	Добавлено правило одобрения
`protected_environment_approval_rule_updated`	Изменено правило одобрения
`protected_environment_approval_rule_deleted`	Удалено правило одобрения
`deployment_approved`	Развёртывание одобрено
`deployment_rejected`	Развёртывание отклонено

Событие об изменении конфигурации относится к проекту или группе, которой принадлежит правило, а также содержит сведения о защищённом окружении и субъекте записи. Для изменённой записи также сохраняются значения до и после изменения.

При изменении принятого решения создаётся новое событие, а при повторе того же решения — нет.

Исключение участника из проекта и отзыв доступа группы к проекту удаляют записи доступа к развёртыванию и правила одобрения этого субъекта без отдельного события. Само исключение уже записано.

Устранение проблем

Большинство таких случаев зависит от того, какие правила действуют на окружение: правило проекта для его имени и правила групп для его уровня развёртывания.

Задача завершилась с ошибкой доступа к защищённому окружению

Пользователь задачи не указан в списке «Разрешено развёртывать» одного из действующих правил либо его учётная запись удалена. Проверьте следующее:

- пользователь должен соответствовать хотя бы одной записи каждого списка «Разрешено развёртывать» правила проекта для этого имени окружения и правил групп для его уровня;
- списки «Разрешено развёртывать» не пусты, поскольку пустой список запрещает развёртывание всем;
- роль пользователя в проекте не ниже роли, выбранной в списке.

Задача развёртывания не запускается

Если у окружения есть правила одобрения, задача остаётся ручной, пока каждое правило не соберёт свои одобрения. Откройте страницу развёртывания и проверьте состояние правил в разделе «Одобрения развёртывания».

Задача также недоступна пользователю, который не указан в списке «Разрешено развёртывать», в том числе после одобрения развёртывания.

Окружение нельзя изменить, остановить или удалить

Изменение любых настроек защищённого окружения, а также его остановка и удаление доступны пользователям из списка «Разрешено развёртывать». Добавьте пользователя в этот список в правиле для данного имени окружения и в правилах групп для его уровня.

Форма одобрения не отображается

Проверьте следующее:

- развёртывание ожидает одобрений, при этом его задача находится в ручном режиме, а сводное состояние отличается от «Все необходимые одобрения получены» и от отклонения;
- ваша учётная запись подходит под правило одобрения этого окружения либо вы уже приняли решение по одному из его правил;
- развёртывание создано не вашей задачей либо в проекте включена настройка «Разрешить инициатору pipeline одобрять развёртывания». Отклонить собственное развёртывание автор может и без этой настройки.

Субъекта нельзя добавить в правило

Сообщение «У субъекта уже есть правило для этого защищённого окружения» означает, что роль, пользователь или группа уже указаны в списке, который вы изменяете. Субъекту отводится одна запись в списке «Разрешено развёртывать» и одна запись в списке «Правила одобрения». Измените существующую запись вместо добавления второй.

Влияние изменений правил на принятые решения

Состояние развёртывания вычисляется по текущим правилам.

Если автор одобрения перестал подходить под правило, для которого оно принято, одобрение перестаёт учитываться, и развёртывание ожидает оставшиеся одобрения. Само решение сохраняется и снова начинает учитываться, как только автор опять подходит под правило.

Если правило одобрения удалено, принятые по нему решения больше не учитываются. При повторном добавлении того же субъекта создаётся новое правило, для которого одобрения собираются заново.

Отклонение продолжает действовать в обоих случаях.

Правило группы не применяется к окружению

Проверьте, что уровень развёртывания окружения совпадает с именем правила группы и что проект входит в группу, которой принадлежит правило, или в одну из её подгрупп. Уровень задаётся параметром `deployment_tier` в конфигурационном файле CI/CD, а иначе определяется по имени окружения.

Правило проекта не применяется к окружению с похожим именем

Имя окружения должно точно совпадать с именем правила проекта с учётом регистра. Поэтому правило `Production` не защищает окружение `production`, а правило `production` — окружение `production/eu`. Чтобы защитить несколько окружений сразу, используйте правило группы для соответствующего уровня развёртывания.

Унаследованное правило нельзя изменить в проекте

Правила из таблицы «Унаследованные защищённые окружения» принадлежат группе. Откройте группу-владельца по ссылке в столбце «Унаследовано от» и измените правило в её настройках.

Интеграция с внешним Vault

Эта функция позволяет настроить интеграцию с Vault-сервером и использовать секреты в CI-пайплайнах. Перед началом работы необходимо настроить Vault-сервер и подготовить соответствующие роли и политики.

Настройка Vault

1. Включите аутентификацию через JWTT:

   ```
   vault auth enable jwt

   vault write auth/jwt/config \
     oidc_discovery_url="https://code.example.com" \
     bound_issuer="https://code.example.com" \
     default_role="code-role"
   ```

2. Создайте роль:

```
vault write auth/jwt/role/code-role - <<EOF
{
  "role_type": "jwt",
  "user_claim": "sub",
  "bound_audiences": ["vault"],
  "bound_claims": {
    "project_id": "23"
  },
  "policies": ["code-policy"],
  "ttl": "1h"
}
EOF
```

> Всегда используйте `bound_claims`, чтобы ограничить доступ к роли. Без этого любой JWT, выданный платформой, сможет получить доступ с этой ролью.

3. Настройте политики:

```
vault policy write code-policy - <<EOF
path "kv/data/code/vault-demo" {
  capabilities = ["read"]
}
EOF
```

Конфигурация CI

Переменные окружения

Для корректной работы с Vault в пайплайне CI/CD необходимо задать следующие переменные окружения:

- `VAULT_SERVER_URL` — **обязательно**. URL-адрес Vault-сервера (например, `https://vault.example.com`).
- `VAULT_AUTH_ROLE` — *опционально*. Название роли в Vault. Если не указано, будет использована роль по умолчанию, заданная в конфигурации метода аутентификации.
- `VAULT_AUTH_PATH` — *опционально*. Путь до метода аутентификации в Vault. По умолчанию используется `jwt`.
- `VAULT_NAMESPACE` — *опционально*. Namespace Vault, если используется многоуровневая иерархия.

Альтернативные имена переменных

У каждой переменной подключения к Vault есть альтернативное имя с префиксом `STRONGHOLD_`. Проект, который хранит секреты для CI/CD в Deckhouse Stronghold, может настроить подключение через эти имена:

Переменная	Альтернативное имя
`VAULT_SERVER_URL`	`STRONGHOLD_SERVER_URL`
`VAULT_AUTH_ROLE`	`STRONGHOLD_AUTH_ROLE`
`VAULT_AUTH_PATH`	`STRONGHOLD_AUTH_PATH`
`VAULT_NAMESPACE`	`STRONGHOLD_NAMESPACE`

Подключение настраивают только имена из таблицы. Например, переменная `STRONGHOLD_TOKEN` не читается.

Значения определяются по следующим правилам:

- значение каждой настройки определяется независимо от остальных, поэтому адрес сервера можно задать в `STRONGHOLD_SERVER_URL`, а роль — в `VAULT_AUTH_ROLE`;
- если для одной настройки заданы оба имени, используется значение переменной с префиксом `VAULT_`;
- пустое значение переменной с префиксом `STRONGHOLD_` равнозначно незаданному, поэтому при пустом `STRONGHOLD_AUTH_PATH` сохраняется путь по умолчанию `jwt`.

Альтернативные имена влияют только на подключение к Vault. Скрипт задачи CI, который читает `$VAULT_SERVER_URL`, получает значение только тогда, когда переменная с таким именем задана.

Использование секретов в CI

Для получения секретов из Vault можно использовать следующий шаблон job’а:

```
stages:
  - test
vault-login:
  stage: test
  image: ruby:3.2
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    DATABASE_PASSWORD:
      vault: code/vault-demo/DATABASE_PASSWORD@kv
      token: $VAULT_ID_TOKEN
  script: echo $DATABASE_PASSWORD
```

Параметры секрета

Пример:

```
DATABASE_PASSWORD:
  vault: code/vault-demo/DATABASE_PASSWORD@kv
  token: $VAULT_ID_TOKEN
  file: false
```

Описание параметров:

1. `vault` (обязательно) — путь к секрету в формате строки `path/to/secret/KEY@ENGINE`, где:
   - `code/vault-demo/` — путь до секрета в Vault;
   - `DATABASE_PASSWORD` — имя поля внутри секрета;
   - `kv` — точка монтирования Secret Engine (по умолчанию — `secret`).

По умолчанию используется `engine kv-v2`. Если необходимо использовать другой engine, можно указать объект вместо строки:

```
DATABASE_PASSWORD:
  vault:
    path: code/vault-demo
    field: DATABASE_PASSWORD
    engine:
      name: 'kv-v1'
      path: 'kv1'
  token: $VAULT_ID_TOKEN
  file: false
```

1. `token` (обязательно) — JWT-токен из секции `id_tokens`, используемый для аутентификации в Vault.
2. `file` (опционально, по умолчанию `true`) — определяет способ предоставления секрета:
   - `true` — секрет сохраняется во временный файл;
   - `false` — секрет передаётся как строка в переменную окружения.

Поля, включённые в JWT

Следующие поля автоматически включаются в JWT-токен и могут использоваться Vault для проверки прав доступа:

Поле	Условие появления	Описание
`jti`	всегда	Уникальный идентификатор токена
`iss`	всегда	Издатель токена (обычно URL Deckhouse Code)
`iat`	всегда	Время выпуска токена (Issued At)
`nbf`	всегда	Время, до которого токен считается недействительным
`exp`	всегда	Время истечения срока действия токена
`sub`	всегда	Subject токена (обычно ID задачи CI)
`namespace_id`	всегда	ID пространства (группы или пользователя)
`namespace_path`	всегда	Путь до пространства (например, `groups/dev`)
`project_id`	всегда	ID проекта
`project_path`	всегда	Путь до проекта
`user_id`	всегда	ID пользователя
`user_login`	всегда	Логин пользователя
`user_email`	всегда	Email пользователя
`pipeline_id`	всегда	ID CI-пайплайна
`pipeline_source`	всегда	Источник запуска пайплайна (push, schedule, MR и т.д.)
`job_id`	всегда	ID задачи CI
`ref`	всегда	Ссылка Git (Git reference)
`ref_type`	всегда	Тип ссылки Git (Git reference) (`branch` или `tag`)
`ref_path`	всегда	

Поле	Условие появления	Описание
		Полный путь ссылки Git (Git reference) (например, `refs/heads/main`)
`ref_protected`	всегда	Признак того, что объект по ссылке Git защищён
`environment`	при наличии	Название окружения (если используется)
`groups_direct`	при наличии (<200 групп)	Пути до групп, в которых состоит пользователь
`environment_protected`	при наличии	Является ли окружение защищённым
`deployment_tier`	при наличии	Тип окружения (`production`, `staging` и т.п.)
`environment_action`	при наличии	Тип действия над окружением (например, `deploy`)

Быстрый старт

В этом разделе приведён пример минимально необходимой настройки интеграции HashiCorp Vault с Deckhouse Code и проверки того, что CI job может получать секреты из Vault.

> ⚠ Данный пример предназначен только для ознакомления. Он не отражает лучшие практики безопасности и использует упрощённую конфигурацию, позволяющую быстро проверить работоспособность интеграции.

Шаг 1. Установка переменных окружения

Установите переменные окружения для Vault и Deckhouse Code.
Некоторые параметры можно оставить как есть, но `VAULT_ADDR`, `VAULT_TOKEN`, `CODE_URL` и `PROJECT_PATH` должны быть заданы вручную.

```
export VAULT_ADDR="https://vault.example.com"
export VAULT_TOKEN="<TOKEN>"

# URL-адрес Deckhouse Code.
export CODE_URL="https://code.example.com"

# Имя роли и политики Vault.
export VAULT_ROLE="code-role"
export VAULT_POLICY="code-policy"

# Путь и данные секрета.
export VAULT_SECRET_PATH="code/vault-demo"
export VAULT_SECRET_FIELD="DATABASE_PASSWORD"
export VAULT_SECRET_VALUE="super-secret-password"

# Значение claim project_path, которое будет проверять Vault.
export PROJECT_PATH="root/my-pr"
```

Шаг 2. Включение метода аутентификации JWT

Включите в Vault метод аутентификации JWT. Без этого Vault не сможет принимать ID-токены, которые Deckhouse Code передаёт в CI jobs.

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X POST "$VAULT_ADDR/v1/sys/auth/jwt" \
  -d '{"type":"jwt"}'
```

Шаг 3. Настройка JWT и OIDC

Следующий запрос:

- задаёт адрес OIDC discovery (`$CODE_URL`);
- указывает ожидаемого issuer;
- определяет *роль по умолчанию*, которую Vault выдаёт при аутентификации.

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X POST "$VAULT_ADDR/v1/auth/jwt/config" \
  --data @- <<EOF
{
  "oidc_discovery_url": "$CODE_URL",
  "bound_issuer": "$CODE_URL",
  "default_role": "$VAULT_ROLE"
}
EOF
```

Шаг 4. Монтирование Secret Engine KV v2

KV v2 — это наиболее распространённый движок секретов Vault. Следующий запрос включает его по пути `/kv`:

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X POST "$VAULT_ADDR/v1/sys/mounts/kv" \
  -d '{"type": "kv-v2"}'
```

Шаг 5. Создание тестового секрета

Создайте секрет, который затем должна будет прочитать CI job. Секрет размещается по пути `code/vault-demo` и содержит одно поле.

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X POST "$VAULT_ADDR/v1/kv/data/$VAULT_SECRET_PATH" \
  --data @- <<EOF
{
  "data": {
    "$VAULT_SECRET_FIELD": "$VAULT_SECRET_VALUE"
  }
}
EOF
```

Шаг 6. Создание ACL-политики

Политика определяет, к каким путям в Vault разрешён доступ. В следующем примере политика предоставляет только право на чтение указанного секрета.

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X PUT "$VAULT_ADDR/v1/sys/policies/acl/$VAULT_POLICY" \
  --data @- <<EOF
{
  "policy": "path \"kv/data/$VAULT_SECRET_PATH\" { capabilities = [\"read\"] }"
}
EOF
```

Шаг 7. Создание роли Vault

Роль определяет:

- тип аутентификации;
- обязательные claims в токене (`project_path`);
- политики, которые получит аутентифицированный субъект;

- допустимые аудитории (`aud`);
- TTL токена.

Deckhouse Code будет выпускать ID-токен с `aud=vault` , а Vault проверит, что значение `project_path` соответствует указанному.

```
curl \
  -H "X-Vault-Token: $VAULT_TOKEN" \
  -X POST "$VAULT_ADDR/v1/auth/jwt/role/$VAULT_ROLE" \
  --data @- <<EOF
{
  "role_type":   "jwt",
  "user_claim":  "sub",
  "bound_audiences": ["vault"],

  "bound_claims": {
    "project_path": "$PROJECT_PATH"
  },

  "policies": ["$VAULT_POLICY"],
  "ttl": "1h"
}
EOF
```

На этом настройка Vault завершена.

Тестирование интеграции в Deckhouse Code

1. Откройте проект, указанный в `PROJECT_PATH` . CI-токен проекта должен совпадать с claim `project_path` . В противном случае Vault отклонит запрос на доступ к секретам.
2. В настройках CI/CD проекта добавьте переменную `VAULT_SERVER_URL` со значением `$VAULT_ADDR` , использованным ранее. Эта переменная сообщает Deckhouse Code, куда следует направлять запросы Vault API.
3. Создайте файл `.gitlab-ci.yml` . Файл запускает тестовую CI job, которая:
   - получает ID-токен с `aud=vault` ;
   - передаёт его в Vault;
   - загружает секрет из KV;
   - выводит значение секрета.

```
stages:
  - test

vault-demo:
  stage: test
  image: alpine
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    DATABASE_PASSWORD:
      vault: code/vault-demo/DATABASE_PASSWORD@kv
      token: $VAULT_ID_TOKEN
      file: false
  script:
    - echo "Raw value (masked by Deckhouse Code):"
    - echo "$DATABASE_PASSWORD"

    - echo
    - echo "Value with spaces (not masked):"
    - printf '%s\n' "$DATABASE_PASSWORD" | sed 's/./& /g'
```

Результат

Если интеграция настроена правильно:

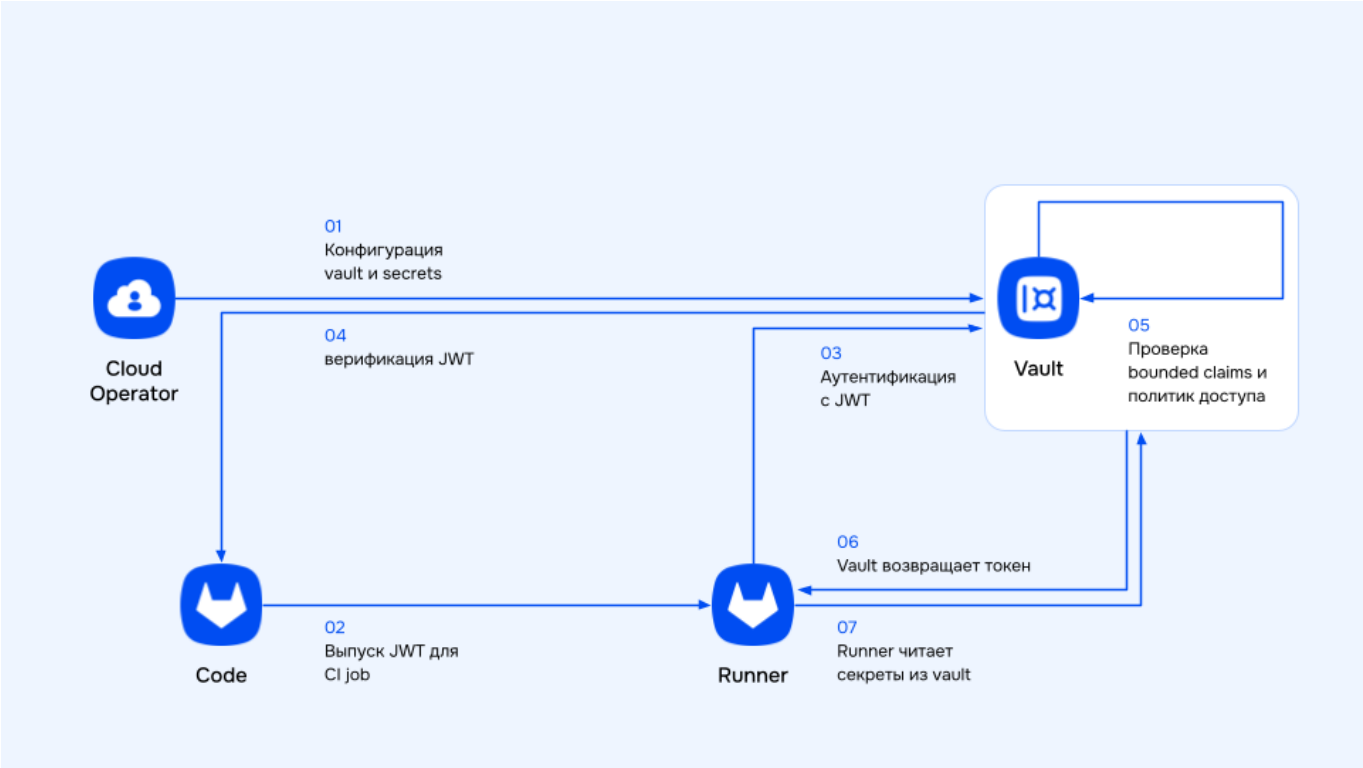
- CI job успешно получит ID-токен;
- Vault проверит значение `project_path` и выдаст доступ;
- секрет будет считан и выведен в лог.

В выводе вы увидите значение поля `DATABASE_PASSWORD`, загруженное напрямую из Vault.

Интеграция с Deckhouse Stronghold

Deckhouse Stronghold — это enterprise-решение для управления секретами, построенное на базе HashiCorp Vault. Интеграция Stronghold с Deckhouse Code позволяет безопасно использовать секреты в CI/CD-пайплайнах без необходимости хранить их в переменных проекта.

Схема взаимодействия



Преимущества интеграции:

- **Централизованное управление секретами** — все секреты хранятся в одном месте с аудитом доступа.
- **Автоматическая ротация** — секреты могут обновляться автоматически без необходимости изменения CI/CD-пайплайнов.
- **Гранулярный контроль доступа** — доступ к секретам ограничивается на основе проекта, ветки, окружения и других параметров.
- **Отсутствие секретов в репозитории** — секреты не хранятся в коде и не передаются через переменные CI.

Примеры использования

В этом разделе приведены типовые сценарии использования секретов из Vault в CI/CD-пайплайнах.

Развёртывание приложения с учётными данными базы данных

Пример получения данных для аутентификации (credentials) для подключения к PostgreSQL во время развёртывания:

```
stages:
  - deploy

deploy-production:
  stage: deploy
  image: alpine/k8s:1.28.0
  environment:
    name: production
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    DB_HOST:
      vault: apps/myapp/production/DB_HOST@kv
      token: $VAULT_ID_TOKEN
      file: false
    DB_USER:
      vault: apps/myapp/production/DB_USER@kv
      token: $VAULT_ID_TOKEN
      file: false
    DB_PASSWORD:
      vault: apps/myapp/production/DB_PASSWORD@kv
      token: $VAULT_ID_TOKEN
      file: false
  script:
    - kubectl create secret generic db-credentials \
        --from-literal=host=$DB_HOST \
        --from-literal=username=$DB_USER \
        --from-literal=password=$DB_PASSWORD \
        --dry-run=client -o yaml | kubectl apply -f -
    - kubectl rollout restart deployment/myapp
  rules:
    - if: $CI_COMMIT_BRANCH == "main"
```

Использование API-ключей для внешних сервисов

Пример интеграции с внешними API (Slack, Telegram, S3):

```
stages:
  - notify
  - backup

notify-slack:
  stage: notify
  image: curlimages/curl:latest
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    SLACK_WEBHOOK_URL:
      vault: integrations/slack/WEBHOOK_URL@kv
      token: $VAULT_ID_TOKEN
      file: false
  script:
    - |
      curl -X POST "$SLACK_WEBHOOK_URL" \
        -H "Content-Type: application/json" \
        -d "{\"text\": \"Deployment completed for $CI_PROJECT_NAME\"}"

backup-to-s3:
  stage: backup
  image: amazon/aws-cli:latest
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    AWS_ACCESS_KEY_ID:
      vault: cloud/aws/backup-user/ACCESS_KEY_ID@kv
      token: $VAULT_ID_TOKEN
      file: false
    AWS_SECRET_ACCESS_KEY:
      vault: cloud/aws/backup-user/SECRET_ACCESS_KEY@kv
      token: $VAULT_ID_TOKEN
      file: false
  script:
    - aws s3 sync ./artifacts s3://my-backup-bucket/$CI_PROJECT_NAME/$CI_COMMIT_SHA/
```

Подписание Docker-образов с помощью Cosign

Пример использования приватного ключа из Vault для подписания образов:

```
stages:
  - build
  - sign

build-image:
  stage: build
  image: docker:24.0.5
  services:
    - docker:24.0.5-dind
  script:
    - docker build -t $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA .
    - docker push $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA

sign-image:
  stage: sign
  image: bitnami/cosign:latest
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  secrets:
    COSIGN_PRIVATE_KEY:
      vault: signing/cosign/PRIVATE_KEY@kv
      token: $VAULT_ID_TOKEN
      file: true
  script:
    - cosign sign --key $COSIGN_PRIVATE_KEY $CI_REGISTRY_IMAGE:$CI_COMMIT_SHA
```

Разграничение доступа по веткам

Пример настройки различных секретов для веток `develop` и `main` с использованием bound claims по ветке (`ref`):

```
stages:
  - deploy

.deploy-template:
  image: alpine/k8s:1.28.0
  id_tokens:
    VAULT_ID_TOKEN:
      aud: vault
  script:
    - echo "Deploying from branch $CI_COMMIT_BRANCH with database $DB_HOST"
    - kubectl set env deployment/myapp DB_HOST=$DB_HOST DB_PASSWORD=$DB_PASSWORD

deploy-develop:
  extends: .deploy-template
  stage: deploy
  variables:
    VAULT_AUTH_ROLE: myapp-develop
  secrets:
    DB_HOST:
      vault: apps/myapp/develop/DB_HOST@kv
      token: $VAULT_ID_TOKEN
      file: false
    DB_PASSWORD:
      vault: apps/myapp/develop/DB_PASSWORD@kv
      token: $VAULT_ID_TOKEN
      file: false
  rules:
    - if: $CI_COMMIT_BRANCH == "develop"

deploy-main:
  extends: .deploy-template
  stage: deploy
  variables:
    VAULT_AUTH_ROLE: myapp-main
  secrets:
    DB_HOST:
      vault: apps/myapp/main/DB_HOST@kv
      token: $VAULT_ID_TOKEN
      file: false
    DB_PASSWORD:
      vault: apps/myapp/main/DB_PASSWORD@kv
      token: $VAULT_ID_TOKEN
      file: false
  rules:
    - if: $CI_COMMIT_BRANCH == "main"
```

Пример настройки роли в Vault с bound claims по ветке для разграничения доступа:

```
# Роль для `develop` — доступ только с ветки `develop`.
vault write auth/jwt/role/myapp-develop - <<EOF
{
  "role_type": "jwt",
  "user_claim": "sub",
  "bound_audiences": ["vault"],
  "bound_claims": {
    "project_path": "myteam/myapp",
    "ref": "develop",
    "ref_type": "branch"
  },
  "policies": ["myapp-develop-policy"],
  "ttl": "1h"
}
EOF

# Роль для `main` — доступ только с ветки `main` (защищённой).
vault write auth/jwt/role/myapp-main - <<EOF
{
  "role_type": "jwt",
  "user_claim": "sub",
  "bound_audiences": ["vault"],
  "bound_claims": {
    "project_path": "myteam/myapp",
    "ref": "main",
    "ref_type": "branch",
    "ref_protected": "true"
  },
  "policies": ["myapp-main-policy"],
  "ttl": "1h"
}
EOF
```

Доступ к Kubernetes от имени пользователя

Пользователи Deckhouse Code отправляют запросы к кластеру через прокси Kubernetes API агента для Kubernetes. Агент может передавать каждый такой запрос в кластер от имени пользователя, который его отправил. Кластер получает в запросе имя пользователя и его группы. Группы соответствуют ролям пользователя в проектах и группах Deckhouse Code. Права пользователя в кластере определяют привязки RBAC Kubernetes к этим именам.

Доступ от имени пользователя настраивает в файле конфигурации агента участник проекта конфигурации агента с ролью Developer или выше. Права в кластере выдаёт администратор кластера.

Настройка доступа

Требования:

- на экземпляре включён Relay (KAS);
- агент для Kubernetes развёрнут в кластере, как описано в разделе «Агенты для Kubernetes»;
- у вас есть роль Developer или выше в проекте конфигурации агента.

Добавьте ключ `user_access` в файл конфигурации агента `.gitlab/agents/<AGENT_NAME>/config.yaml`, где `<AGENT_NAME>` — имя агента. Укажите в ключе `user_access` следующие параметры:

- `access_as` — режим доступа: `user: {}` задаёт доступ от имени пользователя;
- `projects` — полные пути проектов, участники которых получают доступ (не более 500 записей);
- `groups` — полные пути групп, участники которых получают доступ (не более 500 записей).

Пример ключа `user_access`, который даёт доступ участникам проекта `group-1/project-1` и группы `group-2`:

```
user_access:
  access_as:
    user: {}
  projects:
    - id: group-1/project-1
  groups:
    - id: group-2
```

Агент читает файл конфигурации из ветки по умолчанию проекта конфигурации. Изменение в файле начинает действовать, когда оно попадает в ветку по умолчанию.

Следующие ограничения действуют, если на экземпляре выключена настройка «Включить авторизацию на уровне инстанса». Указывайте в `projects` и `groups` только проекты и группы из той же группы верхнего уровня, что и проект конфигурации агента. Записи из других групп Deckhouse Code игнорирует. Если проект конфигурации при этом находится в личном неймспейсе пользователя, Deckhouse Code игнорирует `groups`. В `projects` в этом случае действуют только проекты этого неймспейса.

Укажите в `access_as` ровно один ключ. Если вместо `user: {}` указать `agent: {}`, агент передаёт запросы от имени своего сервисного аккаунта.

Что получает кластер

API-сервер Kubernetes получает запрос пользователя со следующими данными:

- имя пользователя `gitlab:user:<USERNAME>`;
- группа `gitlab:user`, общая для всех пользователей;

- группа `gitlab:project_role:<PROJECT_ID>:<ROLE>` для каждой роли пользователя в каждом проекте из `projects`;
- группа `gitlab:group_role:<GROUP_ID>:<ROLE>` для каждой роли пользователя в каждой группе из `groups`;
- дополнительные поля `agent.gitlab.com/id`, `agent.gitlab.com/username`, `agent.gitlab.com/access_type` и `agent.gitlab.com/config_project_id`.

`<PROJECT_ID>` и `<GROUP_ID>` — числовые идентификаторы проектов и групп из файла конфигурации. Группы `gitlab:project_role` и `gitlab:group_role` пользователь получает только для проектов и групп, в которых у него есть роль Developer или выше.

Deckhouse Code учитывает роль пользователя в проекте из `projects` в следующих случаях:

- пользователь — участник проекта;
- пользователь — участник одной из родительских групп проекта;
- проект предоставлен группе, участник которой — пользователь.

Deckhouse Code учитывает роль пользователя в группе из `groups`, только если пользователь — прямой участник этой группы или одной из её родительских групп. Участие в группе, которой предоставлена группа из `groups`, Deckhouse Code не учитывает. Участие только в подгруппе группы из `groups` Deckhouse Code тоже не учитывает.

Если у пользователя нет роли Developer или выше ни в одном проекте из `projects` и ни в одной группе из `groups`, прокси отклоняет его запросы.

Пользователь получает свою роль и все младшие роли. Порядок ролей от младшей к старшей: `guest`, `planner`, `reporter`, `security_manager`, `developer`, `maintainer`, `owner`. Роль `security_manager` входит в этот список, только если на экземпляре включена роль «Менеджер безопасности». Поэтому привязка к `gitlab:project_role:<PROJECT_ID>:developer` действует и на участников проекта с ролями Maintainer и Owner.

Например, участник проекта 11 с ролью Developer получает следующие группы:

- `gitlab:user`;
- `gitlab:project_role:11:guest`;
- `gitlab:project_role:11:planner`;
- `gitlab:project_role:11:reporter`;
- `gitlab:project_role:11:developer`;
- `gitlab:project_role:11:security_manager`, если на экземпляре включена роль «Менеджер безопасности».

Сервер агентов Kubernetes кеширует результат авторизации запроса, по умолчанию на одну минуту. Поэтому кластер получает новые группы пользователя с задержкой, если пользователя добавили в проект или группу, удалили из них или изменили его роль.

Выдача прав в кластере

Пока группа пользователя не указана в RoleBinding или ClusterRoleBinding, у пользователя нет прав в кластере.

Например, чтобы участники проекта 123 с ролями Maintainer и Owner могли читать большинство объектов в кластере, примените следующий манифест. Манифест связывает группу `gitlab:project_role:123:maintainer` со встроенной ClusterRole `view`:

```
apiVersion: rbac.authorization.k8s.io/v1
kind: ClusterRoleBinding
metadata:
  name: project-123-maintainers-view
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: ClusterRole
  name: view
subjects:
  - apiGroup: rbac.authorization.k8s.io
    kind: Group
    name: gitlab:project_role:123:maintainer
```

Чтобы выдать права только в одном неймспейсе, создайте RoleBinding в этом неймспейсе. Правила привязок RBAC описывает страница документации Kubernetes Using RBAC Authorization.

Подключение к кластеру

1. Откройте проект конфигурации агента и перейдите в «Управление» → «Кластеры Kubernetes». Запишите числовой идентификатор агента.
2. Создайте персональный токен доступа с областями `k8s_proxy` и `read_user`.
3. Запросите метаданные экземпляра:

   ```
   curl --header "PRIVATE-TOKEN: <TOKEN>" "https://<HOSTNAME>/api/v4/metadata"
   ```

   Запишите адрес прокси Kubernetes API из поля `kas.externalK8sProxyUrl` ответа. Если поле `kas.enabled` равно `false`, сервер агентов Kubernetes на экземпляре выключен. Все поля ответа описывает страница Metadata.
4. Добавьте в `kubeconfig` кластер с адресом прокси Kubernetes API:

   ```
   d8 k config set-cluster <CLUSTER_NAME> --server "<K8S_PROXY_URL>"
   ```
5. Добавьте в `kubeconfig` учётные данные. Токен в них состоит из префикса `pat:`, идентификатора агента и персонального токена доступа:

```
d8 k config set-credentials <USER_NAME> --token "pat:<AGENT_ID>:<TOKEN>"
```

6. Создайте контекст и переключитесь на него:

   ```
   d8 k config set-context <CONTEXT_NAME> --cluster <CLUSTER_NAME> --user
   <USER_NAME>
   d8 k config use-context <CONTEXT_NAME>
   ```

7. Проверьте имя пользователя и группы, которые получает кластер:

   ```
   d8 k auth whoami
   ```

   Команда выводит имя пользователя `gitlab:user:<USERNAME>` и группы из раздела «Что получает кластер».

Обозначения:

- `<HOSTNAME>` — адрес экземпляра Deckhouse Code;
- `<K8S_PROXY_URL>` — адрес прокси Kubernetes API из поля `kas.externalK8sProxyUrl`;
- `<AGENT_ID>` — идентификатор агента из первого шага;
- `<TOKEN>` — персональный токен доступа;
- `<CLUSTER_NAME>`, `<USER_NAME>`, `<CONTEXT_NAME>` — имена кластера, учётных данных и контекста в `kubeconfig`, которые вы выбираете сами.

Аналитика CI/CD

Отчёт «Аналитика CI/CD» показывает количество, продолжительность и распределение по статусам завершившихся пайплайнов, созданных за выбранный период (в отдельном проекте или в проектах группы и её подгрупп).

Доступность отчёта

Отчёт проекта доступен участникам с ролью «Наблюдатель» и выше, в том числе тем, кто получил роль в родительской группе.

Настройка проекта «Видимость пайплайнов на уровне проекта» открывает отчёт всем, кому виден сам проект: пользователям с типом «Аудитор», участникам с ролью «Гость», в публичном проекте — любому посетителю, включая не выполнившего вход, во внутреннем — любому вошедшему пользователю. При выключенной настройке отчёт остаётся доступен участникам с ролью «Наблюдатель» и выше.

Для доступа к отчёту проекта читателю нужен также доступ к пайплайнам этого проекта. Если видимость функции «CI/CD» сужена до состава участников, отчёт закрывается для всех, кто не входит в проект, — в том числе для посетителей публичного проекта. Пользователи с типом «Аудитор» сохраняют доступ независимо от этой настройки.

Если видимость функции «Аналитика» ограничена участниками проекта, отчёт закрыт для всех остальных. Пользователи с типом «Аудитор» сохраняют доступ. Если функция выключена, отчёт закрыт для всех, включая участников проекта.

Выключение функции «CI/CD» отчёт не закрывает: у проекта сохраняется уже накопленная история пайплайнов, и отчёт продолжает её отображать. При этом пункт «Аналитика CI/CD» пропадает из меню «Аналитика» — то же происходит и при пустом репозитории проекта. В обоих случаях отчёт остаётся доступен по прямой ссылке.

Отчёт группы доступен участникам с ролью «Наблюдатель» и выше, в том числе тем, кто получил роль в родительской группе, и пользователям с типом «Аудитор».

Просмотр отчёта

Чтобы просмотреть отчёт группы:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «Аналитика CI/CD».

Чтобы просмотреть отчёт проекта:

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «Аналитика CI/CD».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP_PATH>/-/analytics/ci_cd_analytics`
Проект	`https://<HOST>/<PROJECT_PATH>/-/analytics/ci_cd_analytics`

Вместо `<GROUP_PATH>` укажите полный путь группы, включая родительские группы (например, `team/backend`). Вместо `<PROJECT_PATH>` укажите полный путь проекта, включая группы и подгруппы, или имя пользователя для личного проекта (например, `team/backend/app` или `username/app`).

Отчёт группы охватывает пайплайны её проектов и проектов её подгрупп, включая архивные проекты.

Метрики

В строке над графиками показаны итоги периода:

Метрика	Содержимое
«Общее количество запусков пайплайнов»	Число завершившихся пайплайнов, созданных в периоде
«Медианная продолжительность»	Медиана (p50) продолжительности запусков, длившихся больше нуля секунд
«Доля сбоев»	Доля неуспешных запусков среди тех, что завершились успехом или неудачей
«Доля успехов»	Доля успешных запусков среди тех, что завершились успехом или неудачей
«Доля остальных»	Доля отменённых и пропущенных запусков среди всех запусков периода

«Доля сбоев» и «Доля успехов» считаются от запусков, завершившихся успехом или неудачей. Каждая из них округляется до целого отдельно, поэтому в сумме они дают около 100 %: при 5 успешных и 3 неуспешных запусках доля сбоев равна 38 %, а доля успехов 63 %. «Доля остальных» считается от всех запусков периода, поэтому сумма трёх показателей может превышать 100 %. Формула расчёта показана в подсказке значка «Как это рассчитывается?» рядом с «Долей сбоев» и «Долей остальных».

Доля, для знаменателя которой в периоде нет ни одного запуска, показана прочерком. Например, если в периоде есть только отменённые и пропущенные запуски, в «Доле сбоев» и «Доле успехов» показан прочерк, а в «Доле остальных» 100 %.

Когда в проекте есть неуспешные запуски, ссылка «Посмотреть все» под «Долей сбоев» открывает список пайплайнов проекта, отфильтрованный по неуспешным запускам. Выбранная в отчёте ветка переносится в этот список фильтром; со значением «Все ветки» список открывается без фильтра по ветке.

Графики

График «Продолжительность» содержит по линии на каждый перцентиль продолжительности пайплайна:

- «Медиана (50-й процентиль)»;
- «95-й перцентиль».

Вертикальная ось графика размечена в минутах.

График «Статус» содержит по столбцу на каждый день, разделённому на серии:

- «Успешные»;
- «Ошибка»;
- «Другие (отменённые, пропущенные)».

По вертикальной оси графика — количество пайплайнов.

Фильтры

Фильтры над графиками ограничивают метрики и оба графика:

Фильтр	Значения
«Источник»	Событие, создавшее пайплайн: «Push», «Расписание», «Событие запроса на слияние» и остальные. По умолчанию — «Любой источник»
«Ветвь»	Ветка пайплайна. Изначально выбрана ветка по умолчанию проекта, а значение «Все ветки» снимает фильтр. В отчёте группы такого фильтра нет
«Диапазон дат»	«Прошлая неделя», «Последние 30 дней», «Последние 90 дней», «Последние 180 дней». По умолчанию — «Прошлая неделя»

Фильтр «Ветвь» также учитывает пайплайны запросов на слияние, для которых выбранная ветка является исходной. Такие пайплайны попадают в отчёт, если для фильтра «Источник» выбрано «Любой источник» или «Событие запроса на слияние». При выборе другого источника отчёт учитывает только пайплайны самой ветки.

Выбранные фильтры сохраняются в адресе страницы в параметрах `source` и `time`, поэтому настроенный отчёт можно передать ссылкой. В отчёте проекта туда же попадает ветка, в параметре `branch`.

Как считаются пайплайны

Отчёт учитывает пайплайны, созданные в периоде и дошедшие до завершённого статуса: успешные, неуспешные, отменённые и пропущенные. Пайплайн относится к дню своего создания, отсчитанному по UTC, поэтому пайплайн, созданный в один день и завершившийся на следующий, учитывается в дне создания.

Период заканчивается началом текущего дня, поэтому пайплайны, созданные сегодня, в отчёт не попадают, а на графиках нет точки за сегодня.

Перцентили графика «Продолжительность» и метрика «Медианная продолжительность» не учитывают запуски без продолжительности и запуски с нулевой продолжительностью. За день без таких запусков точки на графике нет, а за период без таких запусков вместо медианы показан прочерк.

Отчёт группы с теми же фильтрами и периодом обновляется не чаще раза в минуту, поэтому при повторном просмотре он может отставать от текущего состояния до минуты. Отчёт проекта формируется по каждому запросу.

Если отчёт строится слишком долго, вместо графиков страница показывает сообщение об этом и предлагает сузить отчётный период. Выберите меньшее значение «Диапазона дат» и откройте отчёт заново.

DORA-метрики

Отчёт «DORA-метрики» измеряет поставку группы или проекта за выбранный период: как часто выполняются успешные развёртывания в production-окружения, сколько слитое изменение ждёт своего развёртывания, сколько инциденты остаются открытыми и сколько инцидентов приходится на одно развёртывание. Отчёт группы охватывает проекты группы и её подгрупп.

Доступность отчёта

Отчёт проекта доступен участникам с ролью «Наблюдатель» и выше, в том числе тем, кто получил роль в родительской группе. Участники с ролью «Гость» не видят пункт «DORA-метрики» в меню проекта, а прямой адрес отвечает кодом `404`. Уровень видимости проекта этот доступ не расширяет: в публичном или внутреннем проекте пользователь, не состоящий в проекте, отчёт не видит. Выключение функции «Аналитика» проекта закрывает отчёт для всех: пункт меню пропадает, а прямой адрес отвечает кодом `404`.

Отчёт группы доступен участникам с ролью «Наблюдатель» и выше, в том числе тем, кто получил роль в родительской группе, а также пользователям с типом «Аудитор». Участники с ролью «Гость» не видят пункт в меню группы, а прямой адрес отвечает кодом `404`. Для отчёта проекта пользователю с типом «Аудитор» нужна роль «Наблюдатель» и выше в проекте или в его родительской группе.

Просмотр отчёта

Чтобы просмотреть отчёт группы:

1. В верхней панели нажмите «Искать или перейти к…» и выберите группу.
2. В меню группы перейдите в «Аналитика» → «DORA-метрики».

Чтобы просмотреть отчёт проекта:

1. В верхней панели нажмите «Искать или перейти к…» и выберите проект.
2. В меню проекта перейдите в «Аналитика» → «DORA-метрики».

Также отчёт можно открыть по прямому адресу:

Отчёт	Адрес
Группа	`https://<HOST>/groups/<GROUP>/-/analytics/dashboards/dora_metrics`
Проект	`https://<HOST>/<GROUP>/<PROJECT>/-/analytics/dashboards/dora_metrics`

На странице расположены список периодов, ряд карточек со значением каждой метрики за период и по графику на каждую метрику со значениями по дням.

Метрики

Метрика	Подпись карточки	Значение
«Частота деплоя»	«Успешные деплои в день.»	Число успешных развёртываний, завершённых в периоде, делённое на число дней периода
«Время проведения изменений»	«Медианное время от слияния до деплоя.»	Медианное время от слияния запроса на слияние до завершения успешного развёртывания, которое принесло его в окружение
«Время восстановления сервиса»	«Медианное время, в течение которого инциденты остаются открытыми.»	Медианное время от создания инцидента до его закрытия по инцидентам, закрытым в периоде
«Частота неуспешных изменений»	«Инциденты, делённые на количество деплоев.»	Число инцидентов, созданных в периоде, делённое на число успешных развёртываний, завершённых в нём

Отчёт учитывает развёртывания в окружения с уровнем развёртывания `production`. Развёртывание связывается с запросами на слияние, коммиты которых оно принесло в окружение после предыдущего успешного развёртывания в то же окружение; первое развёртывание в окружение связывается со всеми запросами на слияние, слитыми в развёрнутую ветку. Запрос на слияние, слияние которого зафиксировано после завершения развёртывания, во время проведения изменений не попадает.

Инциденты — задачи с типом «Инцидент». Инцидент относится к проекту в целом и учитывается как инцидент уровня `production`, поэтому «Время восстановления сервиса» и «Частота неуспешных изменений» имеют значения только в проекте с окружением этого уровня. Инцидент, помеченный как дубликат другой задачи, исключается из обеих метрик; переоткрытый инцидент перестаёт считаться закрытым.

Значения хранятся по окружениям и по дням; день начинается и заканчивается по часовому поясу сервера. День пересчитывается заново при событии этого дня: через 5 минут после успешного развёртывания, когда с ним связаны принесённые им запросы на слияние, и сразу при создании, закрытии, переоткрытии инцидента, пометке его как дубликата или смене его типа на «Инцидент» или с него. За период суммируются развёртывания и инциденты его дней, а время проведения изменений и время восстановления сервиса берутся как медианы дневных медиан. Период без успешных развёртываний значения «Частоты неуспешных изменений» не имеет.

Значения записываются в момент события. Развёртывания и инциденты, произошедшие до обновления, которое принесло отчёт, получают значения после загрузки истории администратором, как описано в разделе Загрузка истории; то же относится к дням, посчитанным до изменения списка веток из раздела Ветки, учитываемые во времени проведения изменений.

Карточки метрик

Ряд над графиками содержит по карточке на каждую метрику с её названием, подписью и значением за весь период:

- «Частота деплоя» — развёртываний в день, с суффиксом `/d`;
- «Время проведения изменений» и «Время восстановления сервиса» — в днях, с суффиксом `d`;
- «Частота неуспешных изменений» — в процентах.

Метрика без значения в периоде показана прочерком. Если значения не удалось загрузить, карточка показывает «Не удалось загрузить DORA-метрики.»

Графики

Под карточками расположено по графику на каждую метрику с названием и подписью её карточки: линия с точкой на каждый день периода, вертикальная ось в единицах карточки, горизонтальная ось — «Дата». Пунктирная линия через весь график несёт значение карточки за весь период; на графиках «Время проведения изменений» и «Время восстановления сервиса» она называется «Медиана», на графиках «Частота деплоя» и «Частота неуспешных изменений» — «Среднее». При наведении на точку показываются дата и значение. День, в который у метрики нет значения, точки не имеет.

График без значения ни за один день периода показывает «Нет доступных данных» и «Создайте production-деплои и инциденты, чтобы заполнить этот график.» Если значения не удалось загрузить, график показывает «Не удалось загрузить DORA-метрики.»

Период

Список в правом верхнем углу страницы задаёт период: «Последние 7 дней», «Последние 30 дней», «Последние 90 дней», «Последние 180 дней» или «Произвольный диапазон», который открывает поля выбора даты «От» и «В». Отчёт открывается на «Последние 180 дней». Готовый период заканчивается текущим днём. Произвольный период применяется, когда выбраны обе даты; он заканчивается текущим днём или раньше и ограничен 180 днями. Значок рядом с полями показывает «Даты и время отображаются в часовом поясе UTC. Диапазон дат ограничен 180 днями.»

Когда ни у одной метрики нет значения ни за один день периода, вместо карточек и графиков страница показывает «DORA-метрики недоступны» и «Создайте production-деплои и инциденты, чтобы заполнить этот дашборд.»

Ветки, учитываемые во времени проведения изменений

По умолчанию время проведения изменений развёртывания учитывает все связанные с ним запросы на слияние. В проекте, который продвигает изменения через несколько веток — запрос на слияние в `staging`, затем запрос на слияние из `staging` в `main` и развёртывание `main`, — с развёртыванием связаны оба запроса, и одно изменение считается дважды. DORA-конфигурация проекта называет целевые ветки, запросы на слияние в которые учитываются: с одной веткой `main` слияния в `staging` во время проведения изменений не попадают.

Список действует на окружения уровня `production`; окружение другого уровня сохраняет все связанные запросы на слияние. Ветка или тег, из которых выполнено развёртывание, со списком не сравниваются, поэтому проект, развёртывающий теги, сохраняет своё время проведения изменений. Список должен быть непустым и без пустых или повторяющихся имён; у проекта одна конфигурация.

Конфигурация задаётся из Rails-консоли. Откройте консоль для используемого типа установки:

Пакет для ОС Linux

```
sudo gitlab-rails console
```

Omnibus Docker

```
docker exec -it code gitlab-rails console
```

Helm-чарт

```
d8 k -n code exec -it deploy/code-toolbox -- gitlab-rails console -e production
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rails console -e
production
```

Затем создайте конфигурацию, указав полный путь проекта и имена веток:

```
project = Project.find_by_full_path('<GROUP>/<PROJECT>')
Dora::Configuration.create!(project: project, branches_for_lead_time_for_changes: %w[m
ain])
```

Чтобы изменить список существующей конфигурации:

```
Dora::Configuration.find_by!(project: project).update!(branches_for_lead_time_for_chan
ges: %w[main release])
```

Чтобы снова учитывать все связанные запросы на слияние, удалите конфигурацию командой `Dora::Configuration.find_by!(project: project).destroy!`.

Дни, посчитанные до создания или изменения конфигурации, сохраняют своё время проведения изменений. Чтобы пересчитать их, загрузите историю за затронутый период, как описано в разделе Загрузка истории.

REST API

Значения отчёта доступны через REST API, по одной метрике на запрос:

```
GET /api/v4/projects/:id/dora/metrics
GET /api/v4/groups/:id/dora/metrics
```

`:id` — идентификатор или URL-кодированный полный путь проекта или группы. Запрос аутентифицируется токеном доступа с областью `api` или `read_api`; пользователю токена нужен тот же доступ, что и к странице.

Параметры:

Параметр	Описание
`metric`	Обязательный. Метрика: `deployment_frequency`, `lead_time_for_changes`, `time_to_restore_service` или `change_failure_rate`
`start_date`	Первый день периода в форме `YYYY-MM-DD`. По умолчанию — три месяца до текущего дня
`end_date`	Последний день периода в форме `YYYY-MM-DD`. По умолчанию — текущий день
`interval`	Интервал значений: `daily` (по умолчанию), `monthly` или `all`
`environment_tiers`	Уровни развёртывания окружений, массив: `production`, `staging`, `testing`, `development`, `other`. По умолчанию — `production`

Ответ — массив объектов с полями `date` и `value`. С `daily` и `monthly` массив содержит по объекту на каждый день или месяц периода: `date` — день или первый день месяца, `value` — `null` для дня или месяца без значений; с `all` — один объект с `date`, равным `null`. Частота развёртываний указана в развёртываниях в день, время проведения изменений и время восстановления сервиса — в секундах, частота неуспешных изменений — долей от 0 до 1. Для месяца, обрезанного периодом, частота развёртываний делится на дни месяца внутри периода.

Коды ответа:

- `200` — значения;
- `400` — `start_date` позже `end_date`, период длиннее 180 дней или значение вне перечисленных выше списков;
- `401` — нет токена;
- `403` — у пользователя токена нет доступа к отчёту;
- `404` — проект или группа не найдены.

Пример:

```
curl --header "PRIVATE-TOKEN: <TOKEN>" \
  "https://<HOST>/api/v4/projects/<GROUP>%2F<PROJECT>/dora/metrics?
metric=deployment_frequency&start_date=2026-03-09&end_date=2026-03-11"
```

Ответ для проекта с 2 успешными развёртываниями в окружение уровня `production` 2026-03-10:

```
[
  { "date": "2026-03-09", "value": null },
  { "date": "2026-03-10", "value": 2.0 },
  { "date": "2026-03-11", "value": null }
]
```

Загрузка истории

Значения записываются в момент развёртывания или инцидента. Чтобы учесть развёртывания и инциденты, произошедшие до обновления, которое принесло отчёт, или пересчитать дни после изменения списка веток, администратор запускает rake-задачу `gitlab:dora:backfill` с полным путём проекта и первым и последним днём периода в форме `YYYY-MM-DD`:

Пакет для ОС Linux

```
sudo gitlab-rake 'gitlab:dora:backfill[<GROUP>/<PROJECT>,<START_DATE>,<END_DATE>]'
```

Omnibus Docker

```
docker exec code gitlab-rake 'gitlab:dora:backfill[<GROUP>/
<PROJECT>,<START_DATE>,<END_DATE>]'
```

Helm-чарт

```
d8 k -n code exec deploy/code-toolbox -- gitlab-rake 'gitlab:dora:backfill[<GROUP>
/<PROJECT>,<START_DATE>,<END_DATE>]'
```

Модуль Deckhouse Kubernetes Platform

```
d8 k -n d8-code exec -it -c toolbox deploy/toolbox -- gitlab-rake 'gitlab:dora:bac
kfill[<GROUP>/<PROJECT>,<START_DATE>,<END_DATE>]'
```

Задача принимает один проект; для отчёта группы запустите её для каждого проекта группы. Она ставит в очередь пересчёт каждого дня периода, в который завершилось успешное развёртывание, в окружении этого развёртывания, и каждого дня, в который был создан или

закрыт инцидент. В проекте без окружения уровня `production` дни инцидентов в очередь не попадают. Задача выводит `Enqueued N DORA day recounts.` и завершается; пересчёты выполняются в фоне через Sidekiq, и отчёт заполняется по мере их завершения. День пересчитывается заново, поэтому повторный запуск за тот же период даёт те же значения.

Задача останавливается с ошибкой, если аргумент пропущен (`project, start_date and end_date are required`), проект не найден (`Project not found: <PATH>`), дата указана в другой форме (`invalid date`) или первый день позже последнего (`start_date must not be after end_date`).

Безопасность

Настройки безопасности группы и проекта

Владельцы групп и мейнтейнеры проектов применяют эти настройки для повседневной защиты проектов. Откройте страницу «Настройки» группы или проекта в боковом меню слева, а затем вкладку, указанную в каждом разделе ниже.

Настройки уровня группы

Настройки, которые владелец группы применяет ко всем проектам группы.

Основные

Уровень видимости группы задаётся в разделе «Настройки» → «Основные». При уровне «Приватный» группа и её проекты видны только её участникам.

Права доступа и возможности группы

Настройте права доступа и возможности группы в разделе «Настройки» → «Основные»:

- Могут ли участники приглашать группы за пределами этой группы и её подгрупп.
- Можно ли передавать проекты этой группы другим группам.
- Уровень доступа к wiki группы.
- Протоколы доступа к Git (SSH, HTTP(S)), включённые для группы.
- Минимальная роль, необходимая для создания проектов в группе.
- Минимальная роль, необходимая для создания подгрупп.
- Могут ли участники создавать токены доступа группы и проекта в этой группе. Токен доступа группы может создать только участник с ролью `Owner` ; токен доступа проекта, созданный участником группы, ограничен ролью этого участника в проекте.
- Обязательна ли двухфакторная аутентификация для каждого участника группы, и какова продолжительность периода до её принудительного применения. После включения этой настройки подгруппы не могут задавать собственные правила двухфакторной аутентификации.
- Ограничен ли доступ к содержимому Pages участниками проекта для всех проектов группы.

Репозиторий

Настройки, которые применяются к Git-репозиторию каждого проекта группы.

Токены деплоя

Токены деплоя предоставляют доступ к репозиториям и пакетам группы без привязки к личной учётной записи.

Ветка по умолчанию

Настройте защиту ветки по умолчанию для всех проектов группы в разделе «Настройки» → «Репозиторий». Проект наследует эту настройку от группы, если группа не разрешила её переопределять.

Правила отправки изменений

Настройте правила отправки изменений для всех проектов группы в разделе «Настройки» → «Репозиторий». Если переопределение правил уровня инстанса разрешено, переопределите их здесь для политики, специфичной для группы. Список доступных правил — в разделе Правила отправки изменений.

CI/CD

Настройки, которые применяются к CI/CD-пайплайнам и переменным во всех проектах группы.

Пайплайны

Включите опцию «Включить формат JWT для токенов заданий CI/CD» в разделе «Настройки» → «CI/CD».

Переменные

Маскируйте переменные CI/CD, которые содержат чувствительные значения, в разделе «Настройки» → «CI/CD». Значение замаскированной переменной не отображается в логах задач и событиях аудита.

Настройки уровня проекта

Настройки, которые мейнтейнер или владелец проекта применяет к отдельному проекту.

Основные

Основные настройки проекта определяют, кто может просматривать проект и что может делать без назначенной роли.

Видимость, функции и разрешения проекта

Установите видимость проекта «Приватный», где это возможно, в разделе «Настройки» → «Основные». Включите опцию «Требуйте аутентификацию для просмотра медиафайлов», чтобы запретить неавторизованный доступ к медиафайлам, встроенным в задачи и запросы на слияние, и ограничьте доступ к объектам LFS, запросам на слияние, веткам, реестрам пакетов и

контейнеров, задачам, окружениям и релизам в соответствии с требованиями вашей организации. Скачивание из реестра пакетов по умолчанию запрещено для всех пользователей.

Репозиторий

Настройки, которые применяются к Git-репозиторию проекта.

Правила отправки изменений

Настройте правила отправки изменений для проекта в разделе «Настройки» → «Репозиторий». Если переопределение правил уровня группы разрешено, переопределите их здесь для политики, специфичной для проекта. Список доступных правил — в разделе Правила отправки изменений.

Защищённые ветки

Используйте защищённые ветки, в разделе «Настройки» → «Репозиторий», чтобы определить, какие роли могут сливать изменения в основную и стабильные ветки или отправлять изменения в них напрямую. По умолчанию отправлять изменения в защищённую ветку напрямую могут только участники с ролью `Maintainer` или `Owner`. Подробнее — в разделе Защищённые ветки.

Защищённые теги

Если теги отмечают релизы или другие значимые точки в истории проекта, ограничьте, какая роль может создавать теги по заданному шаблону, в разделе «Настройки» → «Репозиторий». Например, ограничьте создание тегов версий по шаблону `*_v*` ролью `Maintainer`.

Токены деплоя

Токены деплоя предоставляют доступ к репозиторию и пакетам проекта без привязки к личной учётной записи.

Ключи деплоя

Ключи деплоя — альтернатива токенам деплоя для предоставления доступа к репозиторию без привязки к личной учётной записи, в разделе «Настройки» → «Репозиторий». К ним применима та же проверка: область действия, описание и срок действия.

Запросы на слияние

Настройки, которые определяют, как запрос на слияние проходит ревью и сливается в основную или стабильные ветки проекта, в разделе «Настройки» → «Запросы на слияние».

Проверки слияния

- Включите опцию «Пайплайны должны завершиться успешно» и добавьте в пайплайн проекта необходимые проверки безопасности: сканирование секретов, статический и динамический анализ кода, сканирование зависимостей и образов контейнеров на уязвимости.
- Включите опцию «Все дискуссии должны быть закрыты», чтобы код, который ещё проходит ревью, не мог попасть в основную или стабильные ветки.

Назначайте ответственных за части кода с помощью файла `CODEOWNERS` и настраивайте правила апрувов запросов на слияние, чтобы задать минимальное число апруверов, запретить автору одобрять собственный запрос на слияние и ограничить, кто может изменять сами правила апрувов, например подключая файл с политикой ревью из другого проекта через механизм `include`. Подробнее — в разделах CODEOWNERS и Правила апрувов запросов на слияние.

CI/CD

Настройки, которые применяются к CI/CD-пайплайнам, переменным и токенам проекта, в разделе «Настройки» → «CI/CD».

Пайплайны

Включите опцию «Используйте отдельные кеши для защищённых ветвей». Назначьте на файл `.gitlab-ci.yml` ответственного через `CODEOWNERS`, чтобы контролировать, кто может изменять конфигурацию пайплайна, и требуйте ручного подтверждения перед запуском критичного пайплайна или задачи.

Переменные

Отмечайте переменные CI/CD с чувствительными значениями как защищённые и маскируемые и храните секреты в переменных CI/CD либо во внешнем Vault или Deckhouse Stronghold вместо репозитория. Подробнее — в разделе Интеграция с внешним Vault. Включите опцию «Показывать переменные пайплайна», чтобы значения переменных пайплайна отображались в логах задач: это упрощает диагностику сбоев пайплайна и обнаружение неожиданных значений.

Токены запуска пайплайна

Токен запуска пайплайна запускает пайплайн с правами участника, который выпустил токен. Проверяйте такие токены с той же тщательностью, что и разрешения токена задач CI/CD (ниже).

Разрешения токена задач CI/CD

Ограничьте список групп и проектов, которым разрешён доступ к данным этого проекта из их собственных пайплайнов, в разделе «Настройки» → «CI/CD», с помощью белого списка.

Пакеты и реестры

Защищайте пакеты, опубликованные в реестре пакетов проекта, с помощью механизма защищённых пакетов, в разделе «Настройки» → «Пакеты и реестры», ограничивая доступ участникам выше определённой роли. То же применимо к хранилищу образов контейнеров.

Политики выполнения пайплайна

Политика выполнения пайплайна (PEP) добавляет задачи из CI/CD-файла другого проекта в каждый пайплайн покрытых ею проектов. Политика хранится в проекте политик безопасности — том же виде проекта, где хранятся и политики выполнения сканирования, — и покрытый проект не

может убрать или изменить задачи, которые добавляет политика, ни из своего `.gitlab-ci.yml`, ни из настроек CI/CD.

Политика выполнения сканирования или политика выполнения пайплайна?

	Политика выполнения сканирования	Политика выполнения пайплайна
Что выполняет	Один из встроенных сканеров (SAST, Trivy, поиск секретов, PT AI, CodeScoring), каждый со своими настройками	Любую задачу, написанную обычным синтаксисом GitLab CI в выбранном вами файле
Кто определяет задачи	Собственные шаблоны сканеров продукта; политика лишь заполняет их настройки	Вы — `content.include` политики указывает на CI/CD-файл, который вы сами ведёте
Типичное применение	Обязать каждый покрытый проект использовать штатные сканеры	Добавить задачи комплаенса, аудита или уведомлений, которые проект не может пропустить, либо полностью заменить конфигурацию пайплайна проекта

Предварительные условия

Прежде чем создавать политику выполнения пайплайна, убедитесь, что:

- администратор включил для инстанса флаги функций `fe_security_scan_policies` и `fe_pipeline_execution_policies` так, как описано на странице Ранний доступ;
- у инстанса есть лицензия Enterprise Edition;
- к проекту или к группе выше него привязан проект политик безопасности. Если проект не привязан, он создаётся и привязывается при сохранении первой политики, поэтому этот шаг можно пропустить (см. «Где хранятся политики»);
- у вас есть роль `Maintainer` или выше в проекте либо роль `Owner` в группе. Эта роль нужна, чтобы создавать, изменять и тестово запускать политики, а также чтобы привязать или изменить проект политик безопасности. Для группы откройте «Настройки» → «Политика безопасности». У проекта такого раздела нет: перейдите в «Безопасность» → «Политики» и нажмите «Редактировать проект политик».

Включение функциональности

Оба флага по умолчанию выключены. Администратор включает их для инстанса так, как описано на странице Ранний доступ.

Пока хотя бы один флаг выключен или у инстанса нет лицензии Enterprise Edition, карточки «Политика выполнения пайплайна» и «Политика выполнения пайплайна по расписанию» отсутствуют на экране выбора типа политики, список политик не показывает ни одной политики выполнения пайплайна, ни один пайплайн не получает задач политики, политики по расписанию не запускаются, а форма «Запустить новый пайплайн» не заполняется переменными политик. Настройки политик в настройках проекта и в области администратора тоже скрыты. Политика `pipeline_execution_policy` или `pipeline_execution_schedule_policy`, уже записанная в документ, остаётся валидной и просто игнорируется, пока оба флага не включат.

Где хранятся политики

Политика выполнения пайплайна, как и политика выполнения сканирования, хранится в файле `.deckhouse_code/security-policies/policy.yml` проекта политик безопасности, в списке `pipeline_execution_policy` (или `pipeline_execution_schedule_policy` для политики по расписанию, см. ниже).

Чтобы открыть редактор политики, перейдите в «Безопасность» → «Политики» проекта или группы и выберите «Новая политика», затем карточку «Политика выполнения пайплайна» — либо карточку «Политика выполнения пайплайна по расписанию», чтобы вместо этого создать политику, которая запускается по собственному расписанию (см. «Политики по расписанию» ниже). Обе предлагают форму («Режим правил») и представление YAML («Режим .yaml»), между которыми можно переключаться. Когда вы заполните проект, путь к файлу и ref включаемого CI/CD-файла, редактор проверит, существуют ли этот проект, ref и файл, используя ваши собственные права на чтение, и подсветит поле, если проверка не прошла, — сохранению политики это не мешает, кроме случая, когда пустым остаётся проект или файл.

Чтобы сохранить политику, нажмите «Настройка с запросом на слияние». Сохранение политики, как и её удаление, открывает merge request в проекте политик безопасности. Политика начинает действовать только после слияния этого merge request в основную ветку проекта политик безопасности. Если проект политик безопасности ещё не привязан, при сохранении он сначала создаётся и привязывается.

Список «Политики» показывает политики выполнения пайплайна вместе с политиками выполнения сканирования; в каждой строке указан её тип, а для политики по расписанию — ещё и ближайший запланированный запуск (см. «Политики по расписанию» ниже).

Политика добавляет задачи только в пайплайны с источником CI: отправка изменений, веб-интерфейс, расписание, API, триггер, команда чата, внешний сервис, пайплайн другого проекта, событие merge request и событие внешнего pull request. Дочерние и внутренние пайплайны (Web IDE, DAST по запросу, пайплайны политик по расписанию и другие пайплайны, которые продукт создаёт сам) не получают задач политики, как и проверка в CI Lint. Запуск по расписанию политики выполнения сканирования запускает только её сканеры и тоже не получает задач политики.

Проект политик безопасности не покрывается политиками, которые хранит сам. Политики, хранящиеся в других проектах политик безопасности, покрывают его как обычно.

Синтаксис

```
pipeline_execution_policy:
  - name: Compliance jobs                    # обязательное, 1-255 символов
    description: Runs the compliance check in every pipeline
    enabled: true                            # обязательное
    pipeline_config_strategy: inject_policy  # обязательное: inject_policy или
override_project_ci
    suffix: on_conflict                      # необязательное: on_conflict (по
умолчанию) или never
    skip_ci:                                 # необязательное
      allowed: false
# обязательное в этом ключе; false: [skip ci], [ci skip] и ci.skip не пропускают эту
политику
      allowlist:
        users:
          - id: 42                           # этот пользователь всё равно может
пропустить
    no_pipeline:                             # необязательное, той же формы, что
skip_ci; для опции ci.no_pipeline
      allowed: false
    content:
      include:
        - project: compliance/ci
          file: policy-ci.yml
          ref: main                          # необязательное, иначе ветка проекта по
умолчанию
    variables_override:                      # необязательное
      allowed: false                         # проходят только имена из exceptions;
true позволяет переменным пользователей перекрывать переменные политики
      exceptions:
        - DEPLOY_ENVIRONMENT
      dotenv: respect_policy                 # необязательное: respect_policy (по
умолчанию) или allow_override
    policy_scope: {}                         # необязательное, то же определение, что
у политики выполнения сканирования
    metadata:                                # необязательное, произвольные строковые
аннотации
      owner: compliance-team
```

В списке `pipeline_execution_policy` одного документа допустимо не более 5 записей по умолчанию (администратор может задать от 1 до 20, см. Ограничения), а к одному пайплайну в сумме применяется столько же политик выполнения пайплайна — считая вместе проект и его

группы, старшие уровни в первую очередь. Более младшая политика не может вытеснить более старшую.

Справочник полей

Поле	Описание
`name`	Обязательное. Имя политики, 1-255 символов.
`description`	Произвольный текст, описывающий назначение политики.
`enabled`	Обязательное. Активна ли политика.
`content.include`	Обязательное. Ровно одна запись: `project` (обязательно, проект, из которого читается CI/CD-файл), `file` (обязательно, путь к нему), `ref` (необязательно, ветка, тег или коммит; иначе ветка проекта по умолчанию).
`pipeline_config_strategy`	Обязательное. `inject_policy` или `override_project_ci` (см. «Стратегии» ниже).
`suffix`	`on_conflict` (по умолчанию) или `never` — что происходит, когда имя задачи этой политики совпадает с именем другой задачи (см. «Конфликты имён задач и `suffix`»).
`skip_ci`	Можно ли пропустить задачи этой политики директивами `[skip ci]`, `[ci skip]` и `ci.skip`: `allowed` (обязательно, если ключ задан) и список идентификаторов пользователей `allowlist.users`, которым пропуск всё же разрешён. Без ключа политика запрещает пропуск, как при `allowed: false` (см. «`[skip ci]` и отправка изменений без пайплайна»).
`no_pipeline`	Той же формы, что `skip_ci`, — для опции отправки `ci.no_pipeline` (`git push -o ci.no_pipeline`).
`variables_override`	Какие переменные, заданные пользователем, попадают в задачи политики: `allowed` (обязательно внутри этого ключа), `exceptions` (до 100 имён по 255 символов) и `dotenv` (`respect_policy`, по умолчанию, или `allow_override`). При `allowed: true`

Поле	Описание
	переменные, заданные пользователем, перекрывают собственные переменные политики (см. «Переменные»).
`policy_scope`	Какие проекты и группы покрывает политика: `projects.including` / `excluding` и `groups.including` / `excluding`, каждый — список `{id: <ID>}`. То же определение, что у политики выполнения сканирования. Пустой список `including` (`including: []`) означает то же, что отсутствие `including`: все проекты.
`metadata`	Произвольные строковые пары ключ-значение; самим движком политики не читаются.

Стратегии

`pipeline_config_strategy` определяет, что происходит с собственной конфигурацией CI/CD покрытого проекта:

- **`inject_policy`** — собственный `.gitlab-ci.yml` проекта читается и обрабатывается как обычно, а задачи политики встраиваются в него: стадия, нужная политике, которой ещё нет у проекта, вставляется прямо перед ближайшей следующей стадией из порядка самой политики, которая у проекта уже есть, а если такой нет — перед `.post`. Политика без `stages:` использует порядок по умолчанию `build`, `test`, `deploy`, поэтому она тоже падает в проекте, порядок стадий которого ему противоречит, например `stages: [test, build, deploy]`, даже если её единственная задача находится в `test`. Если порядок стадий политики противоречит порядку проекта (например, у проекта `build` идёт раньше `deploy`, а у политики — наоборот), вместо применения политика получает собственную падающую задачу (см. «Ошибки»). Сообщение об ошибке начинается словами «Cyclic dependencies detected when enforcing policies.» и затем называет стадии политики и пайплайна.

  Политика, все задачи которой находятся только в `.pre` и `.post`, не добавляет задач, как и любой пайплайн, в котором есть только эти стадии. Чтобы такие задачи выполнялись, добавьте политике хотя бы одну задачу в другой стадии, например в `.pipeline-policy-pre`.

- **`override_project_ci`** — собственный `.gitlab-ci.yml` проекта не читается вовсе: проект получает пустую конфигурацию, и пайплайн наполняют только стадии `.pre` и `.post`, применяемые политики (как `override_project_ci`, так и `inject_policy`) и обязательные сканирования. Имя пайплайна берётся из `workflow:name` самой младшей политики с `override_project_ci`, у которой оно задано. Если политика отфильтрована собственными

правилами (её `workflow:rules` или `rules` всех её задач), она всё равно заменяет CI проекта: `.gitlab-ci.yml` проекта не читается, а политика не добавляет задач. Если задач не добавляет и никакая другая политика, пайплайн не создаётся. Так `workflow:rules` политики решают за проект, например останавливают пайплайны в ветках. Чтобы вернуть собственные задачи проекта, укажите сам проект в `content.include` другой политики (например, `project: $CI_PROJECT_PATH`, `ref: $CI_COMMIT_SHA`) — тогда они станут задачами этой политики, внутри её стадий.

Политики в пайплайнах merge request

Задачи политики подчиняются обычным правилам CI. Если в CI/CD-файле политики нет `workflow:rules`, а у её задач нет `rules`, задачи получают значение по умолчанию `only: [branches, tags]`, поэтому в пайплайнах merge request они не появляются, и там выполняются только собственные задачи проекта. Чтобы политика выполнялась и в пайплайнах merge request, задайте в её CI/CD-файле `workflow:rules` или `rules` у задач так, чтобы они срабатывали в пайплайнах merge request. Например:

```
workflow:
  rules:
    - when: always

policy-check:
  stage: .pipeline-policy-pre
  script:
    - ./check.sh
```

Здесь `when: always` заставляет политику выполняться в пайплайне с любым источником. Чтобы ограничить её пайплайнами merge request, используйте вместо этого более узкое правило, например `if: $CI_PIPELINE_SOURCE == "merge_request_event"`.

Зарезервированные стадии и блокировка

Пайплайн, к которому применяется хотя бы одна политика выполнения пайплайна — пусть даже только с падающей задачей ошибки, — получает две дополнительные стадии: `.pipeline-policy-pre` первой и `.pipeline-policy-post` последней. Задача проекта, помещённая прямо в одну из этих стадий, отклоняется — так же, как задача в необъявленной стадии. Конфигурация пайплайна при этом признаётся недействительной с ошибкой «`<job> job: chosen stage <stage> is reserved for pipeline execution policies`». Проект может объявить `.pipeline-policy-pre` или `.pipeline-policy-post` в собственном `.gitlab-ci.yml`, пока к нему не применяется ни одна политика: тогда это обычные стадии, и блокировка не действует. Блокировка включается, только когда в стадии есть задачи политики, в том числе задача ошибки политики.

Каждая другая задача пайплайна ждёт завершения `.pipeline-policy-pre`, включая задачу с `needs: []`. Как только `.pipeline-policy-pre` завершается успешно (или с ошибкой при `allow_failure: true`), остальной пайплайн выполняется обычным образом. Если стадия падает

или отменяется, все остальные задачи пропускаются — включая задачи с `when: always` и `when: on_failure`, — и пайплайн падает: задача проекта не может обойти упавшую блокировку, ожидая её с `when: always`. Блокировка так же останавливает остальной пайплайн, когда все её задачи политики пропущены, например когда единственная задача политики в ней имеет `when: on_failure`.

Ошибки

Политика, которую не удалось собрать — её CI/CD-файл или проект отсутствует либо недоступен, её CI невалиден, либо сборка упала по любой другой причине, — не останавливает создание пайплайна. Вместо этого задачи такой политики заменяются одной падающей задачей `<slug политики>:policy_error`, помещённой в `.pipeline-policy-pre`; при совпадении имени она становится `<slug политики>-p<ID проекта политик>-<индекс>:policy_error`. Если и это имя занято, добавляется счётчик: `<slug политики>-p<ID проекта политик>-<индекс>-2:policy_error`, затем `-3` и так далее. Slug строится из латинских букв и цифр имени политики; остальные символы, в том числе кириллица, отбрасываются. Если ничего не остаётся, slug равен `policy-<индекс>`, где индекс — позиция политики в документе, начиная с 0. В логе задачи названы политика, её проект политик безопасности и сама ошибка, после чего задача падает. На остальные применимые политики и на собственные задачи проекта это не влияет.

У задачи ошибки нет `tags` и `image`: она не берёт их из `default:` и переменных проекта, поэтому проект не может сделать так, чтобы она прошла. Ей нужен раннер, который принимает задачи без тегов и запускает POSIX-оболочку, например shell executor или Docker executor, в образе по умолчанию которого есть `sh`. Без такого раннера задача остаётся в ожидании, и пайплайн блокируется, а не проходит.

Конфликты имён задач и `suffix`

Имена задач сравниваются без учёта маркера `parallel` (`job 1/2`, `job: [a]`). Имена задач проекта всегда считаются занятыми. Все политики обрабатываются за один проход в порядке слияния: собственные задачи проекта, политики проекта, затем политики его групп от ближайшей группы вверх, группа верхнего уровня последней. Внутри одного файла политик порядок совпадает с порядком в документе. Задача, которая первой заняла имя, сохраняет его независимо от значения `suffix`.

- **`suffix: on_conflict`** (по умолчанию) — задача, имя которой уже занято проектом или более ранней политикой в порядке слияния, переименовывается в `<имя>:policy-<ID проекта политик>-<индекс>`, маркер `parallel` остаётся в конце. Если и переименованное имя занято задачей проекта или другой политики, добавляется счётчик: `<имя>:policy-<ID проекта политик>-<индекс>-2`, затем `-3` и так далее. Ссылки на переименованную задачу внутри той же политики следуют за новым именем: `needs` и `dependencies` других задач политики, а также `job` в записи `trigger:include:artifact` задачи-триггера политики.
- **`suffix: never`** — имя задачи не меняется. Если любое имя этой политики уже занято задачей проекта или более ранней политикой в порядке слияния, эта политика вместо

применения получает падающую задачу ошибки (см. «Ошибки»). Ошибка называет только задачу, чьё имя занято, а не другую политику: «The job name `X` is already used in the pipeline, and the policy does not allow a suffix (`suffix: never`).»

Например, у группы, её подгруппы и проекта есть по политике с задачей `dup`, а в собственном пайплайне проекта такой задачи нет. Политика проекта сохраняет имя `dup`, а задачи политик подгруппы и группы получают имена `dup:policy-<ID>-<индекс>`.

Порядок политик внутри одного документа имеет значение. В документе группы с политиками `[A (on_conflict), B (never)]`, у обеих из которых есть задача `dup`, A сохраняет `dup`, а B падает с задачей ошибки. При обратном порядке `[B (never), A (on_conflict)]` B сохраняет `dup`, а задача A получает суффикс.

Используйте `suffix: never`, только если на точное имя задачи ссылается что-то вне политики — например, `needs` другой задачи.

`[skip ci]` и отправка изменений без пайплайна

По умолчанию политика не позволяет пропустить свои задачи. Коммит, отправленный с `[skip ci]` или `[ci skip]` в сообщении либо с опцией отправки `ci.skip`, обычно вообще не создаёт пайплайн. Если какая-либо применимая политика запрещает пропуск, пайплайн всё равно создаётся полностью — вместе идут собственные задачи проекта, обязательные сканирования и задачи политики. Политика запрещает пропуск, когда её `skip_ci.allowed` равно `false` либо ключ не задан, а отправивший изменения пользователь не входит в её `allowlist.users`.

Ключ `no_pipeline` работает так же, но только для опции отправки `ci.no_pipeline` (`git push -o ci.no_pipeline`). Отправка без этой опции на него не влияет.

Проект без конфигурации CI/CD — отдельный случай. Политика выполнения пайплайна, которая к нему применяется, всё равно создаёт пайплайн с задачами политики, и никакая настройка для этого не нужна.

Политика, чей пайплайн отфильтрован её собственными правилами (её `workflow:rules` или `rules` всех её задач), ничего не выполняет, поэтому ничего не запрещает.

Укажите `allowed: true` либо перечислите пользователей в `allowlist.users` (по `id`), чтобы разрешить пропуск этой политики как обычно. Когда каждая применимая политика выполнения пайплайна разрешает пропуск, а отказывают только настройки самой политики выполнения сканирования, пайплайн сохраняет только обязательные сканирования, без задач этой политики.

Переменные

Внутри собственного CI/CD-файла политики переменные задачи по умолчанию побеждают: над любой переменной CI/CD, которую определяет пользователь (переменные инстанса, группы и проекта, а также переменные ручного запуска пайплайна, расписания, триггера и ручной задачи), и над dotenv-переменными из собственных зависимостей задачи, если не задано `dotenv:`

`allow_override` (см. ниже). Исключение — `variables_override` с `allowed: true`, о нём ниже. Предопределённые переменные `CI_*` это не затрагивает.

`variables_override` определяет, какие пользовательские переменные вообще попадают в задачу политики, прежде чем поверх них добавляются переменные самой задачи. То же правило действует на пользовательские переменные, которые могут читать `rules` задач политики, её `workflow:rules` и `include:rules`:

`variables_override`	Что попадает в задачу	При совпадении имени с переменной политики
Не задано (по умолчанию для `inject_policy` и `override_project_ci`)	Все переменные, заданные пользователем	Побеждает переменная политики
`allowed: false`	Только имена, перечисленные в `exceptions`	Побеждает переменная политики, в том числе над исключением
`allowed: true`	Все переменные, заданные пользователем, кроме перечисленных в `exceptions`	Побеждает переменная, заданная пользователем

При `allowed: true` переменные, которые определяют пользователи и которых нет в `exceptions`, перекрывают собственные переменные политики с теми же именами. При `allowed: false` проходят только имена из `exceptions`, а переменные политики по-прежнему побеждают. `rules` задач политики видят те же переменные, что и сама задача.

Для политики по расписанию (`pipeline_execution_schedule_policy`) отсутствующий `variables_override`, напротив, по умолчанию означает `allowed: false`: пайплайн по расписанию создаёт бот политик безопасности, а не участник проекта, поэтому ни одна пользовательская переменная не попадает в его задачи, если её имя не указано в `exceptions`.

`dotenv` определяет, как обрабатываются dotenv-переменные из собственных зависимостей задачи политики: `respect_policy` (по умолчанию) пропускает их через то же правило, что и пользовательские переменные; `allow_override` передаёт их без фильтрации, и тогда они тоже перевешивают переменные самой политики. Без `allow_override` dotenv-переменные не перекрывают переменные политики, даже при `allowed: true`. Это относится и к dotenv-переменным из других задач того же пайплайна политики, переданным через `dependencies` или `needs`.

Без `variables_override` пользователь может отключить политику переменной, заданной при запуске пайплайна, если `workflow:rules` политики (или `rules` задачи) зависят от этой

переменной. Чтобы закрыть эту возможность, задайте `variables_override: {allowed: false}` и укажите в `exceptions` только те имена, которые политике действительно нужны.

Переменные `variables:` верхнего уровня и `default:` собственного `.gitlab-ci.yml` покрытого проекта не попадают в задачи политики ни при каком значении `variables_override`: пайплайн политики собирается только из её собственного содержимого CI/CD.

На странице «Запустить новый пайплайн» форма также заполняется заранее переменными, которые CI/CD-файлы политик, применяемых к проекту, объявляют с `description` в `variables:` верхнего уровня. Подставляются только те переменные, которые значение из формы действительно может изменить: у политики задан `variables_override` с `allowed: true`, а имя не указано в `exceptions`. Без `variables_override` или при `allowed: false` побеждают собственные переменные политики, поэтому ни одна из них не подставляется.

Переменная собственного `.gitlab-ci.yml` проекта с тем же именем сохраняет значение и описание проекта. Форма отправляет все подставленные значения во весь пайплайн, поэтому переменная не подставляется, если политики, значения которых форма может переопределить, задают ей разные значения, в корне или в `variables:` задания, в том числе в задании той же политики со значением, отличным от корневого. Переменная с одинаковым значением везде подставляется.

CI-файл политики читается с правами пользователя, который открывает страницу, как и при создании пайплайна: файл, который пользователь не может прочитать, и некорректный файл ничего не добавляют. Это работает и для проекта без `.gitlab-ci.yml`. Список кешируется так же, как подстановка собственных переменных проекта, поэтому изменение политики появится после обновления кеша.

Для задачи, которую добавила политика, GitLab передаёт раннеру имя политики и её настройки `variables_override` (`allowed` и `exceptions`), а раннер, который их поддерживает, выводит их в начале лога задачи. Остальные раннеры их игнорируют. Если у встраиваемой или заменяющей политики нет `variables_override`, передаётся только имя политики; для политики по расписанию передаётся и `allowed: false`.

Доступ к CI-файлу и безопасность

Собственный `content.include` политики читает CI/CD-файл независимо от того, есть ли у автора пайплайна доступ к проекту, только если `content.include.project` — это сам проект политик безопасности. Разрешено ли это, определяет настройка `spp_repository_pipeline_access`, которая читается из самого проекта политик безопасности и разрешается в следующем порядке — побеждает первое заданное значение:

1. заблокированное значение инстанса;
2. значение ближайшей к корню заблокированной группы;
3. собственное значение проекта политик безопасности;
4. значение ближайшей группы над проектом политик безопасности, где оно задано;
5. значение инстанса по умолчанию.

У нового проекта есть собственное значение, и оно включено. Доступ к проекту политик безопасности остаётся включённым, пока проект его не выключит или пока настройку не заблокирует инстанс либо группа. Значения группы и инстанса действуют только для проекта, у которого нет собственного значения.

Чтобы изменить значение проекта политик безопасности, откройте его «Настройки» → «Основные» → «Уровень доступа, доп. функции проекта, права», найдите раздел «Политики выполнения пайплайна» и используйте флажок «Предоставить доступ к настройкам CI/CD для проектов, связанных с этим проектом политики безопасности в качестве источника политик безопасности.» Флажок недоступен, если настройку заблокировал инстанс или группа.

Для CI/CD-файла из любого другого проекта, например `compliance/ci` из примера синтаксиса, этот доступ не действует. Пользователю, который создаёт пайплайн, нужен доступ на чтение к этому проекту, либо этот проект должен разрешить bot-доступ, как описано ниже.

CI/CD-файл политики может также подключить через `include:project` файл из проекта, отличного от её проекта политик безопасности, или получить шаблон CI/CD-компонента — без того, чтобы автор пайплайна имел туда доступ на чтение. Это работает, только если владелец этого другого проекта сам разрешил это в своём разделе «Настройки» → «Основные» → «Уровень доступа, доп. функции проекта, права», блок «Доступ политик выполнения пайплайна к файлам CI/CD», и выполняются эти условия:

- включена настройка «Разрешить доступ к файлам конфигурации CI/CD в этом проекте.» (`pipeline_execution_policy_bot_access_enabled`);
- запрошенный путь совпадает с одним из указанных там шаблонов (до 20 штук, по 255 символов каждый, `pipeline_execution_policy_bot_access_file_patterns`), например `ci/**/*.yml` . Поддерживаются подстановки `*` , `**` , `?` и `[...]` . Фигурные скобки вида `{a,b}` не раскрываются и сопоставляются буквально. Путь с сегментом `.` или `..` отклоняется;
- покрытый проект входит в указанную там группу (`pipeline_execution_policy_bot_access_group_id`), а если группа не указана — входит в корневую группу этого другого проекта. Даже если группа указана, покрытый проект должен входить в ту же корневую группу, что и этот другой проект;
- политика, которая применяется к покрытому проекту, называет этот другой проект в собственном `content.include` .

Последнее условие означает, что вложенный `include:project` или компонент из третьего проекта, который никакая политика не называет в собственном `content.include` , отклоняется.

> ⚠ Защитите основную ветку проекта политик безопасности и каждого проекта, чьи CI/CD-файлы подключает политика, так чтобы отправлять в неё изменения могли только доверенные мейнтейнеры: тот, кто может отправить CI/CD-файл политики, управляет задачами, которые выполняются во всех проектах, покрытых этой политикой, в обход их собственного ревью. Никогда не храните секреты в CI/CD-файле политики — вместо этого используйте защищённые или замаскированные переменные CI/CD.

Политики по расписанию

Политика `pipeline_execution_schedule_policy` не встраивает задачи в собственный пайплайн покрытого проекта: она создаёт свой пайплайн, собранный только из её содержимого CI/CD, от имени бота политик безопасности, по расписанию наподобие cron. У неё нет `pipeline_config_strategy`, `suffix`, `skip_ci` и `no_pipeline`, поскольку она никогда не сливается с пайплайном проекта.

Форма («Режим правил») карточки «Политика выполнения пайплайна по расписанию» раскладывает политику на разделы: «Конфигурация пайплайна» (включаемый CI/CD-файл), «Расписание», «Приостановить запланированные запуски» (настройки `snooze`), «Пользовательские переменные» (`variables_override`) и, для политики уровня группы, «Область действия политики». После того как политика сохранена, появляется ещё и раздел «Тестовый запуск» (см. «Тестовые запуски» ниже).

В разделе «Расписание» поле «Временное окно» принимает число и единицу измерения («Минуты», «Часы» или «Дней»). Значение вне допустимого диапазона сразу заменяется ближайшим допустимым, поэтому при вводе 5 минут поле показывает 10 минут. У политики без `time_window` поле пустое, с сообщением «Это поле обязательно к заполнению».

В документе допустимо не более 5 таких записей, и у каждой ровно одна запись `schedules`:

```
pipeline_execution_schedule_policy:
  - name: Nightly compliance audit
    enabled: true
    content:
      include:
        - project: compliance/ci
          file: scheduled-audit.yml
    schedules:
      - type: daily                 # daily, weekly или monthly
        start_time: '02:00'         # ЧЧ:ММ
        timezone: UTC                # необязательное, по умолчанию UTC
        time_window:                 # обязательное
          value: 3600                # обязательное, секунды, окно разброса
          distribution: random       # обязательное, допустим только random
        branches: [main, release]
# необязательное, до 5 явных имён; иначе ветка по умолчанию
        snooze:                      # необязательное
          until: '2026-11-01T00:00:00Z'
          reason: Paused during the migration
    policy_scope: {}
```

- `start_time` нужно заключать в кавычки (`start_time: "06:00"`). YAML читает `06:00` без кавычек как число, и политика становится недействительной.

- `type` задаёт каденцию: `daily` не требует ничего дополнительно; `weekly` требует `days` — список названий дней недели (`Monday`, …, `Sunday`); `monthly` требует `days_of_month` — список чисел от 1 до 31.
- `timezone` по умолчанию — `UTC`, в отличие от расписаний самой политики выполнения сканирования, которые без явной зоны берут часовой пояс сервера. Это должно быть имя часового пояса IANA, например `Europe/Moscow`; имя часового пояса Rails, например `Moscow`, тоже принимается. Неизвестный часовой пояс делает политику недействительной.
- `time_window` обязателен, в нём обязательны и `value`, и `distribution`. `distribution` допускает только `random`.
- `time_window.value` случайно распределяет запуск пайплайнов всех покрытых проектов в пределах этого числа секунд (от 600 до 2 629 746, то есть примерно от десяти минут до месяца), вместо того чтобы запускать их все одновременно; реальный разброс также ограничен временем, оставшимся до следующего запланированного запуска.
- `branches` явно называет до 5 веток — без шаблонов. Ветка, которой больше не существует, пропускается. Если ни одной из перечисленных веток в проекте нет либо список пуст или отсутствует, используется ветка проекта по умолчанию. Проект с пустым репозиторием пропускается.
- Когда часы переводят назад при переходе на зимнее время, местное время слота может наступить дважды, поэтому расписание может сработать дважды за этот день.
- `snooze.until` (дата и время в формате ISO 8601) и необязательный `snooze.reason` (до 256 символов) пропускают все запуски, которые должны были произойти до этого момента, не отключая саму политику; первый запуск после `until` происходит как обычно.

Когда политика изменяется, отключается, удаляется, либо покрытый проект выходит из `policy_scope`, каждый уже запущенный ею пайплайн, который ещё не завершился, отменяется. То же происходит, когда проект политик безопасности проекта или группы отвязывают или заменяют другим: незавершённые пайплайны, запущенные его политиками по расписанию для этого проекта или группы, отменяются.

Изменения расписания, приостановка и отмена уже запущенных пайплайнов вступают в силу при следующей синхронизации. Синхронизация выполняется каждые 5 минут. При отвязке проекта политик безопасности запущенные пайплайны отменяются сразу.

Для политики по расписанию `variables_override` по умолчанию означает `allowed: false` (см. «Переменные» выше): переменные, нужные запуску по расписанию, перечисляйте в `exceptions` явно. В разделе формы «Пользовательские переменные» выбор «Разрешить все пользовательские переменные» не просто оставляет поле пустым, как для встраиваемой или заменяющей политики: поскольку отсутствующий `variables_override` здесь запрещает все переменные, этот выбор вместо этого записывает в политику `variables_override: {allowed: true}`.

Список «Политики» показывает для политики по расписанию столбец «Следующий запуск».

Тестовые запуски

Откройте уже сохранённую политику по расписанию в редакторе, чтобы увидеть её раздел «Тестовый запуск» — у ещё ни разу не сохранённой политики тестировать пока нечего. Нажмите кнопку «Тестовый запуск», чтобы открыть диалог, выберите до 5 проектов (у политики уровня проекта выбирать не из чего — запускать можно только на её собственном проекте) и нажмите «Запустить»: это сразу же, не дожидаясь расписания, запускает пайплайн политики на каждом выбранном проекте. Включать политику для тестового запуска не обязательно.

Тестовый запуск отличается от запуска по расписанию так:

- Он выполняется на ветке по умолчанию каждого выбранного проекта. Список `branches` игнорируется.
- Он использует политику в том виде, в каком она сохранена в основной ветке проекта политик безопасности, а не версию, открытую в редакторе. Сначала сохраните политику и слейте её merge request.
- Приостановка (`snooze`) игнорируется.
- Каждый выбранный проект должен входить в область действия политики, и у вас должно быть право создавать там пайплайны. Иначе ничего не запускается, и вы получаете ошибку «Эту политику нельзя тестово запустить в проектах: <пути проектов>.»

Под кнопкой тот же раздел показывает последние 20 тестовых запусков политики. Столбцы: «Проект», «Статус пайплайна» (текущий статус созданного запуском пайплайна со ссылкой на него), «Запущен», «Кто запустил» и «Ошибка», где указана причина сбоя запуска. Запуск, который не завершился через сутки после старта, получает статус «Устарело», а его пайплайн отменяется, если ещё активен. Тестовые запуски старше 30 дней удаляются.

Список показывает только запуски в проектах, которые вы можете читать. Ссылка на пайплайн и сообщение об ошибке запуска показываются, только если вы можете просматривать пайплайны его проекта; иначе в столбце «Статус пайплайна» показывается собственный статус запуска («Создано», «Выполняется», «Завершено», «Неудачно» или «Устарело»), а в столбце «Ошибка» — прочерк.

Ограничения

Ограничение	Значение
Записей `pipeline_execution_policy` в документе	5 по умолчанию, настраивается от 1 до 20
Записей `pipeline_execution_schedule_policy` в документе	5
Политик выполнения пайплайна, применяемых к одному пайплайну (проект и его группы вместе)	5 по умолчанию, настраивается от 1 до 20
Записей `content.include`	Ровно 1
Длина `name`	1-255 символов
Записей `variables_override.exceptions`	100, по 255 символов каждая
Записей `schedules` на политику по расписанию	Ровно 1
`time_window.value`	От 600 до 2 629 746 секунд
Записей `branches` (политика по расписанию)	5
Значения `days_of_month`	От 1 до 31
Длина `snooze.reason`	До 256 символов
Шаблонов bot-доступа на проект	20, по 255 символов каждый
Проектов на тестовый запуск	5
Тестовых запусков в списке	Последние 20 на политику, не старше 30 дней

Изменение лимита политик выполнения пайплайна

Лимит может изменить администратор:

1. В области администратора перейдите в «Настройки» → «CI/CD».
2. Разверните раздел «Политики выполнения пайплайна».
3. Задайте «Максимум политик выполнения пайплайна».
4. Нажмите «Сохранить изменения».

Лимит можно задать и через REST API. Отправьте `PUT /api/v4/application/settings` с параметром `fe_application_settings_attributes[pipeline_execution_policy_limit]`, значение от 1 до 20:

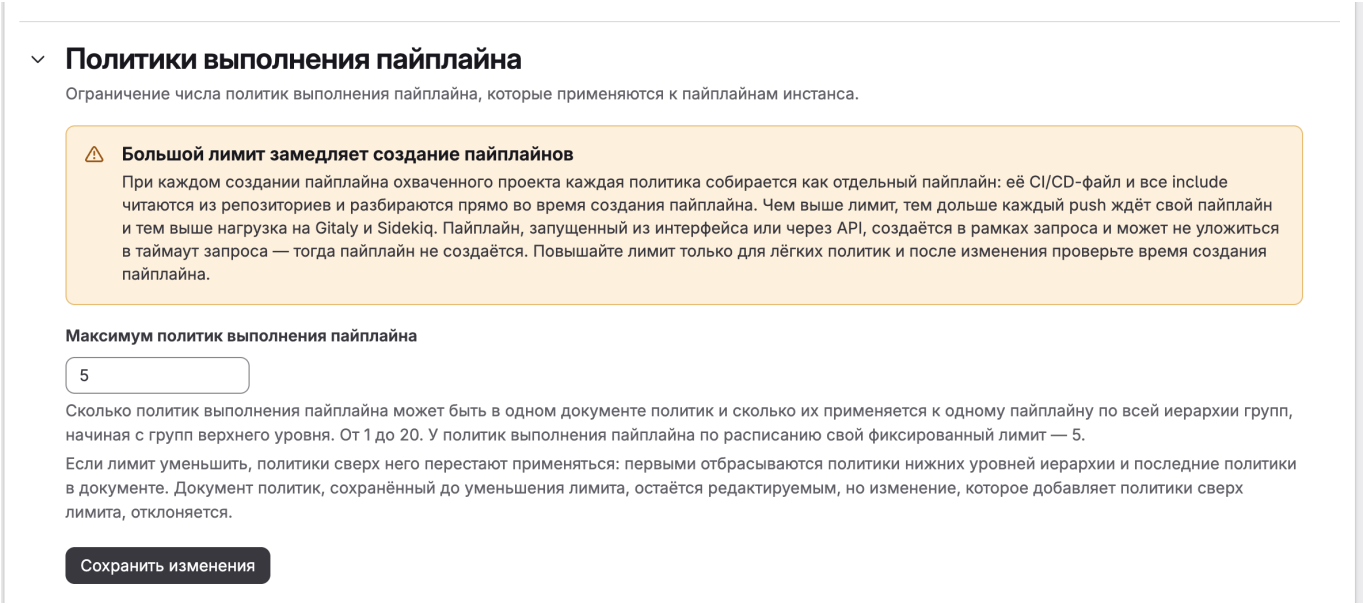
```
curl --request PUT --header "PRIVATE-TOKEN: $ADMIN_TOKEN" \
  --header "Content-Type: application/json" \
  --data '{"fe_application_settings_attributes": {"pipeline_execution_policy_limit":
10}}' \
  "<INSTANCE_URL>/api/v4/application/settings"
```

`GET /api/v4/application/settings` возвращает текущее значение в `fe_application_settings_attributes`.

Раздел виден только при включённых политиках выполнения пайплайна. Одно значение действует и на число записей в документе, и на число политик, применяемых к одному пайплайну. Лимит политик по расписанию остаётся равным 5.

> ⚠ Каждая политика строится как отдельный пайплайн при создании каждого пайплайна охваченного проекта: продукт читает её CI/CD-файл и подключаемые файлы и разбирает YAML. Чем выше лимит, тем дольше каждая отправка изменений ждёт свой пайплайн и тем выше нагрузка на Gitaly и Sidekiq. Пайплайн, запущенный из интерфейса или через API, создаётся прямо в запросе, поэтому может не уложиться в таймаут запроса и не быть создан. Повышайте лимит только для лёгких политик и после этого проверяйте время создания пайплайна.

При уменьшении лимита лишние политики перестают применяться: сначала младшие уровни, а внутри документа — последние. Документ, сохранённый до уменьшения лимита, остаётся редактируемым: политики в нём можно править и удалять, но изменение, которое добавляет политики сверх лимита, отклоняется.



SAST-сканирование с помощью Opengrep

Статический анализ кода (SAST, Static Application Security Testing) в Deckhouse Code ищет уязвимости в исходном коде проекта. За ним стоит сканер Opengrep, а само сканирование называется `opengrep_sast`.

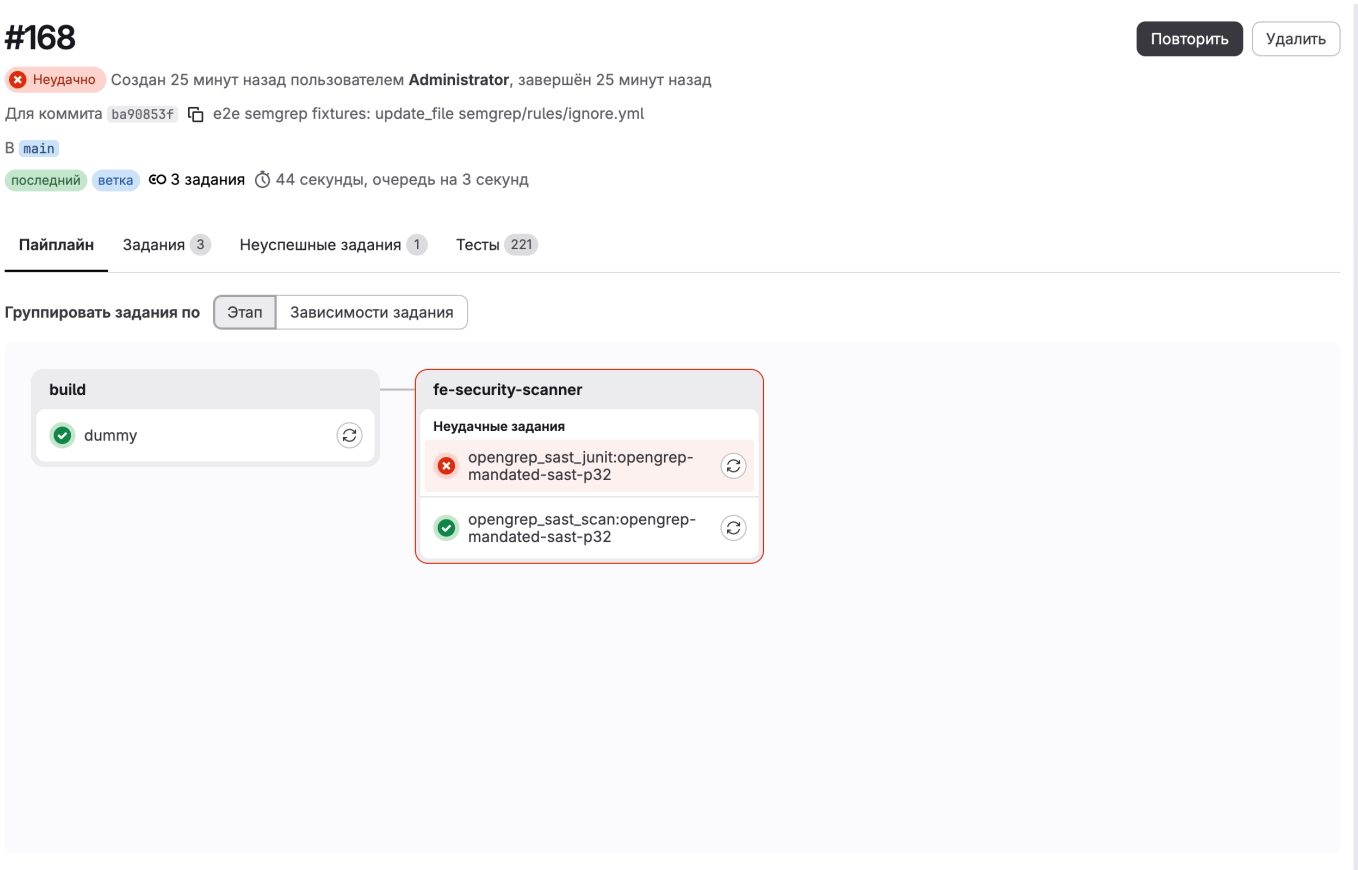
Сканирование обязательное: его подмешивает в пайплайн проекта политика исполнения сканирования, которая лежит в отдельном проекте политик безопасности. Что сканируется и что приводит к ошибке пайплайна, решает политика — то есть команда безопасности. Файл `.gitlab-ci.yml` и настройки CI/CD проверяемого проекта на это решение не влияют.

Что делает сканирование

Сканирование `opengrep_sast` добавляет в пайплайн задачи `opengrep_sast_scan` и `opengrep_sast_junit`. Этап задач называет политика, по умолчанию — `fe-security-scanner` (раздел «Этап задач сканирования» на странице Сканирование с помощью Trivy). Задачи:

Задача	Что делает
`opengrep_sast_scan`	Запускает сканер по репозиторию и публикует его собственный JSON-вывод как артефакт `opengrep-scan/`. В логе задача называет движок и образ анализатора, в котором выполнялась, путь к набору правил и число правил в нём, сколько правил выполнилось и сколько файлов прочитано.
`opengrep_sast_junit`	Читает этот JSON, строит из него отчёт для сводки тестов и отчёт безопасности и решает исход: задача завершается успешно, если ни одна находка не достигла порога блокировки, и с ошибкой в противном случае. Выполняется в образе Ruby.

В пайплайне имя каждой задачи несёт политику, которая её потребовала: `opengrep_sast_scan:<имя политики в виде слага>-p<идентификатор проекта политик>`. Групповая политика и политика проекта, обе требующие это сканирование, дают по паре задач каждая.



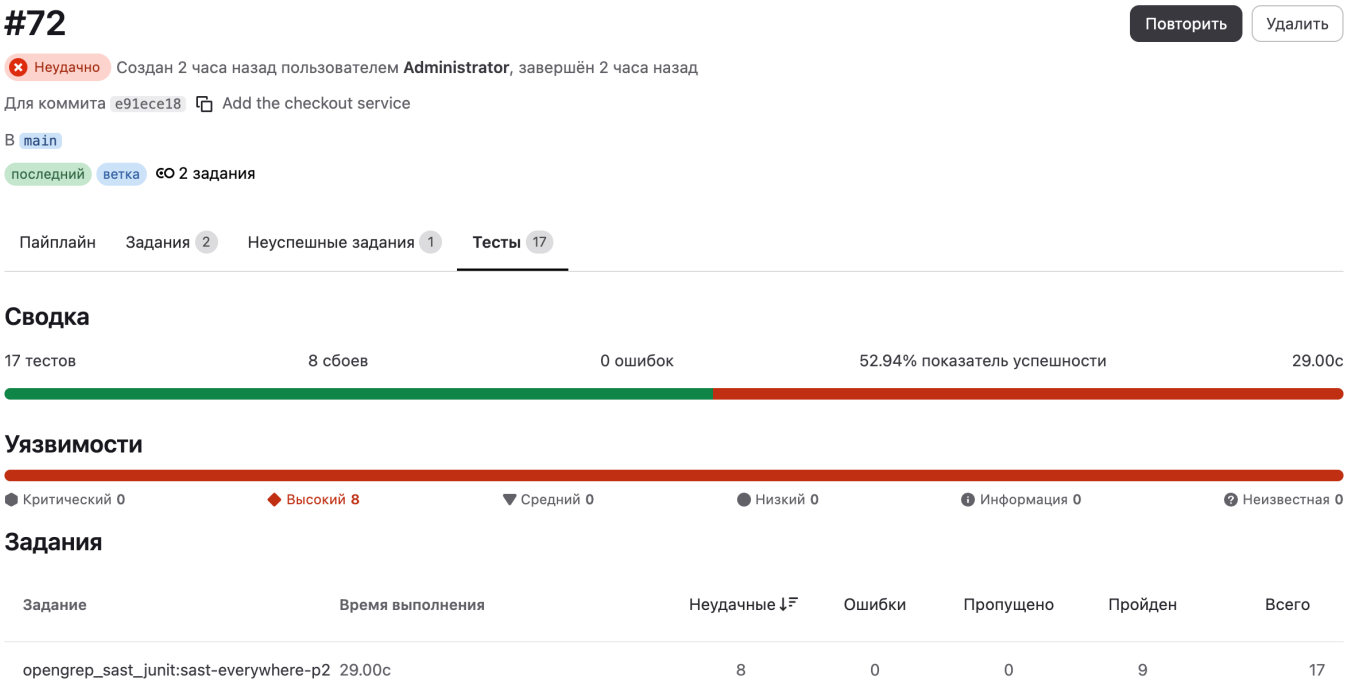
Обе задачи выгружают артефакты с `when: always`, поэтому заблокированный пайплайн сохраняет находки, которые его заблокировали.

```
2671  12:12:13           Should not use md5 to generate hashes. md5 is proven to be vulnerable through the use o
                        f brute-force
2672  12:12:13           attacks. Could also result in collisions, leading to potential collision attacks. Use S
                        HA256 or
2673  12:12:13           other hashing functions instead.
2674  12:12:13
2675  12:12:13            22¦ Digest::MD5.hexdigest(secret)
2676  12:12:13
2677  12:12:13     ❱❱ rules-fe.semgrep-rules.ruby.lang.security.insufficient-rsa-key-size
2678  12:12:13           The RSA key size 1024 is insufficent by NIST standards. It is recommended to use a key
                        length of
2679  12:12:13           2048 or higher.
2680  12:12:13
2681  12:12:13            30¦ OpenSSL::PKey::RSA.new(1024)
2682  12:12:13
2683  12:12:13    ❱❱❱ rules-fe.semgrep-rules.ruby.lang.security.bad-deserialization-yaml
2684  12:12:13           Unsafe deserialization from YAML. Objects in Ruby can be serialized into strings, then
                        later loaded
2685  12:12:13           from strings. However, uses of load and object_load can cause remote code execution. Lo
                        ading user
2686  12:12:13           input with YAML can potentially be dangerous. Use JSON in a secure fashion instead. How
                        ever, loading
2687  12:12:13           YAML from a static file is not dangerous and should not be flagged.
2688  12:12:13
2689  12:12:13            ▶▶¦ Autofix ▶ Psych.safe_load(text)
2690  12:12:13            34¦ YAML.load(text)
2691  12:12:13
2692  12:12:13     ❱❱ rules-fe.semgrep-rules.ruby.rails.security.audit.avoid-tainted-file-access
2693  12:12:13           Using user input when accessing files is potentially dangerous. A malicious actor could
                        use this to
2694  12:12:13           modify or access files they have no right to.
2695  12:12:13
2696  12:12:13            42¦ File.read(params[:path]) if path
2697  12:12:13
2698  12:12:13
2699  12:12:13  ┌──────────────┐
2700  12:12:13  │ Scan Summary │
2701  12:12:13  └──────────────┘
2702  12:12:13  Ran 2190 rules on 28 files: 221 findings.
2703  12:12:13  $ if [ ! -s opengrep-scan/opengrep.json ]; then # collapsed multi-line command
2704  12:12:13  FE_OPENGREP_RULES_IN_SET=2407
2705  12:12:13  FE_OPENGREP_RULES_RUN=2190
2706  12:12:13  FE_OPENGREP_FILES_CONSIDERED=116
2707  12:12:13  FE_OPENGREP_PATHS_SCANNED=28
2708  12:12:13  FE_OPENGREP_NOTHING_TO_SCAN=false
2709  12:12:13  FE_OPENGREP_VERSION_RAN=1.30.0
2710  12:12:13  FE_OPENGREP_RULES_VERSION=2.9.3-fe2
```

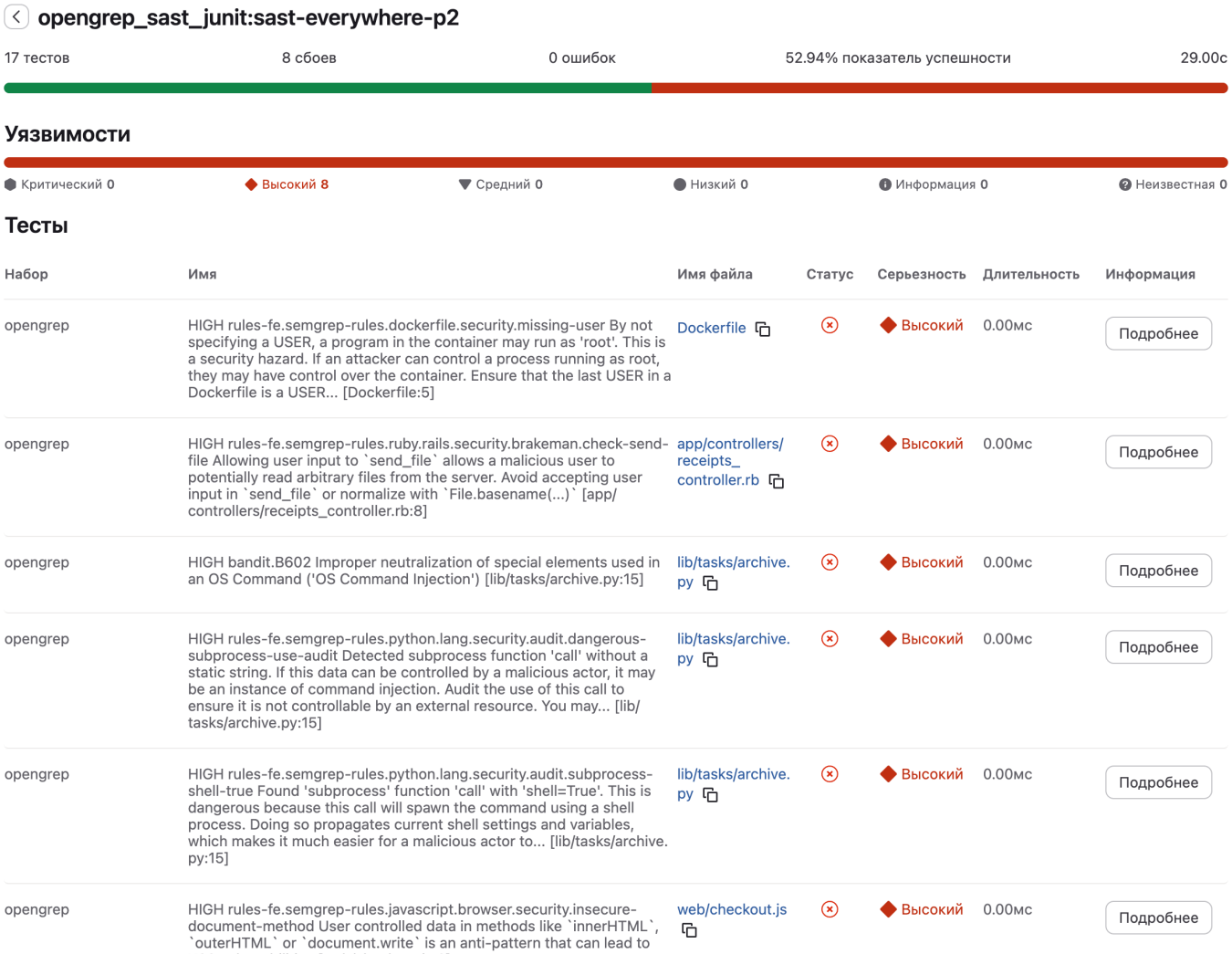
Репозиторий, все файлы которого написаны на языках, которых набор правил не покрывает, завершает задачу сканирования строкой лога `Opengrep: nothing was scanned, so this run publishes no report.`: обе задачи завершаются успешно, а отчёт не публикуется, поэтому ничто ниже по цепочке не прочитает пустой отчёт как исправленные находки.

Находки попадают туда же, куда и находки остальных сканеров политики:

Отчёт	Где смотреть
Тесты (JUnit)	Вкладка «Тесты» пайплайна и сводка тестов в запросе на слияние
Отчёт безопасности (`sast`)	Страница «Отчёт об уязвимостях»
Находки, переданные в DefectDojo	DefectDojo, если настроена интеграция с DefectDojo



Каждый тестовый случай назван по находке: уровень, идентификатор правила (основной идентификатор или, если его нет, `check_id`), первая строка краткого описания или сообщения, обрезанная до 255 символов, и файл со строкой в квадратных скобках. К повторяющемуся имени добавляется счётчик.



opengrep_sast_junit:sast-everywhere-p2

17 тестов | 8 сбоев | 0 ошибок | 52.94% показатель успешности | 29.00с

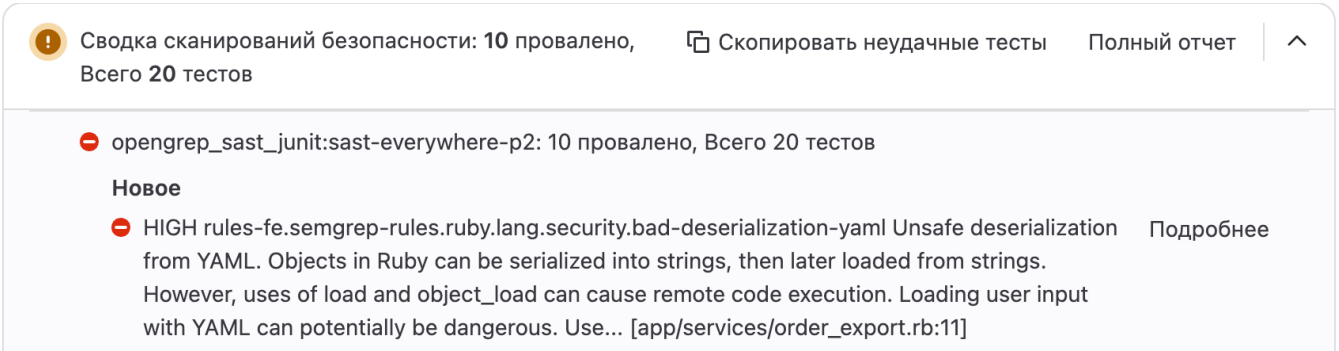
Уязвимости

Критический 0 | Высокий 8 | Средний 0 | Низкий 0 | Информация 0 | Неизвестная 0

Тесты

Набор	Имя	Имя файла	Статус	Серьезность	Длительность	Информация
opengrep	HIGH rules-fe.semgrep-rules.dockerfile.security.missing-user By not specifying a USER, a program in the container may run as 'root'. This is a security hazard. If an attacker can control a process running as root, they may have control over the container. Ensure that the last USER in a Dockerfile is a USER... [Dockerfile:5]	Dockerfile		Высокий	0.00мс	Подробнее
opengrep	HIGH rules-fe.semgrep-rules.ruby.rails.security.brakeman.check-send-file Allowing user input to `send_file` allows a malicious user to potentially read arbitrary files from the server. Avoid accepting user input in `send_file` or normalize with `File.basename(...)` [app/controllers/receipts_controller.rb:8]	app/controllers/receipts_controller.rb		Высокий	0.00мс	Подробнее
opengrep	HIGH bandit.B602 Improper neutralization of special elements used in an OS Command ('OS Command Injection') [lib/tasks/archive.py:15]	lib/tasks/archive.py		Высокий	0.00мс	Подробнее
opengrep	HIGH rules-fe.semgrep-rules.python.lang.security.audit.dangerous-subprocess-use-audit Detected subprocess function 'call' without a static string. If this data can be controlled by a malicious actor, it may be an instance of command injection. Audit the use of this call to ensure it is not controllable by an external resource. You may... [lib/tasks/archive.py:15]	lib/tasks/archive.py		Высокий	0.00мс	Подробнее
opengrep	HIGH rules-fe.semgrep-rules.python.lang.security.audit.subprocess-shell-true Found 'subprocess' function 'call' with 'shell=True'. This is dangerous because this call will spawn the command using a shell process. Doing so propagates current shell settings and variables, which makes it much easier for a malicious actor to... [lib/tasks/archive.py:15]	lib/tasks/archive.py		Высокий	0.00мс	Подробнее
opengrep	HIGH rules-fe.semgrep-rules.javascript.browser.security.insecure-document-method User controlled data in methods like `innerHTML`, `outerHTML` or `document.write` is an anti-pattern that can lead to XSS vulnerabilities [web/checkout.js:2]	web/checkout.js		Высокий	0.00мс	Подробнее

Виджет запроса на слияние «Сводка сканирований безопасности» сравнивает отчёт о тестах пайплайна запроса на слияние с отчётом пайплайна целевой ветки. Находки, которые вносит запрос на слияние, показаны в группе «Новое», за ними — уже существующие в целевой ветке; тесты, которые он исправляет, показаны в группе «Исправлено».

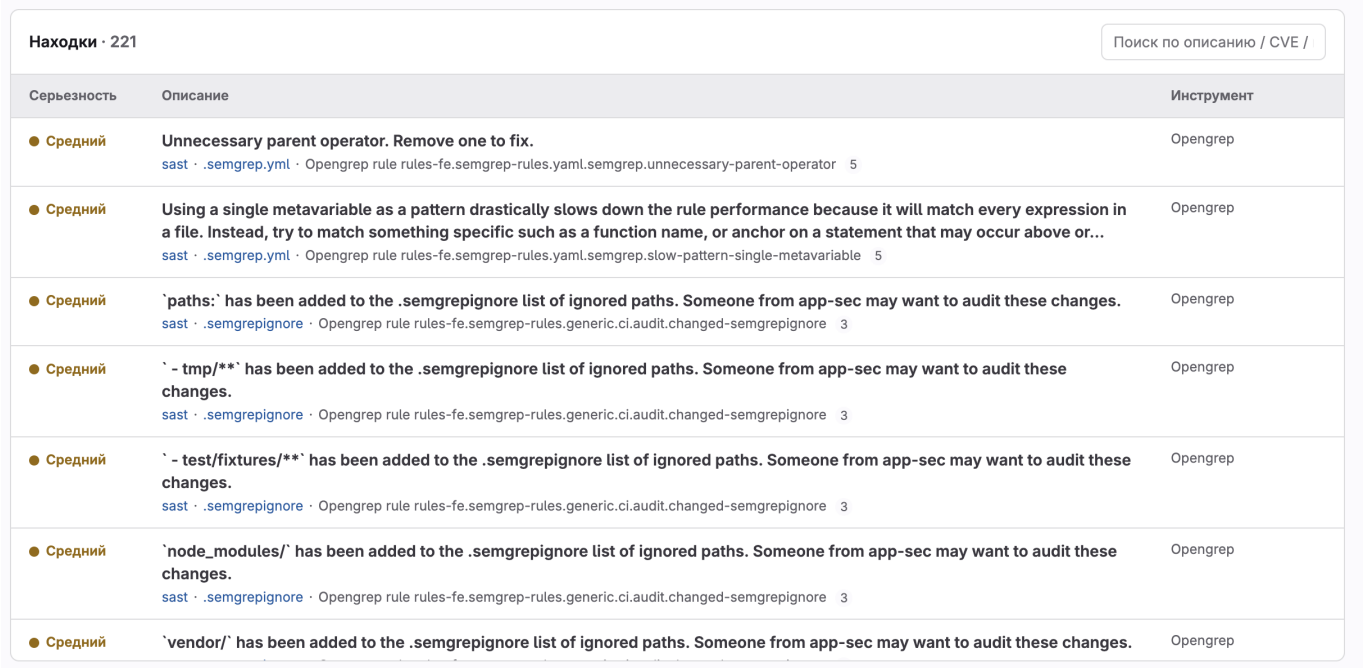


Сводка сканирований безопасности: **10** провалено, Всего **20** тестов | Скопировать неудачные тесты | Полный отчет

opengrep_sast_junit:sast-everywhere-p2: 10 провалено, Всего 20 тестов

Новое

HIGH rules-fe.semgrep-rules.ruby.lang.security.bad-deserialization-yaml Unsafe deserialization from YAML. Objects in Ruby can be serialized into strings, then later loaded from strings. However, uses of load and object_load can cause remote code execution. Loading user input with YAML can potentially be dangerous. Use... [app/services/order_export.rb:11] Подробнее



Предварительные требования

Чтобы сканирование могло запуститься, убедитесь, что:

- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- проект политик безопасности привязан к проекту или к группе выше него;
- доступен GitLab Runner с executor `docker`, и он может загрузить оба образа, которые использует сканирование: образ сканера (раздел «Откуда берётся образ сканера») и образ Ruby, в котором выполняется вторая задача, — по умолчанию `ruby:3.3.10-slim`, установка перенаправляет его переменной окружения сервера `FE_SCANS_REPORT_CONVERTER_IMAGE`.

Включение сканирования

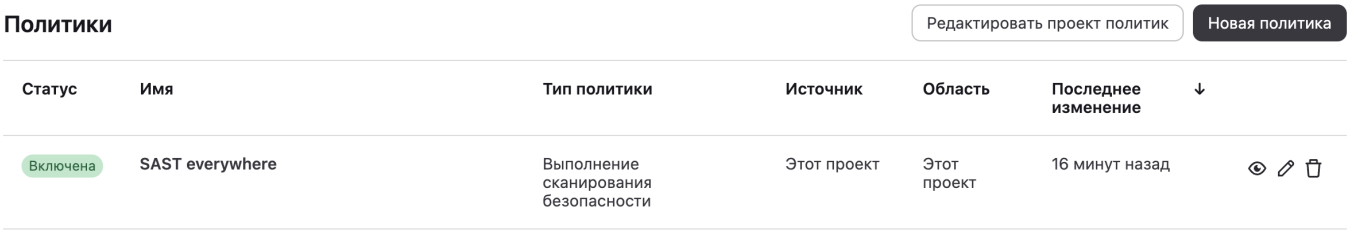
Добавьте действие `opengrep_sast` в файл `.deckhouse_code/security-policies/policy.yml` проекта политик безопасности:

```
scan_execution_policy:
  - name: SAST everywhere
    enabled: true
    rules:
      - type: pipeline
        branches: ["*"]
    actions:
      - scan: opengrep_sast
```

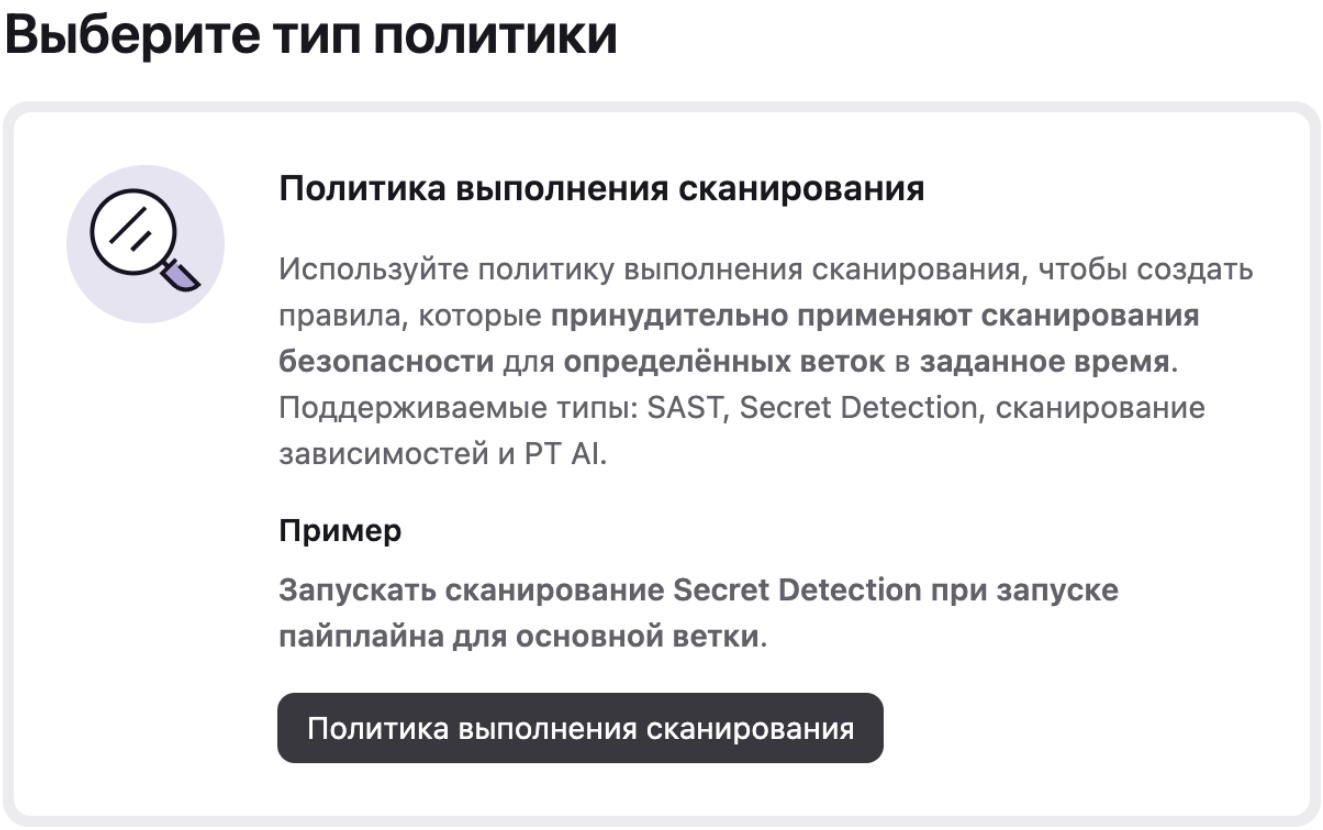
`opengrep_sast` — идентификатор, который схема политики принимает для этого сканирования. Политика, в действии которой написано `semgrep_sast` или `sast`, не проходит проверку и ничего не требует, пока действие не переименовано.

Затем привяжите проект политик к нужному проекту или группе в разделе «Настройки» → «Политика безопасности». После этого каждый пайплайн, попадающий под правила политики, получает две задачи, описанные выше.

Ту же политику можно создать в интерфейсе. В проекте или группе выберите «Безопасность» → «Политики» и нажмите «Новая политика»:



На экране «Выберите тип политики» выберите «Политика выполнения сканирования»:



В редакторе политик это сканирование называется «SAST (встроенный Opengrep)» в списке «Сканирования безопасности» и считается обязательным, пока включена хотя бы одна из его групп, описанных ниже. Условия запуска — флажки под заголовком «Когда запускать сканирования»: «На любом пайплайне», «Запросы на слияние», «Ветка по умолчанию»,

«Сканирование по расписанию» с полем «Расписание» и «По имени ветки» с полем «Шаблон ветки». Флажки сочетаются; «На любом пайплайне» снимает флажки «Запросы на слияние», «Ветка по умолчанию» и «По имени ветки» и держит их недоступными, пока включён сам. Политику без единого условия запуска сохранить нельзя, а отправленную в репозиторий вручную отвергает схема.

Что задаёт политика

Форма действия `opengrep_sast` собрана из групп «Образ сканера», «Правила», «Уровни правил», «Исключаемые категории», «Порог блокировки», «Игнорируемые пути» и «Дополнительные параметры». Все параметры принадлежат политике: значения записываются в задачу сервером при формировании пайплайна, поэтому собственные переменные CI/CD проверяемого проекта их не меняют.

У каждой группы есть флажок рядом с заголовком. Установленный флажок раскрывает группу, и политика забирает эту часть сканирования себе для всех проектов, которые покрывает: «Правила» начинают с пути к набору правил `/rules-fe`, «Уровни правил» — со всех трёх уровней, «Порог блокировки» — с «Высокий и выше», а остальные группы ничего не записывают, пока в них не заполнено поле. Снятый флажок означает, что группа ничего не записывает в политику и эта часть сканирования остаётся с ответом, который задача даёт без неё, — тем же самым. Если снять флажки у всех групп, действие `opengrep_sast` удаляется из политики. Сохраните политику и откройте её заново — группы, в которых что-то задано, откроются сами, поэтому сохранённая политика показывает всё, что задаёт.

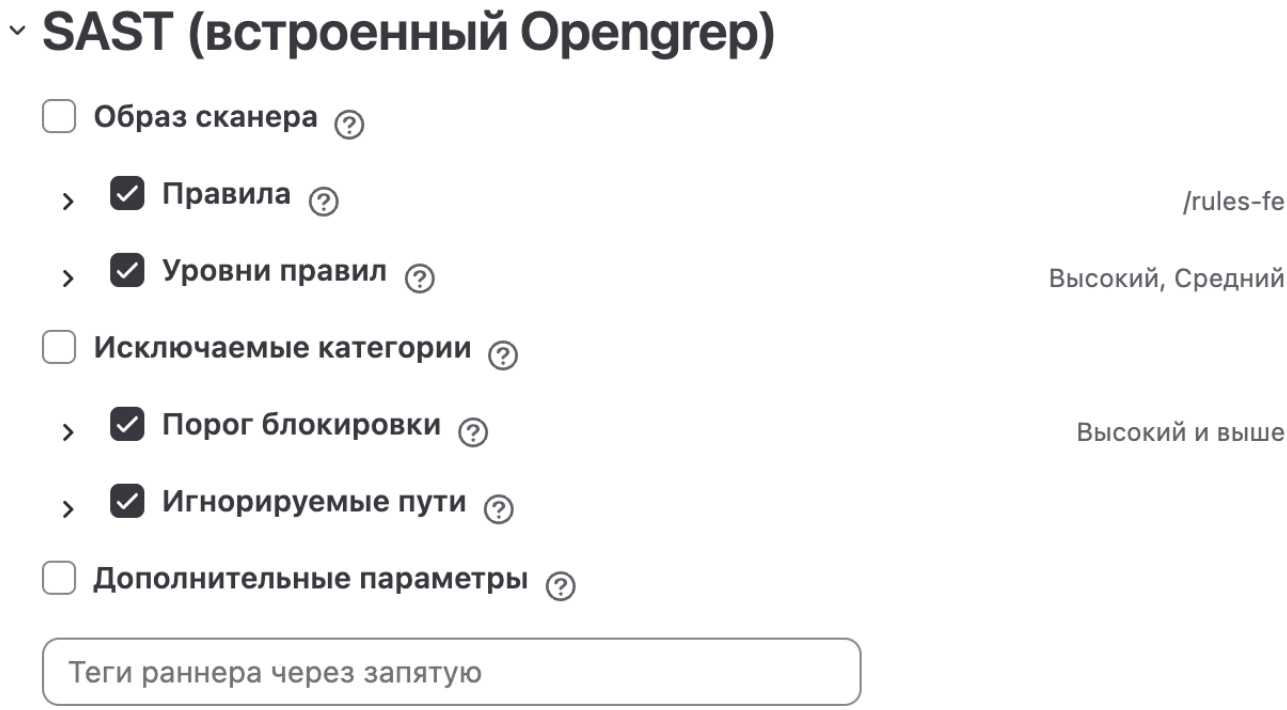
Группа названа по полю, вокруг которого собрана, и это поле нарисовано без собственной подписи внутри группы: «Уровни правил» раскрываются прямо на трёх флажках уровней, «Порог блокировки» — на пороге, «Дополнительные параметры» — на текстовом поле, а «Образ сканера» держит своё поле в строке заголовка. Подсказка («?») рядом с заголовком группы несёт подсказку этого поля либо описание самой группы.

То, что решила свёрнутая группа, написано в конце строки её заголовка:

Группа	Что написано в её свёрнутой строке
«Образ сканера»	Ничего: поле остаётся в строке заголовка, и ссылка на образ читается прямо там
«Правила»	Путь к файлу правил и проект, из которого он читается, либо путь к набору правил в образе
«Уровни правил»	Уровни, включённые политикой, например «Высокий, Средний»
«Исключаемые категории»	Ничего. В группе стоят флажки
«Порог блокировки»	Порог — «Высокий и выше», пока политика не выберет другой
«Игнорируемые пути»	Путь к файлу фильтров и проект, из которого он читается
«Дополнительные параметры»	Ничего. В свободном тексте выбора из списка нет

Поле, которое отвечает одному варианту выбора, показывается только после этого выбора, а значение скрытого формой поля удаляется из политики: названный файл правил убирает с формы и из политики поля «Путь к набору правил в образе», «Правила» и флажки группы «Исключаемые категории».

Политика, написанная вручную, может также задать в `variables` действия переменную `FE_SCANS_OPENGREP_SRC_DIRECTORY` — каталог внутри рабочей копии: сканирование тогда читает только этот каталог и сообщает находки с путями относительно рабочей копии, а каталог, которого в репозитории нет, останавливает задачу.


SAST (встроенный Opengrep)
Образ сканера
Правила
/rules-fe
Уровни правил
Высокий, Средний
Исключаемые категории
Порог блокировки
Высокий и выше
Игнорируемые пути
Дополнительные параметры
Теги раннера через запятую

SAST (встроенный Opengrep)

- [ ] Образ сканера
- [x] Правила
  - Путь к файлу: .opengrep.yml
  - Путь к набору правил в образе: /rules-fe
  - Правила:
  - [ ] Заменить поставленный набор
- [x] Уровни правил
  - [x] Высокий
  - [x] Средний
  - [ ] Информационный
- [ ] Исключаемые категории
- [x] Порог блокировки
  - (•) Высокий и выше
  - ( ) Средний и выше
  - ( ) Любая находка
  - ( ) Только сообщать, ничего не блокировать
- [x] Игнорируемые пути
  - Путь к файлу: .semgrepignore
  - Исключить пути:

```
docs/
*.min.js
```

- [ ] Учитывать .gitignore репозитория

Образ сканера

Группа — это одно поле «Образ сканера» в строке заголовка. Если поле пустое, сканирование выполняется в образе, заданном для этой установки: он показан в поле серым, и его называет подсказка: «Если оставить пустым, сканирование выполняется в образе `<image>`.» Подсказка сообщает и то, что несёт поставляемый образ: «С продуктом поставляется opengrep 1.30.0, в образе 1.30.0-fe2, с набором правил 2.9.3-fe2.»

Поле	Что задаёт
«Образ сканера»	Образ, в котором выполняется сканирование, — ссылка целиком: путь из двух и более частей и тег либо дайджест `@sha256:`, например `registry.example.com/security/opengrep-scan:1.30.0-fe2`. Первая часть без точки и порта читается как путь на Docker Hub. Набор правил берётся из самого образа, поэтому для образа, который его не несёт, правила нужно назвать в политике. Если поле пустое — образ, заданный для этой установки.

Поле ничего не предлагает на выбор: установка, которая зеркалирует сканер или собирает свой образ, называет ссылку, которую продукт перечислить не мог.

Ссылка, устроенная иначе, сохранению политики не мешает. Задача тогда стартует в образе, заданном для установки, и останавливается до сканирования с сообщением `this scan cannot run the configuration the policy asks for: scanner image: …`. Тег вроде `latest` проверку проходит, и тогда сканер меняется всякий раз, когда хранилище образов передвигает тег.

Что означает пустое поле и где администратор задаёт этот образ — в разделе Откуда берётся образ сканера. Для образа в закрытом хранилище образов под полем задаются имена переменных логина и пароля: раздел Учётные данные хранилища образов для сканера.

Правила

Группа задаёт, какие правила выполняет сканирование. Её подсказка: «Какие правила выполняет сканирование. Если ничего не менять — набор, поставляемый с продуктом. Укажите файл, чтобы выполнять свои правила рядом с ним или вместо него.»

Поле	Что задаёт
«Путь к файлу»	Путь к файлу правил внутри репозитория, из которого он читается, например `.opengrep.yml`, показанный в поле серым. Если поле пустое, всё сканирование — это поставленный набор, и файл не читается.
«Проект»	Из какого репозитория читается файл правил, в виде `группа/проект`. Показывается, когда заполнено поле «Путь к файлу». Если поле пустое — из проверяемого проекта, на строящемся коммите.
«Путь к набору правил в образе»	Какой из наборов, лежащих в образе сканера, выполнять (раздел «Наборы правил в образе»). По умолчанию — `/rules-fe`. Показывается, пока файл не назван. Если по этому пути набора нет, сканирование останавливается.
«Правила»	Сами правила, в YAML сканера; записываются в задачу сервером. Показывается, пока файл не назван. Не больше 16 КБ.
«Заменить поставленный набор»	По умолчанию выключено: ваши правила выполняются **вместе** с поставленным набором. Если установить флажок, они выполняются **вместо** него, и всё, что покрывал поставленный набор, перестаёт проверяться.

Источники правил отличаются тем, кто владеет файлом:

Источник	Где лежат правила	Кто ими распоряжается
Поле «Путь к файлу» пустое	В образе сканера, по пути из поля «Путь к набору правил в образе»	Поставка продукта
«Путь к файлу» заполнен, «Проект» пустой	Сканируемый репозиторий, на строящемся коммите	**Проверяемый проект**
«Путь к файлу» заполнен, «Проект» назван	Этот проект, на его ветке по умолчанию	Мейнтейнеры этого проекта
«Правила»	Сама политика	Автор политики

Файл в проверяемом проекте задача читает из своей рабочей копии. Файл в другом проекте задача читает сама, через API, своим токеном задачи CI, поэтому кому разрешено его читать, решает этот проект: его мейнтейнер добавляет проверяемый проект в разделе «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD». Задача останавливается и называет эту настройку, если проект отвечает `401`, `403` или `404`, а также если файла нет или он пустой. Названный проект читается всегда на его ветке по умолчанию, на какой бы ветке ни выполнялся проверяемый пайплайн.

Правила, которые приносит политика — из файла или из поля «Правила», — задача записывает в образ по пути `/rules/policy/rules.yml`, поэтому их находки в отчётах названы `rules.policy.<идентификатор правила>`; находки поставленного набора названы по пути к набору, при пути по умолчанию — `rules-fe.<идентификатор правила>`. Перед сканированием задача проверяет принесённый набор: набор, который сканер не может разобрать, набор без единого правила и набор, правилам которого нужен движок новее, чем в образе, останавливают задачу и печатают вывод сканера. Снятый флажок «Заменить поставленный набор» оставляет поставленный набор выполняться рядом с ними.

Уровни правил

Группа задаёт, правила каких уровней вообще выполняются.

Поле	Что задаёт
«Уровни правил»	«Высокий» (ERROR), «Средний» (WARNING), «Информационный» (INFO). Все три флажка изначально установлены — именно это сканирование и выполняет, пока политика не задала своих уровней.

Уровень — это ровно те правила, которые сканер помечает им: «Высокий» — правила ERROR, «Средний» — WARNING, «Информационный» — INFO; находка попадает в отчёт с уровнем сработавшего правила. Снятый флажок убирает из прогона ровно правила этого уровня.

Последний оставшийся установленный флажок снять нельзя: опустевший список удаляется из политики, а политика, которая не называет уровней, — это то состояние, в котором выполняются все уровни.

«Уровни правил» и «Порог блокировки» выглядят одинаково, а делают разное:

Поле	На что влияет	Находка ниже уровня
«Уровни правил»	Какие правила **выполняются**	Её не существует: правило не выполнялось
«Порог блокировки»	Какие находки **приводят к ошибке пайплайна**	Она есть в отчёте, но не блокирует

Порог покрывает свой уровень и все уровни выше — это и означает «и выше»; уровень правил покрывает правила, помеченные сканером этим уровнем, и никакие другие.

Поэтому исключённый уровень убирает свои находки и из отчёта: отчёт об уязвимостях и DefectDojo становятся меньше, и в логе задачи об этом ничего не сказано. Порог, которого не достигает ни один установленный уровень, не сработает никогда; форма говорит об этом рядом с порогом: «Ни один уровень правил, способный достичь этого порога, не включён, поэтому пайплайн не упадёт никогда.» Уровни, после которых в наборе не остаётся ни одного правила, останавливают задачу сканирования с сообщением `no rule survived the levels this policy runs`.

Исключаемые категории

Группа задаёт, какие категории поставленного набора сканирование не выполняет. Она показывается, пока правила берутся из поставленного набора: названный файл правил убирает группу с формы, потому что у своих правил категорий нет.

Поле	Что задаёт
«Исключить правила для секретов»	По умолчанию выключено. Если установить флажок, правила, которые сообщают об учётных данных, попавших в репозиторий, в прогон не идут.
«Исключить правила для конфигурации»	По умолчанию выключено. Если установить флажок, правила, которые читают конфигурацию и инфраструктурный код — Terraform, Dockerfile, манифесты Kubernetes, конфигурацию CI, — в прогон не идут.
«Исключить правила для вредоносного кода»	По умолчанию выключено. Если установить флажок, правила, которые ищут признаки намеренно скрытого кода, в прогон не идут.

Установленный флажок исключает категорию, снятый оставляет её в прогоне — так же продолжает вести себя политика, написанная до появления этих полей. Подсказка группы говорит, зачем этот выбор нужен: если одну из этих категорий уже покрывает другое обязательное сканирование, о ней сообщат оба, и одна и та же строка станет находкой дважды.

Что лежит в каждой категории и какой набор правил нужен для исключения — в разделе «Категории внутри поставленного набора».

Порог блокировки

Группа задаёт, что находка делает с пайплайном.

Поле	Что задаёт
«Порог блокировки»	Что приводит к ошибке пайплайна. Находки ниже порога всё равно попадают в отчёт и в отчёт об уязвимостях: порог решает только исход пайплайна. По умолчанию — «Высокий и выше».

Значения порога, каждое с подсказкой, которую форма показывает рядом:

Значение	Что приводит к ошибке пайплайна
«Высокий и выше» (по умолчанию)	Находка высокого уровня роняет пайплайн. Средний и информационный только попадают в отчёт.
«Средний и выше»	Средний и высокий уровни роняют пайплайн. Информационный только попадает в отчёт.
«Любая находка»	Любая находка роняет пайплайн, какого бы уровня она ни была.
«Только сообщать, ничего не блокировать»	Ничто не роняет пайплайн. Находки всё равно фиксируются в отчёте и попадают в отчёт об уязвимостях.

Opengrep сообщает об уровнях ERROR, WARNING и INFO, которые сканирование показывает как высокий, средний и информационный. Критического и низкого у него нет, поэтому порога с таким именем не достигала бы ни одна находка. Значение «Только сообщать, ничего не блокировать» вносит отказ от блокировки в политику отдельным значением. Порог вне этого списка, возможный только в политике, написанной вручную, останавливает задачу до сканирования.

Игнорируемые пути

Группа задаёт, что сканирование оставляет за рамками. Её подсказка: «Что сканирование не смотрит. Собственные исключения задачи для вендорного и сгенерированного кода действуют независимо от того, что задано здесь.»

Поле	Что задаёт
«Путь к файлу»	Путь к файлу игнорируемых путей внутри репозитория, из которого он читается, в формате `.semgrepignore`, который читает сканер; в поле серым показан `.semgrepignore`. Если поле пустое, файл не читается.
«Проект»	Из какого репозитория читается этот файл, в виде `группа/проект`. Показывается, когда заполнено поле «Путь к файлу». Если поле пустое — из проверяемого проекта, на строящемся коммите; названный проект читается так же, как файл правил, токеном задачи.
«Исключить пути»	По одному пути или шаблону в строке: сторонний код, фикстуры, сгенерированные файлы. Передаются сканеру при любом файле.
«Учитывать .gitignore репозитория»	По умолчанию выключено: обязательное сканирование смотрит и в игнорируемые файлы, потому что список игнорируемого принадлежит самому проверяемому проекту.

Перед сканированием задача убирает каждый `.semgrepignore`, который несёт репозиторий, — в корне и в подкаталогах. Файл, названный политикой, читается до этого и записывается обратно как единственный файл игнорирования прогона, с дописанными собственными исключениями задачи: `node_modules/`, `vendor/`, `dist/`, `build/`, `.venv/` и `.git/`. Пустой файл останавливает задачу.

«Путь к файлу» с пустым полем «Проект» и «Учитывать .gitignore репозитория» одинаково отдают проверяемому проекту право решать, что осматривает обязательное сканирование: после этого коммит в проверяемом проекте может сузить сканирование, которое его же и проверяет, и в политике это сужение не видно. Команда, которая позволяет каждому проекту называть свои каталоги стороннего кода, фиксирует это делегирование в политике, потому что обе настройки остаются выключенными, пока эту группу не раскроют и одну из них не зададут.

Шаблон, не совпавший ни с чем, проходит проверку формы. Задача печатает, сколько файлов сканер взял в работу, сколько из них убрали фильтры и сколько он прочитал, и завершается с ошибкой, если фильтры убрали всё или в рабочей копии ничего нет.

Дополнительные параметры

Группа задаёт флаги, которых в остальной форме нет.

Поле	Что задаёт
«Дополнительные параметры»	Флаги, передаваемые сканированию как есть.

«Дополнительные параметры» настраивают ресурсы запуска: тайм-ауты, лимиты памяти, максимальный размер файла. Флаги, которыми распоряжается задача, отклоняются: `--config`, `-c`, `-f`, `--output`, `-o`, `--json`, `--json-output`, `--sarif`, `--sarif-output`, `--junit-xml`, `--junit-xml-output`, `--emacs`, `--emacs-output`, `--vim`, `--vim-output`, `--text`, `--text-output`, `--gitlab-sast`, `--gitlab-sast-output`, `--gitlab-secrets`, `--gitlab-secrets-output`, `--metrics` и `--baseline-commit`. Сервер отклоняет их при формировании задачи, и задача останавливается до сканирования, называя флаг.

Подсказка этого поля сообщает и то, куда попадают находки сканирования: они публикуются как отчёт безопасности, а значит, доходят до отчёта об уязвимостях и до всего, что администратор к нему подключил.

Наборы правил в образе

По умолчанию сканирование выполняет набор правил, лежащий внутри образа сканера, по пути `/rules-fe`. В этом каталоге лежат наборы ниже, и поле «Путь к набору правил в образе» выбирает между ними:

Путь в образе	Правил	Что это
`/rules-fe` (по умолчанию)	2825	Наборы ниже вместе
`/rules-fe/semgrep-rules`	1981	Открытый набор `semgrep-rules`
`/rules-fe/gitlab`	483	Правила анализаторов GitLab
`/rules-fe/community`	361	Правила сообщества

Значения измерены в поставляемом образе `opengrep-scan:1.30.0-fe2` и относятся к нему; в более новом образе состав может быть другим. У каждого набора свои условия распространения, и различают их именно каталоги: укажите в поле «Путь к набору правил в образе» один набор, чтобы выполнять только его.

Покрытие набора по числу правил на язык: `python` 618, `terraform` 366, `javascript` 352, `java` 341, `typescript` 235, `go` 195, `c` 194, `scala` 191, `cpp` 161, `ruby` 150, `php` 122, `kotlin` 105, а также `solidity`, `yaml`, `dockerfile`, `swift`, `csharp` и другие. Правило может называть несколько языков, поэтому сумма больше числа правил. Сканер выполняет только правила для языков найденных им файлов, поэтому в логе задачи на репозитории из одного JavaScript выполненных правил заметно меньше, чем в наборе.

Собственные правила задаются полями «Путь к файлу» или «Правила» и, если флажок «Заменить поставленный набор» остался снятым, выполняются вместе с тем набором, который даёт образ.

Категории внутри поставленного набора

Правила в наборе размечены по категориям: категория говорит о том, о чём судит правило.

Категория	О чём судят её правила
Код	Код самой программы
Конфигурация	Конфигурация и инфраструктурный код: Terraform, Dockerfile, манифесты Kubernetes, конфигурация CI
Секреты	Учётные данные, попавшие в репозиторий
Вредоносный код	Признаки намеренно скрытого кода: обфускация, код, собираемый во время выполнения, вызовы через промежуточные слои

Конфигурацию, секреты и вредоносный код политика может исключить — в группе «Исключаемые категории»; правила о коде выполняются в каждом сканировании.

Для исключения нужен набор правил, который несёт индекс своих категорий — файл `CLASSES.tsv` рядом с набором по пути из поля «Путь к набору правил в образе». Образ из поставки этого релиза такого индекса не несёт, поэтому установленный флажок останавливает на нём задачу: она называет категории, которые её просили исключить, и набор правил, у которого нет индекса для этого (раздел «Диагностика»). Флажки применимы к собственному образу, набор которого несёт индекс.

⚠ Правила для секретов и сканирование `gitleaks_secret_detection` смотрят на одно и то же, и каждое сканирование публикует свой отчёт безопасности, поэтому одна и та же строка приезжает двумя находками на двух уровнях: измерено на одном прогоне, Opengrep положил попавший в репозиторий токен в отчёт с уровнем Info, а `gitleaks_secret_detection` — тот же токен с уровнем Critical. Наборы находок не вложены друг в друга, поэтому исключение правил для секретов передаёт категорию целиком `gitleaks_secret_detection` и вместе с ней убирает из SAST-отчёта те находки, которые сделал только Opengrep.
То же рассуждение годится для правил конфигурации, если инфраструктуру уже проверяет отдельный сканер, и для правил вредоносного кода: они читают намерение, поэтому репозиторий, который обфусцирует или собирает код во время выполнения по своим причинам, увидит их срабатывания.

Откуда берётся образ сканера

Образ, в котором выполняется сканирование, разрешается сервером один раз, до старта задачи, и записывается в неё готовым значением. Переменной CI/CD он не становится, поэтому настройки проверяемого проекта не могут направить сканирование на другой сканер. Задача сканирования печатает ссылку на образ, в котором выполнялась, в начале лога строкой `Analyzer image …`, а рядом — движок, который несёт этот образ.

Какой это образ, следует из поля «Образ сканера» политики:

«Образ сканера» в политике	Что загружает сканирование
Заполнено	Эту ссылку, ровно как она написана.
Пустое	Образ, заданный для этой установки: значение переменной окружения `FE_SCANS_OPENGREP_IMAGE` сервера Deckhouse Code, а если переменная не задана — образ, зафиксированный этим релизом.

`FE_SCANS_OPENGREP_IMAGE` — переменная окружения серверных процессов, которые формируют пайплайны: веб-сервера и Sidekiq. Администратор задаёт её в их окружении, а читается она в момент формирования задачи. То же имя среди переменных CI/CD проекта или группы ни на что не влияет: задача никогда не читает образ из переменной. В файле политики это поле — ключ `FE_SCANS_OPENGREP_IMAGE` в `variables` действия; именно его записывает редактор:

```
actions:
  - scan: opengrep_sast
    variables:
      FE_SCANS_OPENGREP_IMAGE: registry.example.com/security/opengrep-scan:1.30.0-fe2
```

Все параметры этого сканирования, включая образ, принадлежат политике, поэтому у Opengrep нет страницы настроек среди интеграций со сканерами; интеграции остальных обязательных сканеров описаны в разделе Интеграции со сканерами безопасности.

Версия из поставки продукта

Этот релиз поставляет `opengrep-scan:1.30.0-fe2` : Opengrep 1.30.0 с набором правил 2.9.3-fe2. Тег нумерует образ, а движок и набор правил внутри него меняются независимо: образ, пересобранный с более новым набором правил, получает новый тег, хотя движок остался прежним. Подсказка рядом с полем «Образ сканера» называет все три номера для поставляемого образа, а лог задачи сканирования печатает версии движка и набора правил, которые выполнялись.

Поле «Образ сканера» принимает обе формы фиксации версии:

- тег, например `1.30.0-fe2` , — его читает человек и с ним можно сравнивать;
- дайджест, например `@sha256:aafbcaff…` , — он называет одну сборку, и подменить её нельзя.

Ссылка без того и другого отклоняется. Хранилище образов — часть ссылки, поэтому зеркальная копия образа называется вместе со своим хранилищем.

Версия, поставляемая с релизом, пересматривается при обновлении Deckhouse Code.

Закрытый контур

Установка без доступа в интернет обеспечивает оба образа, которые использует сканирование:

Образ	По умолчанию	Чем перенаправляется
Сканер	`opengrep-scan:1.30.0-fe2` , из хранилища образов продукта	Переменная окружения сервера `FE_SCANS_OPENGREP_IMAGE` — она перенаправляет каждую политику с пустым полем «Образ сканера»; либо поле «Образ сканера» политики
Сборщик отчётов	`ruby:3.3.10-slim` , с Docker Hub	Переменная окружения сервера `FE_SCANS_REPORT_CONVERTER_IMAGE`

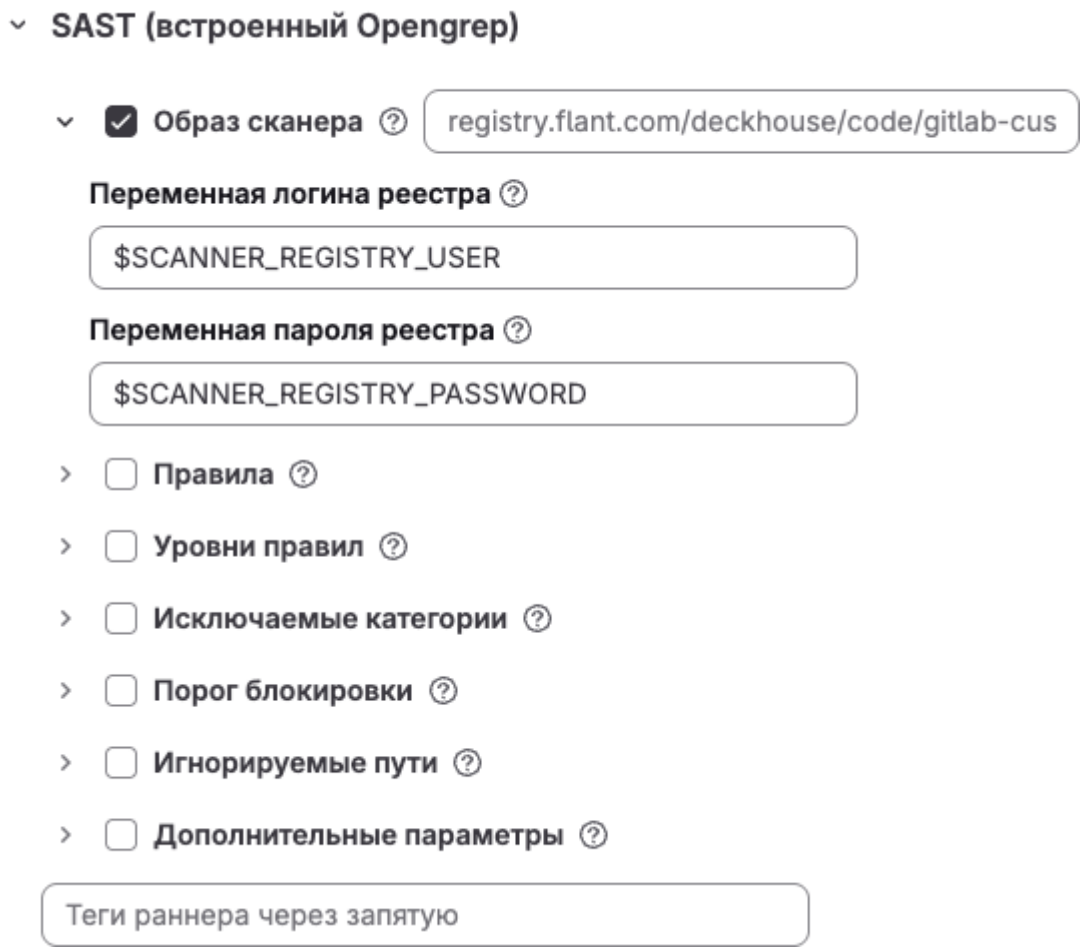
Скопируйте оба образа в своё хранилище образов контейнеров, а затем укажите это хранилище в соответствующей настройке. Если образ, заданный для установки, лежит в хранилище образов контейнеров этого же инстанса Deckhouse Code, редактор политик спрашивает у хранилища этот тег и, если тега нет, дописывает в подсказку рядом с полем «Образ сканера»: «В реестре нет такого тега: сканирование, которое возьмёт этот образ, не сможет запуститься, пока он не появится.» Стороннее хранилище не спрашивается.

> Образ сканера остаётся вне досягаемости dependency proxy. Прокси кеширует образы, лежащие на Docker Hub, а образ сканера публикуется в хранилище образов продукта, поэтому из двух образов прокси кеширует образ Ruby. Установка, которая хочет перестать загружать образ сканера извне при каждом запуске — в закрытом контуре или нет, — копирует образ в своё хранилище и указывает эту копию в переменной `FE_SCANS_OPENGREP_IMAGE` или в поле «Образ сканера».

Учётные данные хранилища образов для сканера

Учётные данные нужны, когда образ сканера, заданный в переменной окружения сервера, в интеграции или в политике, находится в закрытом хранилище образов. Не записывайте пароль в `policy.yml`: файл хранится в репозитории открытым текстом. Политика называет две переменные CI/CD, в которых хранятся логин и пароль, а значения берутся из настроек CI/CD сканируемого проекта или группы.

В редакторе политик это поля «Переменная логина реестра» и «Переменная пароля реестра» (у сканирования CodeScoring они собраны в группе «Учётные данные реестра»).



В файле политики это ключи `variables` действия:

```
actions:
  - scan: opengrep_sast
    variables:
      FE_SCANS_OPENGREP_IMAGE: registry.example.com/security/opengrep-scan:1.30.0-fe2
      FE_SCANS_REGISTRY_LOGIN_VARIABLE: $SCANNER_REGISTRY_USER
      FE_SCANS_REGISTRY_PASSWORD_VARIABLE: ${SCANNER_REGISTRY_PASSWORD}
```

Чтобы настроить доступ:

1. В настройках CI/CD группы или проекта создайте две переменные с теми именами, которые назвала политика, — в примере `SCANNER_REGISTRY_USER` и `SCANNER_REGISTRY_PASSWORD`. Переменные группы, к которой привязана политика, действуют для всех проектов под ней.
2. Сделайте переменную пароля маскируемой. Для обеих переменных оставьте в поле «Тип» значение «Переменная (по умолчанию)». Пароль не должен содержать символы `"` и `\`.
3. В политике укажите имена переменных в виде `$ИМЯ` или `${ИМЯ}`.

Что учитывать:

- Пара одна на все сканирования действия: имена задают обе или ни одной. Имена, начинающиеся с `FE_`, `CI_` и `GITLAB_`, политика отклоняет.
- Учётные данные получают сканирования Opengrep, Gitleaks, CodeScoring и Trivy, каждое — для хранилища своего образа. У Trivy поля находятся в блоке «Общие настройки», сразу после образа сканирования.
- PT AI берёт образ и переменные учётных данных реестра из своей интеграции; политика, которая называет их для действия `ptai`, не сохраняется (Интеграция с PT AI).
- Задачи, которые переводят отчёт сканера в JUnit (`opengrep_sast_junit`, `gitleaks_secret_junit`), учётных данных не получают: их образ задаёт администратор.
- В сканировании образов контейнеров и кластеров Kubernetes с помощью Trivy собственная переменная `DOCKER_AUTH_CONFIG` проекта по-прежнему действует для сканируемых образов; пара из политики используется только для загрузки образа самого Trivy (Сканирование с помощью Trivy).
- Образ, который поставляется с продуктом, загружается без учётных данных, даже если переменные названы: они действуют для образа, заданного в переменной окружения сервера, в интеграции или в политике. Исключение — Trivy: он получает учётные данные и для поставляемого образа `trivy-with-db`, потому что редактор политик записывает этот образ в политику.
- Переменные должны доходить до задачи сканирования. Защищённая переменная доходит только до пайплайнов защищённых веток и тегов. У задач сканирования нет окружения, поэтому переменная с областью окружения, отличной от `*`, до них не доходит. Без переменной загрузка образа завершается ошибкой.
- Групповая политика опирается на групповую переменную с этим именем. Если проект задаёт переменную с тем же именем, она заменяет групповую. При неверном значении загрузка образа завершается ошибкой, и задача останавливается до первой строки сценария.
- Если в проекте задана переменная `DOCKER_AUTH_CONFIG`, в задаче сканирования с учётными данными из политики она не действует, кроме сканирования образов контейнеров и кластеров в Trivy: её целиком заменяют учётные данные для хранилища образа сканера.
- Сами переменные `FE_SCANS_REGISTRY_LOGIN_VARIABLE` и `FE_SCANS_REGISTRY_PASSWORD_VARIABLE` в задачу не попадают.
- Логин и пароль может прочитать любая задача проекта, на который распространяется политика: это обычные переменные CI/CD проекта или группы.

Совместимость с Semgrep

Задача ищет в образе исполняемый файл `opengrep`, а если его нет — `semgrep`, который читает ту же командную строку; образ без обоих останавливает задачу с сообщением `no opengrep in this image`. С Semgrep сканирование выполняется с флагом `--metrics=off`, поэтому ничего не отправляет вендору; у Opengrep телеметрии нет, и выключать нечего. Какой бы движок ни выполнялся, сканирование запускается с `--disable-version-check`, а прежде чем движок

вообще будет найден, задача очищает все унаследованные переменные `OPENGREP_*`, `SEMGREP_*` и `PYTHON*` и сбрасывает `PATH`, поэтому настройки CI/CD проверяемого проекта не могут подменить набор правил, базу сравнения или исполняемый файл того сканирования, которое его проверяет.

Файл игнорирования сохраняет имя `.semgrepignore`, а открытый набор правил внутри образа — имя своего каталога `/rules-fe/semgrep-rules`.

Диагностика

Сканирование не появляется в пайплайне

Проверьте, что:

- проект политик безопасности привязан к проекту, а файл `.deckhouse_code/security-policies/policy.yml` в нём содержит действие `- scan: opengrep_sast`: политика с `semgrep_sast` или `sast` не проходит проверку и пропускается целиком, а файл по прежнему пути `.gitlab/security-policies/policy.yml` не читается;
- правила политики совпадают с пайплайном (ветка, источник пайплайна);
- доступен GitLab Runner с executor `docker`.

Задача завершается с ошибкой «this scan cannot run the configuration the policy asks for»

Сервер отклонил параметр при формировании задачи, и задача сканирования останавливается до сканирования, печатая причину: «Образ сканера» с неполной ссылкой, отклоняемый флаг в поле «Дополнительные параметры», уровень правил или порог блокировки, которых у сканера нет, правила в политике длиннее 16 КБ, «Проект», записанный иначе, чем `группа/проект`, либо названный «Проект» при пустом поле «Путь к файлу». Исправьте параметр в политике.

Задача завершается с ошибкой «does not let this project’s jobs read it» или «is not on the default branch of»

Файл правил или файл фильтров назван с полем «Проект», и либо права доступа токена задач этого проекта («Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD» в этом проекте) не содержат проверяемый проект, либо файла нет на ветке по умолчанию названного проекта. Проект, который задаче читать нельзя, отвечает так же, как отсутствующий файл, поэтому в обоих случаях проверьте права доступа.

Задача завершается с ошибкой «no rule set at … in this image»

Пути, указанного в поле «Путь к набору правил в образе», в разрешённом образе нет. Сверьте путь с таблицей выше и проверьте, какой образ используется на самом деле: лог задачи сканирования называет его строкой `Analyzer image`.

Задача завершается с ошибкой «carries no class index to switch them off by»

В группе «Исключаемые категории» установлен флажок, а разрешённый набор правил не несёт индекса своих категорий `CLASSES.tsv`. Снимите флажки или направьте сканирование на набор

правил с индексом. В логе задачи напечатаны категории, которые её просили исключить, и прочитанный набор правил.

Соседнее сообщение — что индекс не содержит ни одного правила ни в одной из названных категорий — разбирается так же.

Задача завершается с ошибкой «the rule set this policy supplies …»

Набор, названный в поле «Путь к файлу» или записанный в поле «Правила», не прошёл проверку, которую задача выполняет перед сканированием:
`is not valid configuration for this scanner` называет правило, которое сканер не может разобрать, `holds no rules` — пустой набор, `asks for a newer scanner than this image has` — правило, помеченное как несовместимое с движком в образе. За сообщением следует собственный вывод сканера.

Задача завершается с ошибкой «opengrep read no files at all»

Сканирование не осмотрело ничего и поэтому отказывается отчитываться об успехе. Обычные причины — фильтры, убравшие весь репозиторий, либо пустая рабочая копия. В логе задачи напечатано, сколько файлов она взяла в работу и сколько из них убрали фильтры.

Сканирование завершается успешно, но ничего не находит

Чаще всего набор правил не покрывает ни одного языка репозитория: задача сканирования тогда печатает `Opengrep: … file(s) here, and the rule set covers none of their languages.` и не публикует отчёт. Посмотрите лог задачи: в нём напечатан используемый путь к набору правил и число загруженных и выполненных правил. Тот же результат по другой причине даёт сужение полей «Уровни правил» или «Путь к набору правил в образе»: уровень или набор, который политика не включила, не выполняется, поэтому его находок в отчёте нет.

Обнаружение секретов с помощью Gitleaks

Обнаружение секретов в Deckhouse Code ищет учётные данные, попавшие в собственный исходный код проекта. Для этого используется сканер Gitleaks, сканирование называется `gitleaks_secret_detection`. Сканирование обязательное: его задаёт политика выполнения сканирования в отдельном проекте политик; файл `.gitlab-ci.yml` самого проекта его не добавляет. Политика решает, что приводит к ошибке пайплайна; сканируемый проект не может изменить это в своих настройках CI/CD.

Что делает сканирование

Действие `gitleaks_secret_detection` добавляет в пайплайн две задачи на этапе политики — `fe-security-scanner`, если в поле политики «CI-этап» не указан другой. К имени каждой задачи добавляется суффикс `:<короткое имя политики>-p<ID проекта политик>`, поэтому групповая и проектная политики, которые задают одно и то же сканирование, получают отдельные задачи.

Задача	Что делает
`gitleaks_secret_scan`	Запускает Gitleaks по репозиторию и публикует его JSON-вывод как артефакт
`gitleaks_secret_junit`	Строит из этого JSON отчёт JUnit и отчёт о безопасности и завершает задачу с ошибкой, если несоответствие не проходит порог

Обе задачи загружают артефакты с `when: always`, поэтому несоответствия, из-за которых пайплайн завершился с ошибкой, остаются в его артефактах. `gitleaks_secret_junit` также сохраняет несоответствия в формате SARIF, `security-reports/sarif/gitleaks.sarif`, — обычным артефактом для инструментов, которые читают этот формат.

Сканирование читает рабочую копию собираемого коммита. Коммит, который удаляет секрет, снимает несоответствие, но секрет остаётся в истории репозитория, пока историю не перепишут. Редактор политик другого выбора не предлагает. В политике, написанной вручную, можно задать `FE_SCANS_GITLEAKS_SCOPE: history` среди `variables` действия, чтобы читать всю историю, и `FE_SCANS_GITLEAKS_LOG_OPTS`, чтобы сузить её до диапазона коммитов в терминах `git log`.

Файл, хранящийся в Git LFS, сканируется по настоящему содержимому: указатель, который стоит вместо него в репозитории, не читается. Подмодуль не сканируется: читается только то, что оказалось в рабочей копии самого проекта.

Лог задачи сканирования указывает версию Gitleaks, а после завершения сканирования — сколько несоответствий найдено, по правилам. Сканирование, которое ничего не нашло, всё равно публикует отчёт — `[]`.

Предварительные требования

Чтобы сканирование могло запуститься, убедитесь, что:

- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- к проекту или к группе над ним привязан проект политик либо редактор политик создаст и привяжет его при сохранении первой политики по шагам из раздела Включение сканирования;
- доступен раннер, который выполняет задачи в образах контейнеров и может загрузить оба образа сканирования:
  - образ сканера, выбранный так, как описано в разделе Откуда берётся образ сканера; для образа в закрытом хранилище образов политика называет переменные CI/CD, в которых хранятся логин и пароль, в полях переменных реестра;
  - образ `gitleaks_secret_junit`, по умолчанию `ruby:3.3.10-slim`, переопределяется переменной окружения сервера `FE_SCANS_REPORT_CONVERTER_IMAGE`; политика не

передаёт этой задаче учётных данных хранилища образов, поэтому раннер загружает образ без них;

- проект, в котором лежит файл конфигурации или файл игнорируемых несоответствий, названный в политике, разрешает задачам сканируемого проекта читать его: в этом проекте в разделе «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD» указан сканируемый проект или его группа.

Включение сканирования

Чтобы включить сканирование в редакторе политик:

1. В проекте или группе выберите «Безопасность» → «Политики».
2. Нажмите «Новая политика».
3. На экране «Выберите тип политики» выберите «Политика выполнения сканирования».
4. Назовите политику. Новая политика уже содержит это сканирование: раздел «Обнаружение секретов (встроенный Gitleaks)» в блоке «Сканирования безопасности» раскрыт, флажок «Включить этот сканер» установлен. Если флажок снят, установите его.
5. Настройте группы раздела, как описано в разделе Что задаёт политика.
6. Нажмите «Настройка с запросом на слияние».

Это откроет запрос на слияние в проекте политик. Если проект политик ещё не привязан, редактор сначала создаст и привяжет его. Политика вступит в силу после того, как этот запрос на слияние будет принят.

Вместо этого действие можно добавить прямо в файл `.deckhouse_code/security-policies/policy.yml` в проекте политик:

```
scan_execution_policy:
  - name: Secret detection everywhere
    enabled: true
    rules:
      - type: pipeline
        branches: ["*"]
    actions:
      - scan: gitleaks_secret_detection
```

Что задаёт политика

В редакторе политик сканирование `gitleaks_secret_detection` — это раздел «Обнаружение секретов (встроенный Gitleaks)» в блоке «Сканирования безопасности». У раздела один флажок — «Включить этот сканер»; когда он установлен, раздел показывает все свои группы: образ сканера, приводит ли несоответствие к ошибке пайплайна, файл конфигурации, файл игнорируемых несоответствий и дополнительные параметры. Ниже групп — поле тегов раннера для обеих задач сканирования с подсказкой «Теги раннера через запятую». В свёрнутом виде рядом с заголовком раздела показано «Включён» или «Отключён».

Образ сканера

Поле	Что задаёт
«Образ сканера»	Ссылку на образ, в котором выполняется сканирование: путь из двух и более частей и тег или дайджест
«Переменная логина реестра»	Имя переменной CI/CD проекта или его группы, в которой хранится логин для хранилища образов, в виде `$ИМЯ` или `${ИМЯ}`. По умолчанию пусто
«Переменная пароля реестра»	Имя переменной CI/CD, в которой хранится пароль или токен для этого хранилища образов, в том же виде. По умолчанию пусто

Если оставить «Образ сканера» пустым, сканирование выполняется в образе, на который настроен Deckhouse Code, — он показан серым прямо в поле как подсказка; этот образ определяют шаги 2 и 3 из раздела Откуда берётся образ сканера. Для образа с Docker Hub адрес хранилища образов можно не указывать, как в `zricethezav/gitleaks:v8.30.1`. Неполная ссылка, например из одной части `gitleaks:v8.30.1`, не мешает сохранить политику: задача сканирования останавливается с `this scan was not given what the policy asks for: scanner image: …`.

Поля переменных нужны для образа в закрытом хранилище образов. Политика называет переменные, но не их значения. В файле политики это `FE_SCANS_REGISTRY_LOGIN_VARIABLE` и `FE_SCANS_REGISTRY_PASSWORD_VARIABLE` среди `variables` действия. Политика не сохраняется, если:

- заполнено только одно из двух полей;
- значение записано не в виде `$ИМЯ` или `${ИМЯ}`;
- имя начинается с `FE_`, `CI_` или `GITLAB_`.

Чтобы дать сканированию доступ к закрытому хранилищу образов:

1. В сканируемом проекте или его группе создайте две переменные CI/CD типа «Переменная (по умолчанию)»: одну с логином, другую с паролем или токеном. Включите маскирование для переменной с паролем. Пароль не может содержать `"` или `\`.
2. Укажите их имена в полях «Переменная логина реестра» и «Переменная пароля реестра».

Переменные читаются на стороне сканируемого проекта: групповая политика опирается на переменную группы, а переменная проекта с тем же именем занимает её место, и если её значение неверно, загрузка образа завершается с ошибкой. Учётные данные получает только `gitleaks_secret_scan`, и для образа, который поставляется с продуктом, они не используются.

Ронять пайплайн при находках

Флажок, по умолчанию установлен.

Значение	Что делает
Установлен (по умолчанию)	Несоответствие приводит к ошибке пайплайна
Снят	Несоответствие всё равно попадает в отчёт и доходит до отчёта об уязвимостях; пайплайн при этом не останавливается

В файле политики флажку соответствует `FE_SECURITY_FAIL_ON`: `info`, когда он установлен, и `report_only`, когда снят.

Каждое несоответствие получает уровень `CRITICAL`, поэтому здесь нет порога между уровнями, как в SAST-сканировании с помощью Opengrep: несоответствие либо приводит к ошибке пайплайна, либо нет.

Файл конфигурации

Поле	Что задаёт
«Путь к файлу»	Путь к собственной TOML-конфигурации сканера, которую задача сканирования читает из проекта, названного рядом. По умолчанию пусто; тогда сканирование использует только конфигурацию, встроенную в образ сканера
«Проект»	Репозиторий, из которого читается файл, в формате `группа/проект`, с его ветки по умолчанию. Нужен, как только заполнен «Путь к файлу»

В файле могут быть секция `[extend]`, списки исключений (`allowlist`) отдельных правил и выключенные правила. Набор, встроенный в образ сканера, остаётся в силе, пока файл не определит иначе; файл без секции `[extend]` читается как его расширение. Секция `[extend]` принимается, только если в ней есть `useDefault = true` и, кроме него, не более чем `disabledRules`. Задача останавливается до сканирования, если в `[extend]` есть что-то ещё, если в файле вторая секция `[extend]`, `allowlist` верхнего уровня или `disabledRules` вне `[extend]`.

Задача сканирования читает файл своим токеном задачи, поэтому проект, указанный в поле «Проект», должен дать сканируемому проекту доступ, описанный в разделе Предварительные требования.

Политика сохраняется, если поле «Проект» пустое или в нём указан сам сканируемый проект, но задача сканирования затем останавливается до сканирования. Собственный файл `.gitleaks.toml` сканируемого проекта сканирование не читает, он остаётся в рабочей копии.

Файл игнорируемых находок

Поле	Что задаёт
«Путь к файлу»	Путь к файлу с отпечатками несоответствий, по одному в строке, в том виде, в каком их печатает сканер. По умолчанию пусто; тогда не исключается ни одно несоответствие
«Проект»	Репозиторий, из которого читается файл, по тем же правилам, что и «Файл конфигурации» → «Проект»

Собственный файл `.gitleaksignore` сканируемого проекта удаляется из рабочей копии перед сканированием, поэтому файл, названный в этом поле, — единственное место, где несоответствие можно исключить. Комментарий `gitleaks:allow` на строке секрета тоже не учитывается: сканирование передаёт флаг `--ignore-gitleaks-allow`, поэтому комментарий рядом с секретом не заглушает несоответствие по нему.

Дополнительные параметры

Поле	Что задаёт
«Дополнительные параметры»	Флаги, передаваемые сканированию как есть, для всего, чего нет в этой форме

Примечание рядом с полем напоминает, что результаты сканирования публикуются как отчёт о безопасности, поэтому попадают в отчёт об уязвимостях и в системы, которые к нему подключил администратор.

Флаги, которые задаёт сама задача, отклоняются с указанием имени флага: те, что уводят или меняют отчёт, подменяют конфигурацию, исключают несоответствия (`--baseline-path`, `--ignore-gitleaks-allow`, `--enable-rule`), задают код выхода, маскирование, баннер, цвета или диагностику, называют коммиты для чтения (`--log-opts`), переключают сканер в другой режим (`--pre-commit`, `--staged`), задают платформу для ссылок на несоответствия (`--platform`) или печатают справку вместо сканирования. Уровни логирования `warn`, `error`, `fatal`, `panic` и `disabled` останавливают задачу: она читает лог сканера.

Откуда берётся образ сканера

Сервер определяет образ один раз, до старта задачи, и записывает его как готовое значение: настройки CI/CD самого сканируемого проекта не могут его переопределить. Порядок разрешения:

1. поле «Образ сканера» самой политики, если это полная ссылка на образ: путь из двух и более частей и тег или дайджест;
2. иначе образ, заданный в переменной окружения `FE_SCANS_GITLEAKS_IMAGE` сервера Deckhouse Code;
3. иначе образ этого релиза, `zricethezav/gitleaks:v8.30.1`.

`FE_SCANS_GITLEAKS_IMAGE` — переменная окружения серверных процессов, которые формируют пайплайны: веб-сервера и Sidekiq; администратор задаёт её в их окружении. Переменная CI/CD проекта или группы с тем же именем ни на что не влияет. В файле политики поле «Образ сканера» — это ключ `FE_SCANS_GITLEAKS_IMAGE` в `variables` действия.

Заполненное поле, которое не является полной ссылкой, не откатывается к шагам 2–3 для сканирования: задача останавливается с `this scan was not given what the policy asks for: scanner image: …` ещё до сканирования. Сама задача при этом запускается в образе из шага 2 или 3, поэтому этот образ должен загружаться.

Если у Deckhouse Code нет доступа в интернет, этот образ зеркалируют в своё хранилище образов, а зеркальную ссылку указывают в переменной окружения сервера `FE_SCANS_GITLEAKS_IMAGE` или в поле политики «Образ сканера». Если хранилище образов требует входа, заполните в политике поля переменных реестра.

Куда попадают результаты

Отчёт	Где смотреть
Тесты (JUnit)	Вкладка «Тесты» пайплайна и виджет «Сводка сканирований безопасности» запроса на слияние
Отчёт о безопасности (обнаружение секретов)	Страница «Отчёт об уязвимостях»
Несоответствия, переданные в DefectDojo	DefectDojo, если настроена интеграция с DefectDojo

Каждый тестовый случай назван `CRITICAL <правило> <описание> [<файл>:<строка>]`. Описание — то, которое Gitleaks даёт правилу; правило без описания названо своим идентификатором один раз. К повторяющемуся имени добавляется `#2`, `#3` и так далее. Описание несоответствия заканчивается отдельной строкой `Gitleaks rule: <правило>`.

Успешный тестовый случай `gitleaks scan` сообщает, что сканировалось (рабочая копия коммита или история), сколько отпечатков в файле игнорируемых несоответствий политики и какие дополнительные параметры были переданы, если были.

Имя задачи несёт короткое имя политики и ID её проекта политик, и DefectDojo сверяет тесты по этому имени целиком: переименование политики или перенос в другой проект политик начинает в DefectDojo новый тест.

Диагностика

Сообщение задачи	Что оно значит
`gitleaks is missing from this scan job image`	В выбранном образе нет `gitleaks` в `PATH`. Проверьте раздел Откуда берётся образ сканера
`this scan was not given what the policy asks for: <detail>`	Сервер отклонил часть политики ещё до старта задачи. В пояснении названо поле: `scanner image` (не полная ссылка), `blocking` (значение, отличное от `info` и `report_only`), `what the scan reads` (область, отличная от `working_copy` и `history`), `additional arguments` (флаг, который задаёт сама задача), `configuration file` или `ignore file` (путь без проекта, проект без пути либо неверно записанные путь или проект). Несколько причин разделяются точкой с запятой
`this policy names a configuration file in the project being scanned, which this scan never reads`	В поле «Файл конфигурации» → «Проект» указан сам сканируемый проект. Укажите проект, в котором лежит файл
`this policy names an ignore file in the project being scanned, which is the file this scan removes from the checkout`	В поле «Файл игнорируемых находок» → «Проект» указан сам сканируемый проект. Укажите проект, в котором лежит файл
`<project> does not let this project's jobs read it`	Проект с файлом не указывает сканируемый проект в разделе «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD»
`<path> is not on the default branch of <project>, or that project does not exist`	Путь или проект указаны неверно либо права доступа токена задач этого проекта не пускают сканируемый проект
`<path> in <project> is empty, and this policy names it`	Названный файл пуст. Заполните файл или очистите поле «Путь к файлу»
`this scanner's image carries no curl, python3 or wget, so a file in another project cannot be read`	В собственном образе сканера нет инструментов, которые загружают файл конфигурации или игнорируемых несоответствий
`the configuration this policy names would switch off the set this scan is for: …`	Файл, названный в «Файле конфигурации», содержит конструкцию, которая заменила бы

Сообщение задачи	Что оно значит
	или опустошила штатный набор правил или которую сканер не прочитал бы: вторую секцию `[extend]`, секцию или ключ `allowlist` верхнего уровня, ключ `extend`, записанный через точку, `[extend]` без `useDefault = true` или с ключом, отличным от `useDefault` и `disabledRules`, либо `disabledRules` вне `[extend]`
`the policy names rules this scanner does not have: …`	Запись `disabledRules` в файле конфигурации называет правило, которого нет у Gitleaks
`this scan was asked to log at <level>, which is quieter than this job can read back`	В «Дополнительных параметрах» задан уровень логирования `warn`, `error`, `fatal`, `panic` или `disabled`
`gitleaks exited with status <n>: the scan did not complete`	Сам сканер завершился с ошибкой; в логе задачи есть его строки об ошибке
`gitleaks exited with status 0 but wrote no report: the scan did not run`	Сканер остановился, ничего не просканировав и не вернув код ошибки. Прочитайте строки сканера в логе задачи
`ruby is missing from this converter job image`	В выбранном образе конвертера нет `ruby` в `PATH`
`The scan did not run: it left no report behind` (в тестовом случае `gitleaks scan`)	`gitleaks_secret_scan` завершился с ошибкой или был отменён; `gitleaks_secret_junit` не публикует несоответствий из неполного отчёта и завершается с ошибкой

Сканирование с помощью Trivy

Сканирование Trivy (`trivy`) проверяет проект сканером Trivy с открытым исходным кодом. Сканирование обязательное: политика выполнения сканирования добавляет его задачи в пайплайны проекта, и сканируемый проект не может отключить сканирование или изменить его настройки. Политика хранится в файле `.deckhouse_code/security-policies/policy.yml` отдельного проекта политик; `.gitlab-ci.yml` сканируемого проекта при этом не меняется. Сканирование выполняется отдельными задачами модулей: сканирование зависимостей, сканирование лицензий, обнаружение секретов, сканирование образов контейнеров и

сканирование кластеров Kubernetes. Все настройки Trivy задаются в политике; интеграции в разделе «Настройки» → «Интеграции» у этого сканирования нет.

Что делает сканирование

Модули, их задачи и отчёты:

Модуль	Задача	Отчёты	Что приводит задачу к ошибке
«Сканирование зависимостей»	`trivy_sca`	`dependency_scanning`, `cyclonedx`, `junit`	Несоответствие, достигшее общего порога блокировки
«Сканирование лицензий»	`trivy_license`	`cyclonedx`, `junit`	Несоответствие, достигшее общего порога блокировки
«Обнаружение секретов»	`trivy_secrets`	`secret_detection`, `junit`	Любой секрет (раздел «Обнаружение секретов»)
«Сканирование образа контейнера»	`trivy_container`, по одной задаче на образ	`container_scanning`, `junit`	Несоответствие, достигшее общего порога блокировки
«Сканирование кластеров Kubernetes»	`trivy_k8s`	`cluster_image_scanning`, `junit`	Несоответствие, достигшее общего порога блокировки

Эти правила действуют для каждого модуля и каждой его задачи:

- Все задачи выполняются на этапе пайплайна, который называет политика (раздел «Этап задач сканирования»), по умолчанию — `fe-security-scanner`. У задач нет `needs`, поэтому они ждут все более ранние этапы пайплайна и получают их артефакты.
- Имя задачи начинается с имени задачи из таблицы выше. Остальная часть имени называет политику, которая добавила задачу, а для модуля образов контейнеров — ещё и строку с образом. Например, `trivy_sca:trivy-everywhere-p42` — задача сканирования зависимостей политики `Trivy everywhere` из проекта политик с ID `42`. Имя одинаково в пайплайнах пуша и в пайплайнах по расписанию. Кириллица в имени политики транслитерируется: `Ночной скан` превращается в `nochnoi-skan`.
- Каждая политика, которая требует Trivy, получает собственные задачи. Это верно и тогда, когда Trivy требуют несколько политик одного `policy.yml`. Задачи различаются частью имени, которая называет политику, и сохраняют `stage`, `tags` и `variables` своей политики. Поэтому две политики с одинаковыми настройками сканируют дважды. Две политики одного

файла не могут иметь одинаковое имя. Два разных имени, которые дают одинаковую часть имени задачи, например `Trivy everywhere` и `trivy-everywhere`, сохраняются. Тогда их задачи получают одно имя, и выполняется только одна из них.
- Первая строка лога каждой задачи называет политику, которая её добавила: `This scan runs because the security policy "<name>" mandates it. Policies: <URL>`.
- Каждая задача выгружает артефакты с `when: always`, поэтому отчёты сохраняются, даже если задача завершилась с ошибкой.
- Перед запуском Trivy задача удаляет из окружения все переменные `TRIVY_*` и пишет об этом в лог: `Trivy: <NAME> is set in this job's environment and is ignored.`
- Сервер записывает настройки политики прямо в задачу, и переменные CI/CD сканируемого проекта с такими же именами их не меняют.
- Задачи прерываемые: если в проекте включена настройка «Автоматическая отмена избыточных пайплайнов», более новый пайплайн той же ветки отменяет их.
- Задачи всегда сканируют отправленный коммит. Настройки Git сканируемого проекта на это не влияют. Задачи загружают всю историю (`GIT_STRATEGY=fetch`, `GIT_DEPTH=0`) и удаляют из рабочей копии все неотслеживаемые файлы (`GIT_CLEAN_FLAGS=-ffdx`). Поэтому первая загрузка большого репозитория занимает больше времени. Политика может задать другие значения в `variables` действия. Исключение — задача `trivy_k8s`: она не читает репозиторий и ничего не загружает (`GIT_STRATEGY=none`).

Этап задач сканирования

Ключ `stage` задаёт этап пайплайна для всех сканирований политики. В редакторе политик ключу соответствует поле «Этап пайплайна». Этап один на всю политику. Если поле пустое, редактор не записывает ключ `stage`.

`stage` в политике	Где выполняются задачи сканирования
Не задан	На этапе `fe-security-scanner`, который политика добавляет в конец этапов проекта, перед `.post`
Этап, который перечислен в `stages:` проекта	На этом этапе, там, где его поставил проект
Этап, которого нет у проекта	На этом этапе, который политика добавляет в конец этапов проекта, перед `.post`

Для проекта со `stages: [build, test, deploy]` значение `stage: test` запускает сканирования на этапе `test`. Для того же проекта значение `stage: security` даёт этапы `build, test, deploy, security`. Чтобы поставить сканирования в другое место, перечислите этап в собственном `stages:` проекта в нужной позиции. Можно перечислить и `fe-security-scanner`.

Если политика называет этап, который есть у проекта, проект определяет, когда запускается обязательное сканирование. Проект может поставить этот этап раньше сборки. Пайплайн по

расписанию и пайплайн проекта без `.gitlab-ci.yml` содержат только сканирования, поэтому в них политика всегда добавляет этап сама.

Редактор отклоняет имя этапа:

- из одних пробелов;
- с пробелом в начале или в конце;
- длиннее 255 символов;
- `.pre` и `.post`.

Для `.post` редактор выводит ошибку `Сканирования не могут выполняться на этапе .post.`

Если политика с таким `stage` написана вручную, пайплайн её отбрасывает. Остальные политики файла продолжают работать.

Предварительные требования

Прежде чем сканирование сможет запуститься, убедитесь, что:

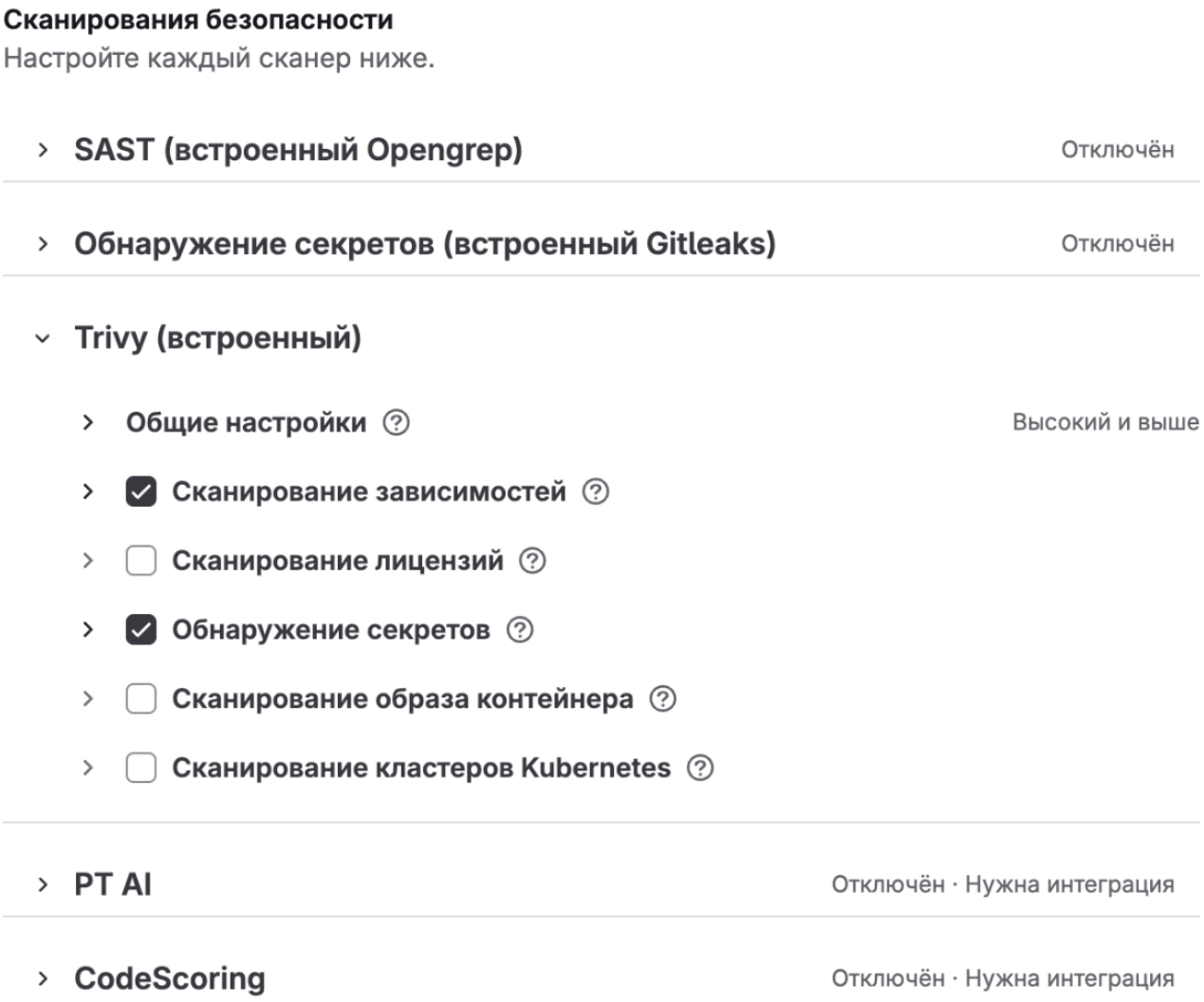
- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- проект политик привязан к проекту или к группе выше него;
- проекту доступен GitLab Runner с executor `docker`, который может загрузить образ сканирования (раздел «Образ сканирования и базы уязвимостей») и разрешает политику загрузки `always` (раздел «Политика загрузки образа на раннере»);
- раннер работает на `linux/amd64`, либо политика указывает образ сканирования, собранный под архитектуру раннера;
- для модуля образов контейнеров образ существует к моменту запуска сканирования: задача сборки отправляет его в хранилище образов или сохраняет как артефакт на более раннем этапе;
- для модуля кластеров к кластеру подключён агент для Kubernetes, и он разрешает сканируемому проекту доступ в своём `ci_access` (раздел «Доступ к кластеру»).

Включение сканирования

Политику можно составить в редакторе политик или написать в формате YAML.

Режим правил

1. Откройте проект или группу и перейдите в «Безопасность» → «Политики».
2. Нажмите «Новая политика», затем в карточке «Политика выполнения сканирования» нажмите одноимённую кнопку.
3. В поле «Имя» введите имя политики, например `Trivy everywhere`.
4. В разделе «Когда запускать сканирования» отметьте, какие пайплайны сканировать, например «На любом пайплайне».
5. Если политика должна запускать только Trivy, в разделе «Сканирования безопасности» снимите флажок «Включить этот сканер» у сканирования «Обнаружение секретов (встроенный Gitleaks)».
6. Раскройте «Trivy (встроенный)» и установите флажки модулей, которые нужно включить, например «Сканирование зависимостей» и «Обнаружение секретов».
7. Нажмите «Настройка с запросом на слияние».



Новая политика открывается с отмеченными условиями «Запросы на слияние» и «Ветка по умолчанию» и с включённым сканированием «Обнаружение секретов (встроенный Gitleaks)». Флажок «На любом пайплайне» снимает и блокирует флажки «Запросы на слияние», «Ветка по умолчанию» и «По имени ветки». В блоке «Предпросмотр .yaml» справа от формы редактор показывает политику, которую запишет. Свёрнутый блок сканера показывает в заголовке, запускает ли политика сканер:

- «Отключён», если сканер выключен;
- включённые модули у «Trivy (встроенный)»;
- «Включён» у сканера без модулей.

У сканера без настроенной интеграции заголовок дополнительно показывает «Нужна интеграция».

Если к проекту ещё не привязан проект политик, при сохранении редактор создаёт его и привязывает: для проекта — `<путь проекта>-security-policy-project` в той же группе, для группы — `<путь группы>-security-policy-project` внутри неё. Затем редактор открывает в проекте политик запрос на слияние с политикой; политика начинает действовать, когда этот запрос на слияние принят.

Режим .yaml

Переключите редактор в «Режим .yaml» над формой или отредактируйте файл `.deckhouse_code/security-policies/policy.yml` в проекте политик напрямую.

Режим правил | Режим .yaml

```yaml
scan_execution_policy:
  - name: Trivy everywhere
    description: ''
    enabled: true
    rules:
      - type: pipeline
        branch_type: all
    actions:
      - scan: trivy
        variables:
          FE_SCANS_TRIVY_SCAN_IMAGE: 'registry.flant.com/deckhouse/code/gitlab-custom/trivy-with-db:0.74.0'
          FE_SCANS_TRIVY_FAIL_ON: high
          FE_SCANS_TRIVY_SCA: 'true'
          FE_SCANS_TRIVY_DETECTION_PRIORITY: precise
          FE_SCANS_TRIVY_SECRETS: 'true'

```

Добавьте действие `scan: trivy`. Каждый модуль включается собственной переменной со значением `'true'`: `FE_SCANS_TRIVY_SCA`, `FE_SCANS_TRIVY_LICENSE`, `FE_SCANS_TRIVY_SECRETS`, `FE_SCANS_TRIVY_CONTAINER`, `FE_SCANS_TRIVY_K8S`. Схема политики отклоняет действие `trivy`, если ни одна из них не названа.

```
scan_execution_policy:
  - name: Trivy everywhere
    enabled: true
    rules:
      - type: pipeline
        branch_type: all
    actions:
      - scan: trivy
        variables:
          FE_SCANS_TRIVY_SCAN_IMAGE: 'registry.example.com/trivy-with-db:0.74.0'
          FE_SCANS_TRIVY_FAIL_ON: high
          FE_SCANS_TRIVY_SCA: 'true'
          FE_SCANS_TRIVY_DETECTION_PRIORITY: precise
          FE_SCANS_TRIVY_SECRETS: 'true'
```

`FE_SCANS_TRIVY_SCAN_IMAGE` обязательна: политику без неё сохранить нельзя, и ошибка называет переменную. Политика, написанная вручную, может не указывать `FE_SCANS_TRIVY_FAIL_ON` и `FE_SCANS_TRIVY_DETECTION_PRIORITY`: тогда сканирование выполняется с порогом `high` и приоритетом `precise`.

Политика, которую вы сами добавили в проект политик, действует, только когда этот проект политик привязан: к проекту — кнопкой «Редактировать проект политик» в разделе «Безопасность» → «Политики», к группе — в разделе «Настройки» → «Политика безопасности».

Каждый пайплайн, который подходит под правила политики, получает задачи включённых модулей; в примере выше это `trivy_sca` и `trivy_secrets`.

Флажок модуля записывает в политику переменную, которая включает модуль, и значения по умолчанию:

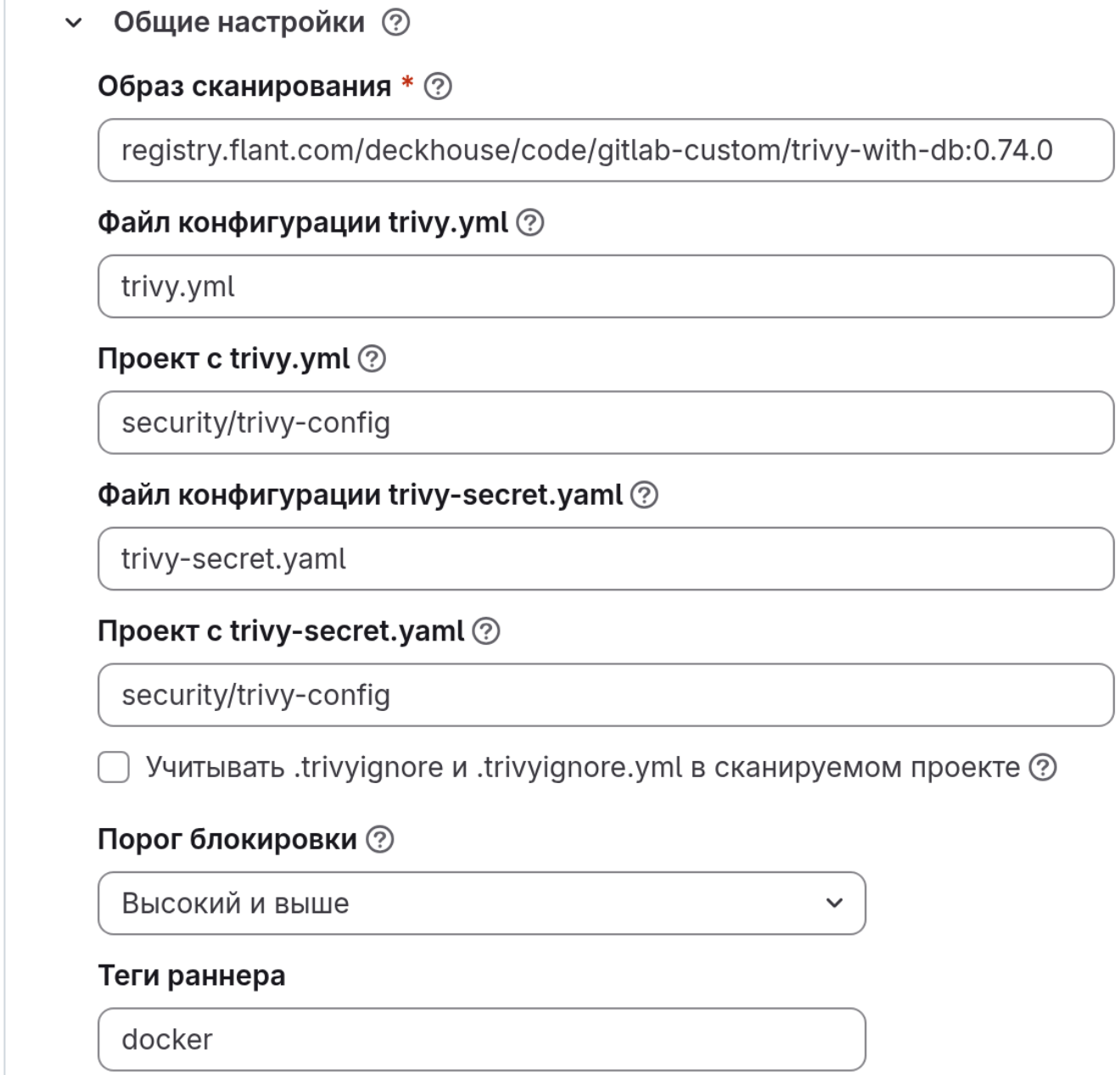
Флажок	Что записывается в политику
«Сканирование зависимостей»	`FE_SCANS_TRIVY_SCA: 'true'`, `FE_SCANS_TRIVY_DETECTION_PRIORITY: precise`
«Сканирование лицензий»	`FE_SCANS_TRIVY_LICENSE: 'true'`
«Обнаружение секретов»	`FE_SCANS_TRIVY_SECRETS: 'true'`
«Сканирование образа контейнера»	`FE_SCANS_TRIVY_CONTAINER: 'true'`, `FE_SCANS_TRIVY_CONTAINER_IMAGES: $CI_APPLICATION_REPOSITORY: $CI_APPLICATION_TAG`, `FE_SCANS_TRIVY_CONTAINER_DETECTION_PRIORITY: precise`
«Сканирование кластеров Kubernetes»	`FE_SCANS_TRIVY_K8S: 'true'`

Флажок первого включённого модуля дополнительно заполняет поле «Образ сканирования» блока «Общие настройки» образом текущего релиза и записывает `FE_SCANS_TRIVY_FAIL_ON: high`. Отмеченный модуль раскрывает свой блок настроек. У каждого блока есть подсказка рядом с заголовком.

Общие настройки

Блок «Общие настройки» применяется к каждому включённому модулю и не имеет собственного флажка.

Режим правил



Поля «Проект с trivy.yml» и «Проект с trivy-secret.yaml» появляются, только когда заполнено поле файла над ними; очистка поля файла убирает проект из политики.

Режим .yaml

Эти поля — переменные действия `trivy`, а «Теги раннера» — список `tags` этого действия:

```
actions:
  - scan: trivy
    variables:
      FE_SCANS_TRIVY_SCAN_IMAGE: 'registry.example.com/trivy-with-db:0.74.0'
      FE_SCANS_TRIVY_CONFIG_FILE: trivy.yml
      FE_SCANS_TRIVY_CONFIG_PROJECT: security/trivy-config
      FE_SCANS_TRIVY_SECRET_CONFIG: trivy-secret.yaml
      FE_SCANS_TRIVY_SECRETS_RULES_PROJECT: security/trivy-config
      FE_SCANS_TRIVY_FAIL_ON: high
      FE_SCANS_TRIVY_SCA: 'true'
    tags:
      - docker
```

Поле	Переменная	Что задаёт
«Образ сканирования»	`FE_SCANS_TRIVY_SCAN_IMAGE`	Образ, в котором выполняется задача каждого модуля, со встроенными базами уязвимостей
«Переменная логина реестра»	`FE_SCANS_REGISTRY_LOGIN_VARIABLE`	Переменная CI/CD в виде `$ИМЯ`, в которой хранится логин для хранилища образа сканирования. Задаётся вместе с переменной пароля или не задаётся совсем; см. Учётные данные хранилища образов для сканера
«Переменная пароля реестра»	`FE_SCANS_REGISTRY_PASSWORD_VARIABLE`	Переменная CI/CD в виде `$ИМЯ`, в которой хранится пароль или токен для этого хранилища
«Файл конфигурации trivy.yml»	`FE_SCANS_TRIVY_CONFIG_FILE`	Путь к файлу конфигурации Trivy, общему для всех модулей. Пусто — файл не передаётся
«Проект с trivy.yml»	`FE_SCANS_TRIVY_CONFIG_PROJECT`	Проект, из которого читается trivy.yml, в формате `group/project`. Пусто — файл читается из сканируемого репозитория
«Файл конфигурации trivy-secret.yaml»	`FE_SCANS_TRIVY_SECRET_CONFIG`	Путь к правилам и спискам исключений поиска секретов. Пусто — встроенные правила Trivy
«Проект с trivy-secret.yaml»	`FE_SCANS_TRIVY_SECRETS_RULES_PROJECT`	Проект, из которого читается trivy-secret.yaml. Пусто — файл читается из сканируемого репозитория
«Учитывать .trivyignore и .trivyignore.yml в сканируемом проекте»	`FE_SCANS_TRIVY_USE_IGNOREFILE`	По умолчанию выключено. Действует на все модули,

Поле	Переменная	Что задаёт
		кроме модуля кластеров. Подробнее — под таблицей
«Порог блокировки»	`FE_SCANS_TRIVY_FAIL_ON`	Порог задач сканирования зависимостей, лицензий, образов контейнеров и кластеров, по умолчанию «Высокий и выше» (раздел «Блокировка»)
«Теги раннера»	`tags` действия	Теги раннера через запятую; применяются к задаче каждого модуля

«Образ сканирования» редактор заполняет образом текущего релиза, `trivy-with-db:0.74.0` из хранилища образов продукта, и политика продолжает использовать этот образ после обновления продукта, пока кто-нибудь не отредактирует поле. Образа по умолчанию на сервере нет. Пока включён хотя бы один модуль, поле обязательно, и форму с пустым полем сохранить нельзя. Сервер не сохраняет политику без `FE_SCANS_TRIVY_SCAN_IMAGE`, даже если она написана вручную. Ошибка называет переменную. Политика без образа, которая уже лежит в проекте политик, продолжает действовать: её остальные сканирования выполняются. Вместо задач Trivy пайплайн получает одну задачу `trivy_no_image`. Она сразу завершается с ошибкой и называет `FE_SCANS_TRIVY_SCAN_IMAGE` и поле «Образ сканирования». Значение — полная ссылка на образ с тегом, например `registry.example.com/trivy-scan:0.74.0`, либо с дайджестом (`@sha256:` и 64 шестнадцатеричных символа). Ссылка без тега или дайджеста и ссылка с символом `$` отклоняются. Сервер записывает образ в задачу, поэтому сканируемый проект не может его изменить.

Флажок «Учитывать .trivyignore и .trivyignore.yml в сканируемом проекте» по умолчанию выключен: задача передаёт пустой список игнорирования, и `.trivyignore` сканируемого проекта не может скрыть несоответствие. Если флажок установлен, задача ищет в корне репозитория сначала `.trivyignore.yml`, затем `.trivyignore.yaml` и применяет первый найденный файл; если не найден ни один, Trivy читает `.trivyignore`. Лог называет применённый файл и сообщает, когда `.trivyignore` пропущен из-за использования YAML-файла. Модуль образов контейнеров тоже читает файл игнорирования из репозитория, а не из образа. Модуль кластеров никогда не читает файл игнорирования, при любом состоянии флажка: его задача всегда передаёт `--ignorefile /dev/null`. Включённый флажок позволяет разработчику сканируемого проекта скрыть несоответствие без ведома автора политики. Формат описан в справочнике Trivy по фильтрам.

Настройки каждого модуля

Каждый блок модуля содержит собственные поля «Пропускать каталоги», «Пропускать файлы» и «Дополнительные аргументы»; значение действует только на задачу этого модуля, потому что

модули выполняются отдельными задачами. У модуля кластеров есть «Дополнительные аргументы» и собственное поле «Тайм-аут», а списков пропуска нет (раздел «Сканирование кластеров Kubernetes»). В имени переменной `<MODULE>` — это `SCA`, `LICENSE`, `SECRETS` или `CONTAINER`, в зависимости от модуля, которому принадлежит поле:

Поле	Переменная	Что задаёт
«Пропускать каталоги»	`FE_SCANS_TRIVY_<MODULE>_SKIP_DIRS`	По одному каталогу или шаблону в строке. Для модуля образов контейнеров — пути внутри файловой системы образа
«Пропускать файлы»	`FE_SCANS_TRIVY_<MODULE>_SKIP_FILES`	По одному файлу или шаблону в строке. Для модуля образов контейнеров — пути внутри образа
«Дополнительные аргументы»	`FE_SCANS_TRIVY_<MODULE>_EXTRA_ARGS`	Флаги, которые передаются в запуск Trivy этого модуля как есть, для флагов, которых нет в форме

«Дополнительные аргументы» ставятся в командной строке перед флагами формы, поэтому поле формы имеет приоритет над тем же флагом, написанным здесь. Файл конфигурации передаётся последним, поэтому «Дополнительные аргументы» не могут направить сканирование на другой файл конфигурации. Флаг, который сужает отчёт, например `--severity`, приводит к тому, что сканирование сообщает меньше, чем нашло. Флаг, заданный формой, полностью заменяет одноимённый ключ в trivy.yml: `scan.skip-files` в trivy.yml не действует, если заполнено поле «Пропускать файлы».

Тайм-аут модуля задаётся флагом `--timeout` в поле «Дополнительные аргументы», например `--timeout 15m`; подсказка поля приводит его как пример. Ключ `timeout` в общем trivy.yml задаёт тайм-аут для всех модулей, а `--timeout` его переопределяет. Модуль кластеров — исключение. Он всегда передаёт `--timeout` из своего поля «Тайм-аут». Если поле пустое, модуль передаёт `20m`. Ключ `timeout` из trivy.yml на модуль не действует. Если не задано ни то ни другое, действует значение Trivy по умолчанию, 5 минут, которого может не хватить большому репозиторию. Некорректное значение не проверяют ни форма, ни схема политики: Trivy отклоняет его при запуске, и задача завершается с ошибкой (раздел «Trivy отклоняет значение тайм-аута»). Примеры в разделах модулей ниже показывают эти поля заполненными.

Сканирование зависимостей

Модуль находит в рабочей копии репозитория пакеты операционной системы и зависимости приложения с известными уязвимостями. Собственные поля модуля:

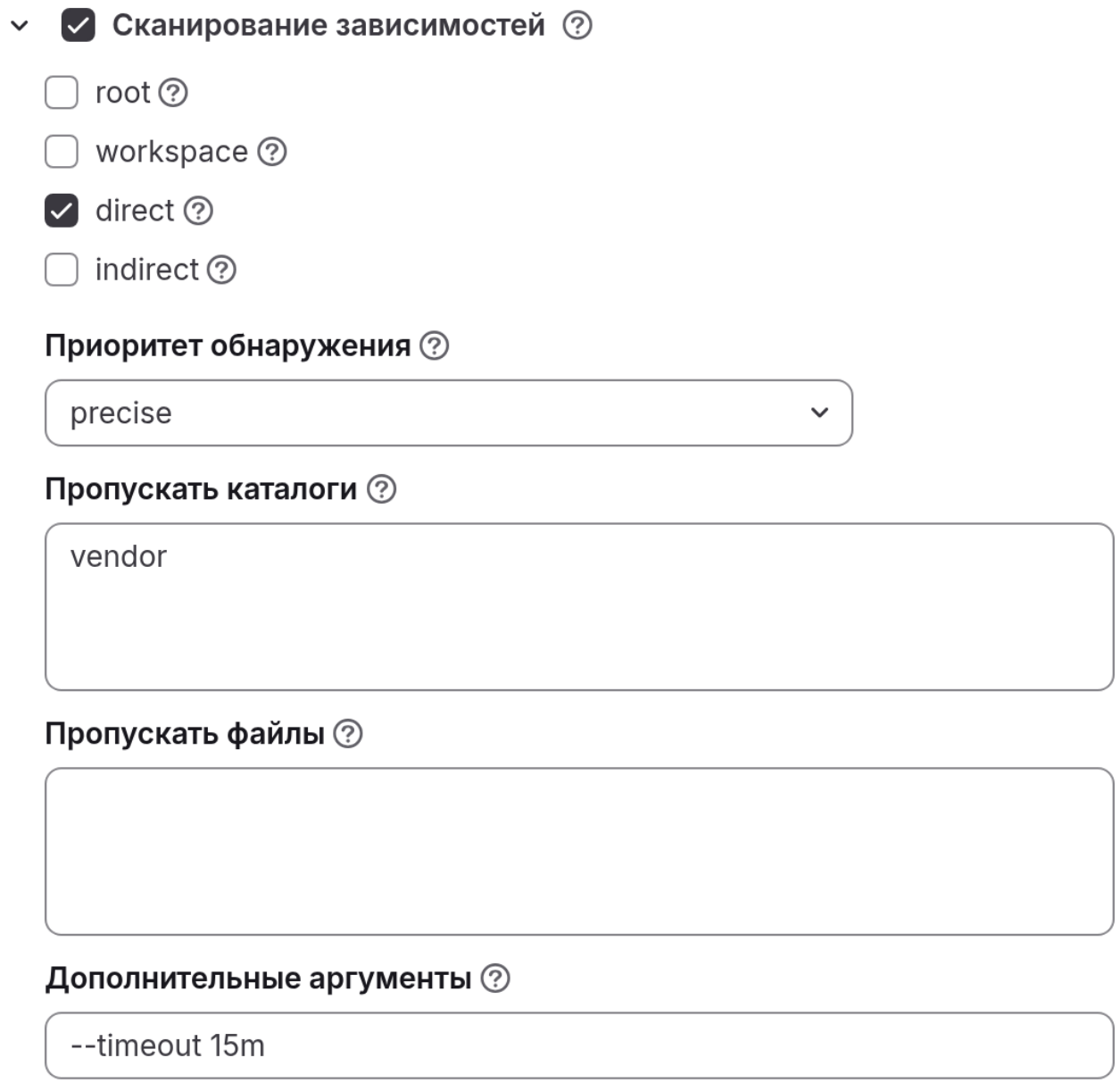
Поле	Переменная	Что задаёт
«Связи пакетов»	`FE_SCANS_TRIVY_PKG_RELATIONSHIPS`	Какие связи зависимостей попадают в отчёт, через запятую из `root`, `workspace`, `direct`, `indirect`. Пусто — все
«Приоритет обнаружения»	`FE_SCANS_TRIVY_DETECTION_PRIORITY`	`precise` (по умолчанию у Trivy) сообщает только то, что Trivy относит к зависимости уверенно; `comprehensive` сообщает больше, вместе с большим числом ложных срабатываний

В блоке у флажков `root`, `workspace`, `direct` и `indirect` нет подписи «Связи пакетов»: они стоят первыми, сразу под заголовком блока. У каждого значения есть подсказка:

Значение	Что означает
`root`	Сама цель сканирования
`workspace`	Объявлено в манифесте одного из пакетов, когда несколько пакетов лежат в одном репозитории
`direct`	Объявлено в манифесте самого проекта
`indirect`	Подтянуто другим пакетом

Несколько значений записываются через запятую, например `direct,indirect`.

Режим правил



В примере в отчёт попадают только прямые зависимости, каталог `vendor` пропускается, а тайм-аут увеличен до 15 минут.

Режим .yaml

```
actions:
  - scan: trivy
    variables:
      FE_SCANS_TRIVY_SCA: 'true'
      FE_SCANS_TRIVY_PKG_RELATIONSHIPS: direct
      FE_SCANS_TRIVY_DETECTION_PRIORITY: precise
      FE_SCANS_TRIVY_SCA_SKIP_DIRS: vendor
      FE_SCANS_TRIVY_SCA_EXTRA_ARGS: '--timeout 15m'
```

Многострочное поле, например «Пропускать каталоги» с несколькими каталогами, записывается в YAML как блок с одним значением в строке (`|-`).

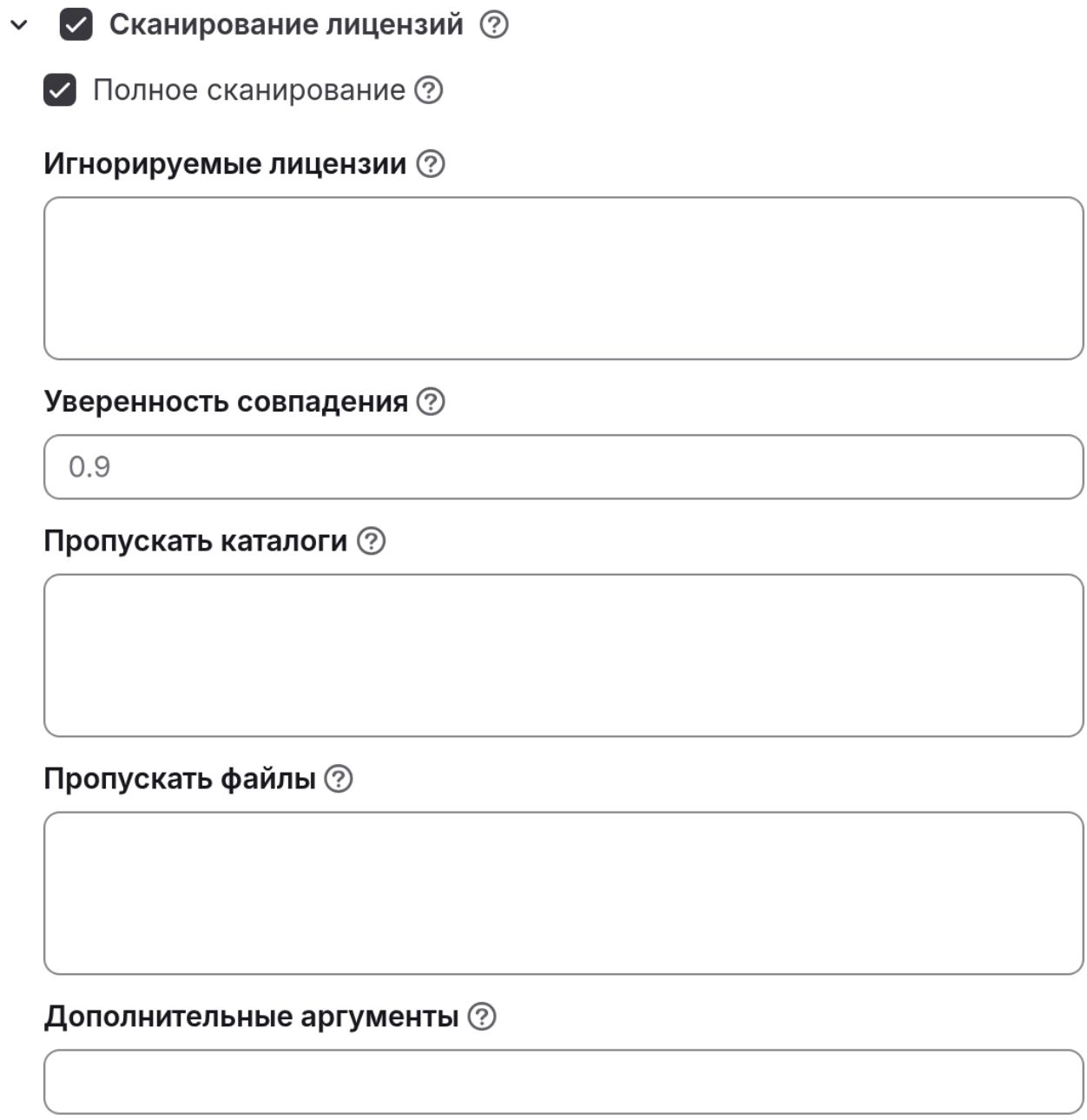
Задача запускает Trivy дважды: один раз — чтобы записать SBOM (Software Bill of Materials) в формате CycloneDX, `gl-sbom.cdx.json`, второй — чтобы найти уязвимости. Из SBOM данные берут страницы «Список зависимостей» и «Соответствие лицензий». «Связи пакетов» и «Приоритет обнаружения» действуют только на поиск уязвимостей, поэтому SBOM всегда содержит всё дерево зависимостей, а пакет, отфильтрованный из отчёта об уязвимостях, остаётся в списке зависимостей. Неудачный запуск, который строит SBOM, только выводит предупреждение и не влияет на найденные уязвимости. Несоответствия ниже порога блокировки остаются в отчёте: порог решает только исход задачи. Сканирование ничего не загружает и читает базы уязвимостей, встроенные в образ сканирования. Если одной из баз в образе нет, задача останавливается до сканирования (раздел «Диагностика»). Зависимости, которые задача сборки проекта передала через `cache:` или `artifacts:`, оказываются в рабочем каталоге задачи и тоже сканируются.

Сканирование лицензий

Модуль определяет категории лицензий пакетов, которые находит Trivy. Собственные поля модуля:

Поле	Переменная	Что задаёт
«Полное сканирование»	`FE_SCANS_TRIVY_LICENSE_FULL`	По умолчанию выключено. Включено: Trivy читает тексты лицензий и заголовки файлов в дереве исходного кода вместо метаданных пакетов. Медленнее и сообщает больше, включая ложные совпадения
«Игнорируемые лицензии»	`FE_SCANS_TRIVY_LICENSE_IGNORED`	По одному идентификатору SPDX в строке, например `MIT` или `Apache-2.0`: несоответствия по ним отбрасываются
«Уверенность совпадения»	`FE_SCANS_TRIVY_LICENSE_CONFIDENCE`	Число от 0 до 1: насколько точно текст лицензии должен совпасть при полном сканировании. Значение Trivy по умолчанию — `0.9`; меньшее значение сообщает о более слабых совпадениях

Режим правил



В примере включено «Полное сканирование».

Режим .yaml

```
actions:
  - scan: trivy
    variables:
      FE_SCANS_TRIVY_LICENSE: 'true'
      FE_SCANS_TRIVY_LICENSE_FULL: 'true'
```

Каждая лицензия попадает в одну из категорий, и у каждой категории своя критичность:

Категория	Критичность
`forbidden`	Критическая
`restricted`	Высокая
`reciprocal`	Средняя
`notice`, `permissive`, `unencumbered`	Низкая

Лицензия, которую Trivy не распознаёт, считается неизвестной и не приводит задачу к ошибке ни при каком пороге. При пороге по умолчанию «Высокий и выше» лицензия категории `forbidden` или `restricted` приводит задачу к ошибке. Категория лицензии задаётся в trivy.yml (раздел «Категории лицензий в trivy.yml») и описана в справочнике Trivy по сканеру лицензий.

Задача выгружает отчёт JUnit и SBOM `gl-sbom-license.cdx.json`. Отчёта безопасности у неё нет: проверка лицензии отражается как результат теста, и несоответствия по лицензиям видны только на вкладке «Тесты». Страница «Соответствие лицензий» и столбец «Лицензия» страницы «Список зависимостей» объединяют лицензии компонента из обоих SBOM: SBOM модуля зависимостей берёт лицензии из метаданных пакетов, а в SBOM этого модуля «Полное сканирование» добавляет лицензии из текстов исходного кода. Поэтому SBOM модуля лицензий только добавляет лицензии к тому, что показывают страницы. Модуль лицензий не использует базы уязвимостей.

Лицензии неустановленных зависимостей

Найдёт ли Trivy лицензию зависимости без установки пакетов, зависит от формата lock-файла. Одни lock-файлы несут лицензию каждого пакета; другие — только имя и версию, и тогда Trivy читает лицензию из метаданных установленного пакета: `node_modules/<package>/package.json` для npm, `.dist-info/METADATA` для Python, установленный gemspec для Ruby. Лицензии Go Trivy всегда берёт из кеша модулей.

Lock-файл	Лицензия без установки пакетов
`package-lock.json` с `lockfileVersion` 2 или 3	Да
`composer.lock`	Да
`package-lock.json` с `lockfileVersion` 1	Нет
`yarn.lock` версии 1	Нет
`Cargo.lock`	Нет
`Gemfile.lock`	Нет
`poetry.lock`	Нет

То же самое верно для лицензий в SBOM, который пишет модуль зависимостей: там, где lock-файл не несёт лицензию и пакеты не установлены, столбец «Лицензия» списка зависимостей остаётся пустым.

Задача сканирования не устанавливает зависимости проекта. Устанавливайте их на более раннем этапе в `.gitlab-ci.yml` сканируемого проекта, в его собственном образе, и передавайте дальше через `cache:` или `artifacts:`:

```
install:
  stage: build
  image: node:20
  script:
    - npm ci
  cache:
    key:
      files: [package-lock.json]
    paths:
      - node_modules/
```

Проект без этапа сборки может включить «Полное сканирование»: оно читает тексты лицензий и заголовки файлов в дереве исходного кода, ничего не устанавливая. Отчёт тогда показывает лицензии репозитория и стороннего кода, скопированного в дерево, а лицензии зависимостей, которые lock-файл только называет, остаются неизвестными.

Категории лицензий в trivy.yml

Категории — `forbidden`, `restricted`, `reciprocal`, `notice`, `permissive` и `unencumbered`; у каждой лицензии, которую распознаёт Trivy, есть встроенная категория (у MIT это `notice`). Если перечислить лицензию под категорией в trivy.yml, она добавляется в эту категорию и остаётся во встроенной. Лицензия оказывается сразу в двух категориях, и Trivy заново выбирает одну из них

при каждом запуске, независимо от того, менялся ли код или конфигурация. Поэтому проверка лицензии на неизменном коде то проходит, то нет: при пороге по умолчанию «Высокий и выше» запуск, выбравший `forbidden` (критическая), приводит задачу к ошибке, а запуск, выбравший `notice` (низкая), завершается успешно.

Чтобы это исправить, переопределите также встроенную категорию лицензии. Тогда лицензия останется ровно в одной категории. Например, для MIT:

```
license:
  forbidden:
    - MIT
  notice: []
```

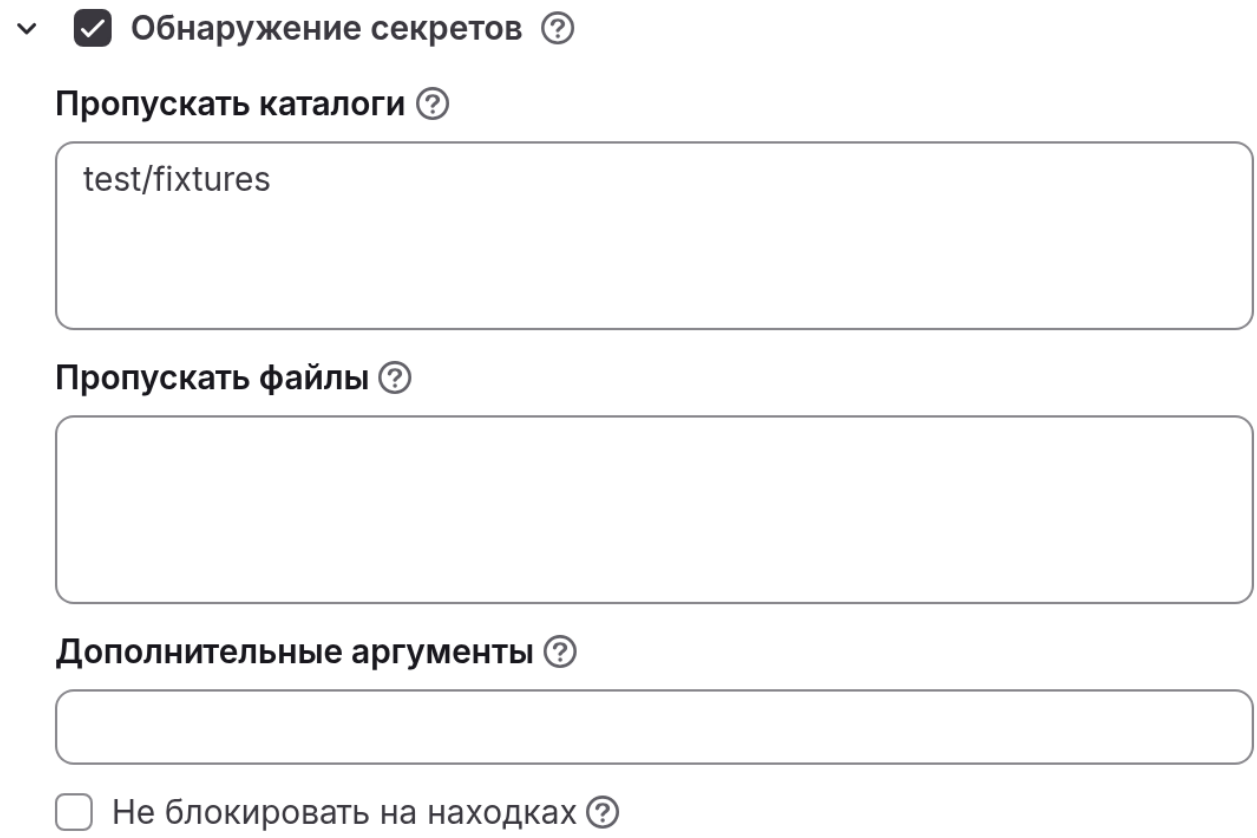
Результат стабилен, если лицензия перечислена под своей же встроенной категорией (`license.notice: [MIT]`), или если ни один просканированный пакет не несёт эту лицензию.

Обнаружение секретов

Модуль ищет в репозитории секреты, попавшие в коммиты. Собственное поле модуля:

Поле	Переменная	Что задаёт
«Не блокировать на находках»	`FE_SCANS_TRIVY_SECRETS_REPORT_ONLY`	По умолчанию выключено. Включено: несоответствия попадают в отчёт, а задача завершается успешно

Режим правил



В примере пропускается каталог `test/fixtures`: иначе тестовые ключи в нём были бы отмечены как утечка секретов.

Режим .yaml

```
actions:
  - scan: trivy
    variables:
      FE_SCANS_TRIVY_SECRETS: 'true'
      FE_SCANS_TRIVY_SECRETS_SKIP_DIRS: test/fixtures
```

Любой секрет приводит задачу к ошибке независимо от его критичности. Флажок «Не блокировать на находках» присваивает задаче вместо этого порог `report_only`. Модуль читает только файлы рабочей копии на сканируемом коммите и не просматривает историю git, поэтому секрет, добавленный в одном коммите и убранный в следующем, не находится. Лог сообщает об этом: `Trivy: searching the checked-out tree for secrets; git history is not read.` Модуль не

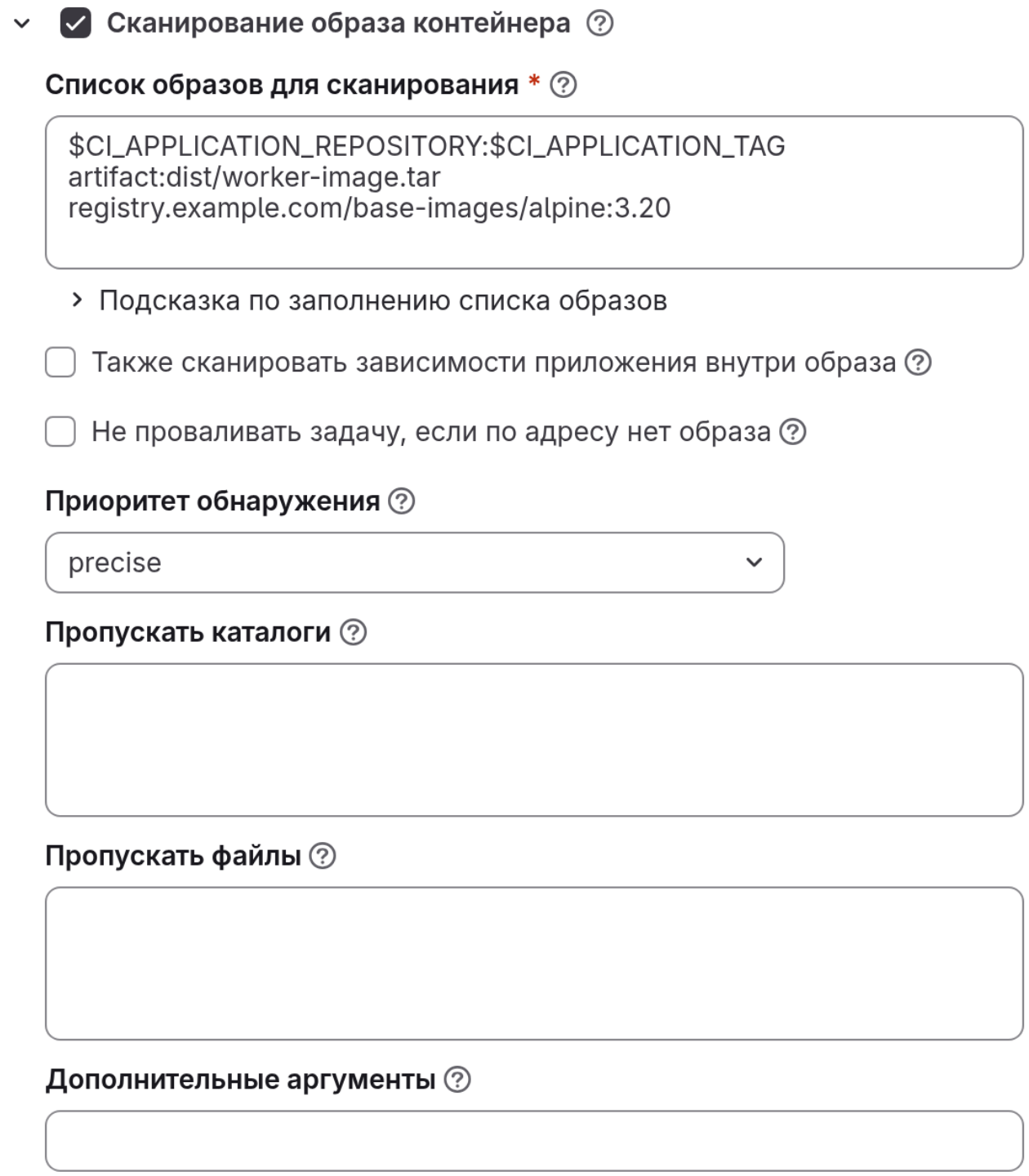
использует базы уязвимостей и работает на раннере без доступа в интернет. Модуль читает trivy.yml и trivy-secret.yaml, названные в блоке «Общие настройки» (раздел «Файлы конфигурации»).

Сканирование образов контейнеров

Модуль сканирует образ, собранный пайплайном: пакеты операционной системы его базового образа и всё, что добавлено во время сборки. Несоответствия попадают в отчёт об уязвимостях как сканирование контейнера. Собственные поля модуля:

Поле	Переменная	Что задаёт
«Список образов для сканирования»	`FE_SCANS_TRIVY_CONTAINER_IMAGES`	Обязательное. По одному образу в строке, не больше 20 строк (раздел «Список образов для сканирования»)
«Также сканировать зависимости приложения внутри образа»	`FE_SCANS_TRIVY_CONTAINER_APP_DEPS`	По умолчанию выключено: сканируются только пакеты операционной системы образа
«Не проваливать задачу, если по адресу нет образа»	`FE_SCANS_TRIVY_CONTAINER_ALLOW_MISSING`	По умолчанию выключено: отсутствующий образ приводит задачу к ошибке
«Приоритет обнаружения»	`FE_SCANS_TRIVY_CONTAINER_DETECTION_PRIORITY`	`precise` или `comprehensive`, как в модуле зависимостей

Режим правил



В примере заданы образ, который отправил пайплайн, образ, который задача сборки сохранила как артефакт, и базовый образ из другого хранилища. Блок «Подсказка по заполнению списка образов» под полем объясняет формы строк и допустимые переменные.

Режим .yaml

```
actions:
  - scan: trivy
    variables:
      FE_SCANS_TRIVY_CONTAINER: 'true'
      FE_SCANS_TRIVY_CONTAINER_DETECTION_PRIORITY: precise
      FE_SCANS_TRIVY_CONTAINER_IMAGES: |-
        $CI_APPLICATION_REPOSITORY:$CI_APPLICATION_TAG
        artifact:dist/worker-image.tar
        registry.example.com/base-images/alpine:3.20
```

Задача читает образ из хранилища образов и никогда — из локального демона Docker на раннере: она выполняет `trivy image --image-src remote`. Образ, собранный на раннере и никуда не отправленный, задача не найдёт: отправьте его в хранилище или сохраните как артефакт.

Каждая строка поля «Список образов для сканирования» сканируется отдельной задачей со своим статусом. Имя задачи — `trivy_container:<номер строки>-<цель>` с суффиксом политики: `<номер строки>` считается с 1, а `<цель>` — строка как она написана в политике, без подстановки переменных, в нижнем регистре, с заменой остальных символов на `-`, обрезанная до 40 символов. Имя остаётся одинаковым от коммита к коммиту, поэтому «Итоги тестирования» запроса на слияние сопоставляют наборы тестов базовой и новой версии.

Задача ищет в образе и уязвимости, и секреты, независимо от того, включён ли модуль обнаружения секретов. Секрет, попавший в образ, оценивается по общему порогу блокировки, как и уязвимости образа (раздел «Блокировка»).

«Также сканировать зависимости приложения внутри образа» по умолчанию выключено: модуль зависимостей уже сообщает о зависимостях приложения из репозитория, и сканирование в обоих местах сообщило бы об одной уязвимости дважды. Включите этот флажок, чтобы поймать то, что попало в образ в обход репозитория.

«Не проваливать задачу, если по адресу нет образа» касается только отсутствующего образа: хранилище ответило `MANIFEST_UNKNOWN` или `NAME_UNKNOWN`, либо файл `artifact:` не найден на диске. Отказ авторизации, недоступное хранилище, тайм-аут и любая другая ошибка всё равно приводят задачу к ошибке. Включайте этот флажок для веток, которые не собирают образ, например при изменениях в документации.

Лог задачи называет образ, который был фактически прочитан, вместе с его дайджестом: `trivy_container: scanned <image>, digest sha256:…, <OS>`. Для артефакта он выводит путь к файлу и ID образа.

Для этого модуля «Пропускать каталоги» и «Пропускать файлы» — пути внутри образа, а trivy.yml, trivy-secret.yaml и файл игнорирования берутся из репозитория: названного проекта или

сканируемого. Уязвимость, которую дистрибутив пометил как `not_affected`, `will_not_fix`, `end_of_life` или `fix_deferred`, отражается как пропущенная и не учитывается порогом; то же верно для модуля зависимостей (раздел «Блокировка»). Уязвимый пакет базового образа устраняется пересборкой на более новом базовом образе; об этом задача сообщает, когда завершается с ошибкой.

Список образов для сканирования

Строка поля «Список образов для сканирования» имеет одну из следующих форм:

Форма	Подробности
Адрес, собранный из перечисленных ниже переменных	Например, `$CI_APPLICATION_REPOSITORY:$CI_APPLICATION_TAG`
`artifact:<путь>`	Образ, который задача сборки сохранила как артефакт: путь внутри рабочей копии репозитория, относительный, без `..` и без шаблонов. Строка, начинающаяся с `artifact:`, всегда читается как путь, поэтому образ, который действительно называется `artifact`, указывайте полностью: `docker.io/library/artifact:1.0`
Полная ссылка на образ с тегом или дайджестом	Из любого хранилища образов, например `postgres:15-alpine` или `registry.example.com/group/project@sha256:<DIGEST>`

Строка без тега или дайджеста отклоняется при сохранении политики; так же отклоняется одиночный `$CI_REGISTRY_IMAGE`. Флажок модуля записывает в поле строку `$CI_APPLICATION_REPOSITORY:$CI_APPLICATION_TAG`: по этому адресу образ отправляет сборка Auto DevOps. Те же сведения, что в этом разделе, редактор показывает под полем, в блоке «Подсказка по заполнению списка образов». Редактор помечает больше 20 строк и не даёт сохранить: «Не больше 20 строк, сейчас .»

В адресе разрешены только перечисленные ниже переменные; любая другая `$ПЕРЕМЕННАЯ` отклоняется при сохранении политики.

Переменная	Значение
`$CI_REGISTRY_IMAGE`	Хранилище образов контейнеров сканируемого проекта, например `registry.example.com/group/project`
`$CI_COMMIT_REF_SLUG`	Ветка или тег в виде тега образа: `feature/foo` становится `feature-foo`. Для сборок, которые называют тег образа по ветке
`$CI_COMMIT_SHA`	Полный SHA коммита пайплайна. Для сборок, которые называют тег образа по коммиту
`$CI_COMMIT_TAG`	Тег Git этого пайплайна. Строка с этой переменной пропускается в пайплайне ветки
`$CI_APPLICATION_REPOSITORY`	Стандартный путь к образу: `$CI_REGISTRY_IMAGE/$CI_COMMIT_REF_SLUG` для ветки, `$CI_REGISTRY_IMAGE` для тега. Проект, который переопределил переменную для собственной сборки Auto DevOps, здесь не учитывается. Его образ указывайте полностью
`$CI_APPLICATION_TAG`	Стандартный тег образа: `$CI_COMMIT_SHA` для ветки, `$CI_COMMIT_TAG` для тега

Сервер подставляет в задачу значения этих переменных из проекта, ветки и коммита пайплайна, с приоритетом над одноимённой переменной CI/CD сканируемого проекта, поэтому проект не может направить обязательное сканирование на другой образ. Строка пропускается в пайплайне, который не может её заполнить: строка с `$CI_COMMIT_TAG` в пайплайне ветки, или строка с `$CI_REGISTRY_IMAGE` либо `$CI_APPLICATION_REPOSITORY`, когда хранилище образов выключено для проекта или для экземпляра. Остальные задачи выводят в лог каждую пропущенную строку: `Trivy container: line <N> skipped (<reason>): <line>`. Когда ни одна строка не подходит пайплайну или поле пусто, выполняется одна задача `trivy_container` (плюс суффикс политики): она завершается с ошибкой, а при установленном флажке «Не проваливать задачу, если по адресу нет образа» — успешно, с сообщением в логе.

Пример поля из нескольких строк приведён в разделе «Сканирование образов контейнеров».

Учётные данные хранилища образов

Чтение образа требует учётных данных, которые зависят от того, где он хранится. Не помещайте учётные данные хранилища образов в политику: она хранит свои значения в открытом виде.

Если адрес образа совпадает с `CI_REGISTRY`, образ лежит в хранилище образов этой установки. Задача читает такой образ только собственным токеном CI/CD. Задача никогда не отправляет этот токен в другое хранилище. Образ из другого приватного проекта этой установки читается, только если тот проект разрешает доступ токену задачи сканируемого проекта: в проекте с образом откройте «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD» и добавьте туда сканируемый проект. Иначе задача завершается с ошибкой `UNAUTHORIZED` и `DENIED: access forbidden`. Для хранилища образов этой установки задача не читает учётные данные из переменных сканируемого проекта.

Образ в другом хранилище образов читается по `DOCKER_AUTH_CONFIG` либо по паре `CS_REGISTRY_USER` и `CS_REGISTRY_PASSWORD` из переменных CI/CD сканируемого проекта. `DOCKER_AUTH_CONFIG` — это JSON в формате `~/.docker/config.json`:

```
{"auths": {"registry.example.com": {"auth": "<BASE64_USER_PASSWORD>"}}}
```

`<BASE64_USER_PASSWORD>` — это `user:password`, закодированное в Base64. Пароль передаётся в Trivy через стандартный ввод и не попадает в лог задачи, даже если переменная не замаскирована. Задача печатает команду Trivy, которую выполняет, без пароля.

Образ, сохранённый как артефакт

`artifact:dist/image.tar` читает `$CI_PROJECT_DIR/dist/image.tar`, и задача сборки того же пайплайна должна его там оставить:

- задача сборки выполняется на этапе раньше этапа сканирований (по умолчанию `fe-security-scanner`), поскольку у задачи сканирования нет `needs`, и она загружает артефакты всех более ранних этапов;
- файл перечислен в `artifacts:paths` и не истёк к моменту запуска сканирования;
- образ сохранён вместе с тегом, например `docker save myapp:1.2 -o dist/image.tar`: имя образа в несоответствии берётся из тега внутри архива, поэтому `docker save <ID образа>` без тега даёт несоответствия без имени образа;
- для запуска сканирования в пайплайнах запросов на слияние задаче сборки нужны `rules`: задача без `rules` или `only` не добавляется в пайплайн запроса на слияние, тогда как задачи политики добавляются в него всегда, и тогда задача строки `artifact:` завершается с ошибкой.

Например, в `.gitlab-ci.yml` сканируемого проекта:

```
build-image:
  stage: build
  rules:
    - when: on_success
  script:
    - docker build -t myapp:$CI_COMMIT_SHORT_SHA .
    - mkdir -p dist
    - docker save myapp:$CI_COMMIT_SHORT_SHA -o dist/image.tar
  artifacts:
    paths: [dist/image.tar]
    expire_in: 1 day
```

Артефакту не нужны ни хранилище образов, ни учётные данные.

Пайплайны без сборки образа

Задачи политики добавляются в пайплайны любого типа, включая пайплайны запросов на слияние, и обходят `workflow:rules` проекта:

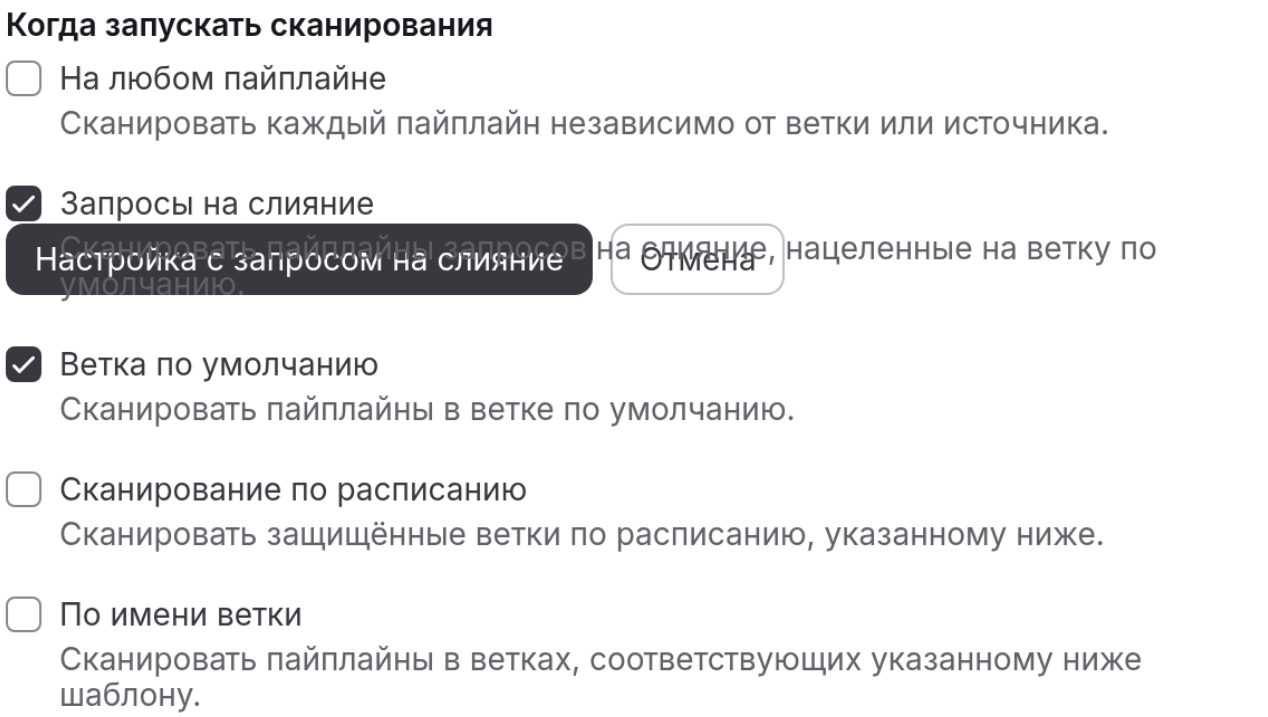
- проект без задач для пайплайнов запросов на слияние всё равно получает такой пайплайн, и в нём есть только сканирования;
- настройка «пайплайн запроса на слияние вместо пайплайна ветки» через `workflow:rules` не убирает дублирование: пайплайн ветки остаётся, и в нём тоже есть только сканирования;
- проект без собственного `.gitlab-ci.yml` всё равно получает пайплайны, в которых есть только задачи сканирования.

В пайплайне, где сборка не выполнялась, задачи строк `artifact:` и адресов, куда сборка должна была отправить образ, завершаются с ошибкой. Например, в пайплайне, который запускает правило политики с `type: schedule`, выполняются только задачи сканирования, и задачи строк `artifact:` в нём завершаются с ошибкой, а адрес хранилища образов работает, если образ там уже есть.

Политика, которой не нужно сканирование при каждой отправке изменений, может ограничить правило источниками пайплайна.

Режим правил

Снимите флажок «На любом пайплайне» и отметьте «Запросы на слияние» и «Ветка по умолчанию»: сканирование тогда выполняется в пайплайнах запросов на слияние, нацеленных на ветку по умолчанию, и в пайплайнах, которые запускает отправка изменений в ветку по умолчанию.



Режим .yaml

Эти флажки записывают правила ниже. `pipeline_sources.including` также принимает `api`, `chat`, `external`, `external_pull_request_event`, `merge_request_event`, `pipeline`, `push`, `schedule`, `trigger`, `web`. Вручную написанное правило может добавить источники, которых нет среди флажков, например `web`.

```
rules:
  - type: pipeline
    branch_type: target_default
    pipeline_sources:
      including:
        - merge_request_event
  - type: pipeline
    branch_type: default
    pipeline_sources:
      including:
        - push
```

Либо установите флажок «Не проваливать задачу, если по адресу нет образа» для веток, которые ничего не собирают.

Сканирование кластеров Kubernetes

Модуль сканирует образы нагрузок, которые работают в перечисленных неймспейсах кластера Kubernetes. Задача `trivy_k8s` запускает `trivy k8s` на раннере и обращается к кластеру через агент для Kubernetes по его `ci_access`. Kubeconfig тот же, который агент выдаёт обычной задаче CI/CD. Модуль ничего не устанавливает в кластер, не настраивает агент и не клонирует сканируемый репозиторий. Модуль сообщает только об уязвимостях в пакетах образов. Модуль не проверяет конфигурацию, RBAC, секреты в манифестах, узлы и control plane.

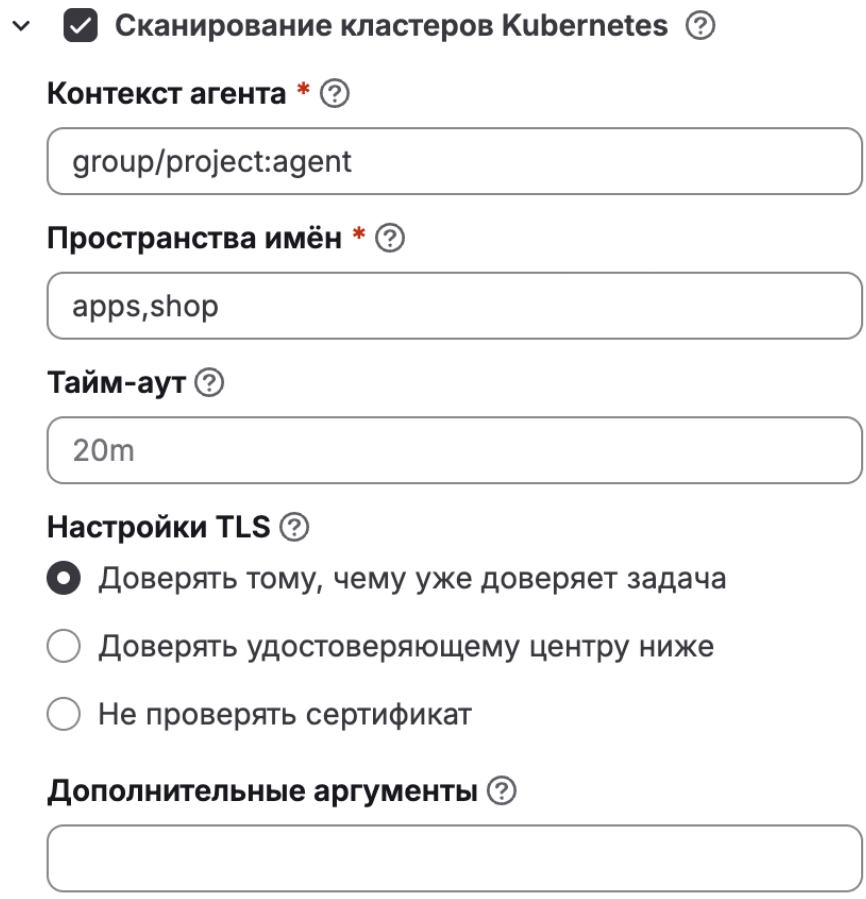
Собственные поля модуля:

Поле	Переменная	Что задаёт
«Контекст агента»	`FE_SCANS_TRIVY_K8S_AGENT`	Обязательное. Агент в виде пути его проекта, двоеточия и имени, например `group/project:agent`
«Пространства имён»	`FE_SCANS_TRIVY_K8S_NAMESPACES`	Обязательное. Имена неймспейсов через запятую без пробелов, например `apps,shop`
«Тайм-аут»	`FE_SCANS_TRIVY_K8S_TIMEOUT`	Сколько времени Trivy может потратить на всё сканирование кластера: длительность в формате Go, например `20m`, `1h30m` или `90s`. Пусто — `20m`
«Настройки TLS»	`FE_SCANS_TRIVY_K8S_TLS_MODE`	Как задача проверяет сертификат прокси агента (раздел «Настройки TLS прокси агента»). По умолчанию — `system`
«Удостоверяющий центр»	`FE_SCANS_TRIVY_K8S_CA_CERT`	Сертификат в формате PEM. Появляется и становится обязательным при варианте «Доверять удостоверяющему центру ниже»
«Дополнительные аргументы»	`FE_SCANS_TRIVY_K8S_EXTRA_ARGS`	Флаги `trivy k8s`, которых нет в форме. Задача передаёт их без изменений, например `--parallel 2`

Редактор не принимает пустой список неймспейсов: без списка Trivy сканирует весь кластер. Редактор отклоняет тайм-аут без единицы (`20`), с заглавной единицей (`10M`) и нулевой (`0s`). Пока поле не исправлено, создать запрос на слияние нельзя. Задача ставит «Дополнительные аргументы» перед своими флагами. Если флаг задан в обоих местах, действует флаг задачи: `--scanners`, `--include-namespaces`, `--timeout` и другие. Если снять флажок модуля, редактор удаляет из политики все переменные модуля.

Режим правил

1. В блоке «Trivy (встроенный)» установите флажок «Сканирование кластеров Kubernetes».
2. В поле «Контекст агента» введите агент, например `group/project:agent`.
3. В поле «Пространства имён» введите неймспейсы для сканирования, например `apps,shop`.
4. В разделе «Когда запускать сканирования» установите флажок «Сканирование по расписанию» и заполните поле «Расписание» (раздел «Когда запускать сканирование кластеров»).



Режим .yaml

```
scan_execution_policy:
  - name: Cluster scan
    enabled: true
    rules:
      - type: schedule
        cadence: '0 3 * * *'
        timezone: Europe/Moscow
        branch_type: default
    actions:
      - scan: trivy
        variables:
          FE_SCANS_TRIVY_SCAN_IMAGE: 'registry.example.com/trivy-with-db:0.74.0'
          FE_SCANS_TRIVY_K8S: 'true'
          FE_SCANS_TRIVY_K8S_AGENT: 'group/project:agent'
          FE_SCANS_TRIVY_K8S_NAMESPACES: 'apps,shop'
```

Собственного порога и образа у модуля нет: действуют «Порог блокировки» и «Образ сканирования» из блока «Общие настройки». Модуль применяет общий trivy.yml. Если файл назван без проекта, задача загружает его с ветки по умолчанию сканируемого проекта (раздел «Файлы конфигурации»). Модуль не ищет секреты, поэтому trivy-secret.yaml к нему не относится.

Одна политика сканирует один кластер: для второго кластера нужна вторая политика со своим агентом. Политику с двумя действиями `trivy`, в которых включён модуль кластеров, сохранить нельзя, даже если она написана вручную. Каждая политика с модулем кластеров получает собственную задачу `trivy_k8s`, а каждая политика по расписанию — собственный пайплайн.

Настройки TLS прокси агента

Задача обращается к кластеру через прокси агента этой установки по TLS. «Настройки TLS» определяют, как задача проверяет сертификат прокси:

Вариант	Значение в политике	Чему доверяет задача
«Доверять тому, чему уже доверяет задача»	`system`	Системному хранилищу сертификатов образа сканирования. Подходит, если сертификат прокси выдан публичным удостоверяющим центром
«Доверять удостоверяющему центру ниже»	`custom_ca`	Только сертификату из поля «Удостоверяющий центр»: удостоверяющему центру, который подписал сертификат прокси, или самому сертификату прокси
«Не проверять сертификат»	`skip_verify`	Любому сертификату. Любой, кто перехватит соединение между раннером и прокси, получит токен задачи. Используйте этот вариант только в тестовой установке

При `custom_ca` политику нельзя сохранить без сертификата, который начинается с `-----BEGIN CERTIFICATE-----`:

```
FE_SCANS_TRIVY_K8S_TLS_MODE: custom_ca
FE_SCANS_TRIVY_K8S_CA_CERT: |
  -----BEGIN CERTIFICATE-----
  MIIDdzCCAl+gAwIBAgIU...
  -----END CERTIFICATE-----
```

Настройка касается только соединения задачи с прокси агента. Соединение агента с установкой не меняется. Обе переменные объявлены в задаче, поэтому переменные CI/CD сканируемого проекта их не переопределяют.

Доступ к кластеру

Задача выполняется в каждом проекте, на который распространяется политика. `KUBECONFIG` с контекстом агента задача получает, только когда выполнены все условия:

- проект или его группа перечислены в `ci_access` конфигурации агента, `.gitlab/agents/<имя агента>/config.yaml` в проекте агента;
- проект агента находится в той же группе верхнего уровня, что и сканируемый проект, если не включена авторизация агентов на уровне всей установки;

- в записи `ci_access` нет `environments`: задача сканирования выполняется вне окружения;
- в записи не задано `protected_branches_only: true`, либо пайплайн выполняется на защищённой ветке.

```
ci_access:
  projects:
    - id: group/scanned-project
  groups:
    - id: group
```

Проекту самого агента запись в `ci_access` не нужна. В кластере у задачи права агента (`access_as: agent`, значение по умолчанию записи `ci_access`). Чарт `gitlab-agent` по умолчанию выдаёт агенту `cluster-admin`. Самому сканированию нужны `get` и `list` на Pod, Deployment, ReplicaSet, StatefulSet, DaemonSet, Job, CronJob, ReplicationController, Secret и ServiceAccount. Параметр чарта `rbac.useExistingRole` сужает права агента.

Без доступа задача завершается с ошибкой до запуска Trivy и перечисляет все четыре условия (сообщение `has no KUBECONFIG` в разделе «Диагностика»).

Когда запускать сканирование кластеров

Модуль работает с любым правилом политики. Задача `trivy_k8s` выполняет одну и ту же работу в пайплайне по расписанию и в пайплайне пуша. Состояние кластера не зависит от коммита. Правило `schedule` сканирует кластер раз в период. Правило `pipeline` сканирует те же образы заново при каждом пуше в подходящую ветку.

Учётные данные хранилищ образов кластеров

Trivy берёт пароль к хранилищу образов из кластера: из Secret, который назван в `imagePullSecrets` пода или его ServiceAccount. Для таких образов ничего настраивать не нужно.

Узлы могут загружать образы по учётным данным kubelet или через плагин kubelet для получения учётных данных. Если при этом у подов нет `imagePullSecrets`, Trivy не видит учётных данных и не сканирует образ. Чтобы узнать, у каких подов есть `imagePullSecrets`, выполните команду:

```
d8 k -n <NAMESPACE> get pods -o jsonpath='{range .items[*]}{.metadata.name}{"\t"}
{.spec.imagePullSecrets}{"\n"}{end}'
```

Для образов, у подов которых нет `imagePullSecrets`, задайте `DOCKER_AUTH_CONFIG` в переменных CI/CD сканируемого проекта, в формате из раздела «Учётные данные хранилища образов» выше. Задача записывает `DOCKER_AUTH_CONFIG` в `~/.docker/config.json` перед сканированием и сообщает об этом в логе: `Trivy k8s: using DOCKER_AUTH_CONFIG for images the cluster holds no pull secret for.` Trivy сначала пробует учётные данные из кластера и читает этот файл, когда их нет.

Образы загружает раннер. Хранилище образов, доступное только узлам кластера, для сканирования недоступно.

Если раннер не загрузил образ, на вкладке «Тесты» появляется ошибка `Could not scan <образ>` с сообщением Trivy. Несоответствий по этому образу нет. Если не удалось просканировать только часть ресурсов, задача остаётся успешной и пишет в лог, сколько ресурсов из скольких не просканировано: `trivy_k8s: 1 of 4 resources could not be scanned; each is an error in the test summary.` Если не удалось просканировать ни одного ресурса, сканирование не завершено. Задача завершается с ошибкой `ERROR: the scan itself did not complete, so this pipeline scanned no cluster.` Отчёт об уязвимостях получает отчёт незавершённого сканирования без несоответствий. В DefectDojo ничего не выгружается. Неймспейс без нагрузок ошибкой не считается.

Несоответствия сканирования кластеров

Несоответствия попадают в отчёт об уязвимостях с источником «Сканирование образов кластеров». Расположение несоответствия — нагрузка в виде `namespace/kind/name`, затем образ без тега и пакет, например `apps/Deployment/web nginx:libssl1.1`. Один образ в двух нагрузках даёт два несоответствия, поскольку исправлять нужно обе нагрузки. Одна и та же нагрузка в двух кластерах тоже даёт два несоответствия: они различаются агентом, которого в расположении не видно. Повторное сканирование и новый тег того же образа снова дают те же несоответствия.

Trivy не сообщает имена контейнеров, поэтому несоответствие называет контейнер по образу, без хранилища и тега: `redis`, `nginx`, `alpine`. Имя совпадает с `spec.containers[].name` только у контейнеров, названных по своему образу.

На вкладке «Тесты» у теста сканирования кластеров нет файла: нагрузка указана в квадратных скобках в имени теста и строкой `Resource: namespace/kind/name` в его тексте. Лог задачи не выводит индикатор прогресса Trivy. Вместо него при каждом изменении счётчика задача пишет в лог `Trivy k8s: N of M resources scanned`. Исходный отчёт Trivy лежит в артефактах задачи: `security-reports/raw/trivy-k8s.json`.

Интеграция с DefectDojo выгружает отчёт парсером «GitLab Container Scan» в тест с именем задачи, `trivy_k8s:<часть имени политики>-p<ID>`. Отдельного поля для нагрузки у этого парсера нет, поэтому заголовок несоответствия называет пакет, образ, нагрузку и уязвимость, например `apt 1.4.8 in redis:5.0.0 (apps/Deployment/cache): CVE-2019-3462 …`. Две политики с модулем кластеров в одном проекте дают два теста. После переименования политики DefectDojo получает новый тест.

Ограничения сканирования кластеров

- **Один кластер в нескольких проектах сканируется несколько раз.** Задача выполняется в каждом проекте, на который распространяется политика, и у каждого проекта своя копия несоответствий. Сканируйте кластер из одного проекта: привяжите политику с модулем только к нему или сузьте её через `policy_scope`.

- **Сканируемый проект может перенаправить сканирование.** Переменные Kubernetes от агента попадают в задачу раньше переменных CI/CD проекта. Поэтому переменная `KUBECONFIG`, заданная в сканируемом проекте, заменяет kubeconfig агента. Задача сканирует кластер, который выбрал проект, или завершается с ошибкой о контексте.
- **Лимит Docker Hub.** Docker Hub ограничивает анонимные загрузки с `index.docker.io` на один IP-адрес. Действующие лимиты приведены на странице лимитов Docker Hub. После исчерпания лимита Trivy сообщает `TOOMANYREQUESTS`, и образ становится ошибкой на вкладке «Тесты».
- **Отчёт растёт вместе с образами.** Неймспейс с тремя небольшими образами даёт отчёт около 1 МБ. Один образ `python:3.9` даёт отчёт 33 МБ, а исходный отчёт Trivy — 51 МБ. Отчёт больше лимита размера артефактов установки (`max_artifacts_size`, по умолчанию 100 МБ) не выгружается. Сканируйте крупные неймспейсы в разных политиках.

Файлы конфигурации

Часть настроек Trivy читает из файлов, которые названы в блоке «Общие настройки»: trivy.yml, собственного файла конфигурации Trivy, и trivy-secret.yaml, правил и списков исключений поиска секретов. В примере раздела «Общие настройки» оба файла читаются из проекта `security/trivy-config`.

Задачи всех модулей читают один и тот же trivy.yml: с указанным проектом — из ветки по умолчанию этого проекта, без проекта — из сканируемого репозитория, без файла — конфигурация не передаётся. У модуля кластеров нет рабочей копии. Файл trivy.yml, названный без проекта, задача загружает с ветки по умолчанию сканируемого проекта как снимок всего дерева. На большом репозитории задача выполняется дольше. Для сканирования кластеров укажите проект, в котором лежит trivy.yml. Один trivy.yml одинаково подходит модулям, которые работают с репозиторием, модулю образов контейнеров и модулю кластеров: Trivy без предупреждения пропускает ключи, которые не использует конкретный запуск. Пару trivy-secret.yaml читают модуль обнаружения секретов и модуль образов контейнеров. Если пара не задана, оба используют встроенные правила Trivy. Модуль образов контейнеров никогда не подхватывает trivy-secret.yaml из репозитория самостоятельно. Правила поиска секретов, записанные прямо в trivy.yml, Trivy молча игнорирует: такие правила должны быть в trivy-secret.yaml. Не задавайте порог в trivy.yml: его ключ `severity` вырезает несоответствия ниже указанного уровня из отчёта, и они не попадают даже в отчёт об уязвимостях. Что приводит задачу к ошибке, решает порог блокировки политики.

Относительные пути внутри trivy.yml и файлы, которые Trivy ищет по умолчанию (`trivy.yaml`, `trivy-secret.yaml`, `.trivyignore`), разрешаются относительно рабочего каталога задачи — рабочей копии сканируемого проекта, а не цели сканирования. Для модуля образов контейнеров это значит, что пути в trivy.yml указывают на репозиторий, а не на образ. При пустом поле файла задача не передаёт файл конфигурации, и Trivy в этом случае тоже не подставляет собственные файлы по умолчанию.

Обе пары, trivy.yml и trivy-secret.yaml, устроены одинаково. Пустое поле проекта означает сканируемый репозиторий: задача читает файл из рабочей копии на сканируемом коммите, и тогда конфигурацию сканирования контролирует сам сканируемый проект. Политика, которая не должна позволять сканируемому проекту менять конфигурацию, называет проект явно. Названный проект задача читает собственным токеном задачи CI/CD, всегда из его ветки по умолчанию. Токен несёт права того, кто запустил пайплайн: если у этого пользователя нет доступа к проекту, задача завершается с ошибкой. Откройте доступ одним из способов: сделайте проект с файлами внутренним или публичным; добавьте сканируемый проект (или его группу) в список разрешённых токена задач CI/CD проекта с файлами; сделайте того, кто запускает пайплайны, участником проекта с файлами. При сохранении политики названный проект проверяется только на существование и наличие репозитория. Права доступа не проверяются, потому что читателем файла станет тот, кто запустит пайплайн позже.

Поле проекта редактор показывает, только пока заполнено поле файла; очистка поля файла убирает проект из политики. Политику, написанную вручную с проектом и без файла, сохранить нельзя: ошибка называет поле, которое нужно заполнить. Уже действующая политика с проектом и без файла останавливает задачу перед сканированием, назвав пустое поле. Файл без проекта читается из сканируемого репозитория.

Задача останавливается, если названный проект не удалось прочитать, файла нет в источнике, файл пуст или путь абсолютный либо содержит `..` (схема политики тоже отклоняет такие пути). Лог задачи всегда называет источник файла, а для названного проекта — ещё и коммит, на котором файл прочитан: `Trivy: configuration read from <project> at <commit SHA>.` и `Trivy: <VARIABLE> applied from <source> (<file>).` Этот коммит — единственное место, где видно, какие правила применило сканирование. Формат trivy.yml описан в справочнике Trivy по файлу конфигурации, а формат trivy-secret.yaml — в справочнике Trivy по сканеру секретов.

Блокировка

Задачи `trivy_sca`, `trivy_license`, `trivy_container` и `trivy_k8s` завершаются с ошибкой, когда несоответствие достигает порога блокировки. Все четыре читают общий «Порог блокировки», `FE_SCANS_TRIVY_FAIL_ON`, и завершаются успешно, если порога не достигло ни одно несоответствие. Задача `trivy_secrets` общий порог не читает: её завершает с ошибкой любой секрет, какой бы ни была его критичность. Флажок «Не блокировать на находках» присваивает ей порог `report_only`. Несоответствия ниже порога всё равно попадают в отчёты и на вкладку «Тесты»: порог решает только исход задачи.

Название	Значение в политике	Что приводит задачу к ошибке
«Только критический»	`critical`	Критическое несоответствие
«Высокий и выше» (по умолчанию у модулей сканирования зависимостей, лицензий, образов контейнеров и кластеров)	`high`	Высокое и критическое
«Средний и выше»	`medium`	Среднее и выше
«Низкий и выше»	`low`	Низкое и выше
«Любая находка»	`info`	Несоответствие любой распознанной критичности
«Ничего не блокировать»	`report_only`	Ничего; несоответствия всё равно отражаются в отчётах

Несоответствие с неизвестной критичностью (`UNKNOWN`) не приводит задачу к ошибке ни при каком пороге, включая «Любая находка». Уязвимость, которую дистрибутив пометил как `not_affected` , `will_not_fix` , `end_of_life` или `fix_deferred` , отражается на вкладке «Тесты» как пропущенная и никогда не приводит задачу к ошибке. На вкладке «Тесты» несоответствие на уровне порога или выше — это неудачный тест, ниже порога — пройденный, отложенное — пропущенный. Сканирование без несоответствий даёт один тест, «No findings».

В версии Trivy, которая поставляется в образе, у секретов разная критичность: ключ доступа AWS и токен GitHub — критической, приватный ключ — высокой, JWT и вебхук Slack — средней, публикуемый ключ Stripe — низкой. Секрет, попавший в образ, оценивается по общему порогу, как и уязвимости образа: при пороге по умолчанию «Высокий и выше» секрет средней критичности, например JWT, попадает в отчёт, но не приводит задачу сканирования образа к ошибке (раздел «Сканирование образов контейнеров»).

`FE_SECURITY_FAIL_ON` , записанная прямо в `variables` действия `trivy` , игнорируется. Задача также завершается с ошибкой, если сам Trivy не отработал до конца, независимо от порога: тогда лог показывает `ERROR: the scan itself did not complete, so this pipeline scanned nothing.` Задача, завершившаяся с ошибкой, печатает, сколько несоответствий достигло порога, например `ERROR: 3 finding(s) at or above high.` (у задачи лицензий — `licence(s)` , у задачи секретов — `secret(s)`).

Где отображаются результаты

Несоответствия и отчёты появляются в разных местах в зависимости от того, что это за данные.

Где	Что там видно
Вкладка «Тесты» пайплайна	По одному набору JUnit на каждую выполнившуюся задачу: несоответствия на уровне порога и выше — как неудачные тесты, ниже — как пройденные, отложенные — как пропущенные
«Итоги тестирования» запроса на слияние	Те же наборы, сопоставленные с целевой веткой
«Отчёт об уязвимостях» проекта	Несоответствия `trivy_sca` (сканирование зависимостей), `trivy_secrets` (обнаружение секретов), каждой задачи `trivy_container` (сканирование образов контейнеров) и `trivy_k8s` (сканирование образов кластеров). Несоответствия по лицензиям сюда не попадают
Страницы «Список зависимостей» и «Соответствие лицензий»	SBOM от `trivy_sca` и `trivy_license`, из последнего пайплайна ветки по умолчанию. Подробнее — на странице Список зависимостей и соответствие лицензий
DefectDojo	Отчёты сканирования зависимостей, секретов, образов контейнеров и образов кластеров, если настроена интеграция с DefectDojo

Когда страница «Отчёт об уязвимостях» доступна в проекте и как с ней работать, описано на странице Отчёт об уязвимостях. В отчёте об уязвимостях инструментом этих несоответствий указан Trivy, какой бы модуль их ни нашёл.

Образ сканирования и базы уязвимостей

Trivy выполняется в собственном образе, помеченном версией Trivy, которую несёт: по умолчанию это `trivy-with-db:0.74.0` из хранилища образов продукта (раздел «Общие настройки»). Образ несёт Trivy, обе базы Trivy — базу уязвимостей и базу Java — по пути `/var/cache/trivy` (`db/trivy.db`, `java-db/trivy-java.db`), Ruby с библиотеками `json` и `csv` для сборки отчётов и `git` для чтения файлов конфигурации из проекта. Опубликованный образ по умолчанию собран только под `linux/amd64`: раннеру другой архитектуры, например `arm64`, нужен собственный образ сканирования, собранный под его архитектуру (раздел «Собственный образ сканирования»).

Задачи запускают Trivy с выключенными обновлением баз и проверкой версии, поэтому сканирование ничего не загружает и не нуждается в выходе в интернет; базы остаются такими же свежими, как на момент сборки образа. `trivy --version` в начале лога каждой задачи печатает дату сборки базы (`UpdatedAt`) и `NextUpdate` — по этой строке видно, насколько устарела встроенная база. Хранилище, где лежит образ по умолчанию, отклоняет анонимную загрузку. Раннер загружает образ сканирования так же, как любой образ задачи: ему нужен доступ к этому хранилищу либо копия образа в доступном ему хранилище, указанная в поле «Образ сканирования».

Политика загрузки образа на раннере

Базы обновляются пересборкой образа под тем же тегом, поэтому каждая задача Trivy просит у раннера `pull_policy: [always, if-not-present]` : загружать образ при каждом запуске и использовать локальную копию, если хранилище недоступно. Раннер применяет политики, которые просит задача и которые сам разрешает: пересечение списка задачи с `allowed_pull_policies` раннера, а если `allowed_pull_policies` не задан — с его собственным `pull_policy` . У executor Docker `pull_policy` по умолчанию равен `always` ; чтобы базы обновлялись, раннер должен разрешать `always` . Раннер, у которого `pull_policy` — это `if-not-present` , а `allowed_pull_policies` не задан, загружает образ один раз и дальше сканирует с теми же базами.

Раннер разрешает	Что происходит
`always` и `if-not-present`	Образ загружается при каждом запуске; если хранилище образов недоступно, используется локальная копия
только `always`	Образ загружается при каждом запуске; если хранилище образов недоступно, задача завершается с ошибкой
только `if-not-present`	Образ загружается один раз; свежие базы не приходят
Ни одна, например только `never`	Задача завершается с ошибкой ещё до запуска, с сообщением об ошибке политики загрузки раннера

Чтобы разрешить обе политики для задач Trivy, не меняя политику раннера по умолчанию для остальных задач, настройте executor Docker в `config.toml` раннера:

```
[[runners]]
  executor = "docker"
  [runners.docker]
    pull_policy = ["if-not-present"]
    allowed_pull_policies = ["always", "if-not-present"]
```

Задачи, которые не называют политику загрузки, по-прежнему используют `if-not-present`; задачи Trivy получают обе. `pull_policy` раннера должен содержать хотя бы одну из разрешённых политик. У executor Kubernetes те же ключи — в `[runners.kubernetes]`. Раннер, который ограничивает образы задач через `allowed_images`, должен разрешить образ сканирования.

Закрытый контур

Самому сканированию интернет не нужен: базы уже в образе. Установке без выхода в интернет нужен образ сканирования в доступном ей хранилище: скопируйте образ в собственное хранилище и укажите копию в поле «Образ сканирования» политики. Политика сохраняет названный образ и после обновления продукта. Чтобы перевести политику на новый образ, обновите «Образ сканирования».

Собственный образ сканирования

Собирайте собственный образ сканирования для раннера другой архитектуры процессора или чтобы самим решать, когда обновлять базы. Встройте базы по тем же путям, `/var/cache/trivy/db/trivy.db` и `/var/cache/trivy/java-db/trivy-java.db`, доступным для записи пользователю задачи. В образе также нужен `git`, чтобы читать файлы конфигурации из проекта, и Ruby с библиотеками `json` и `csv`, чтобы собирать отчёты. Задача, в образе которой нет Ruby, устанавливает его через `apk` или `apt-get`, если один из них есть в образе, а иначе завершается с ошибкой. Способ встроить базы: `trivy --cache-dir /var/cache/trivy image --download-db-only` и `trivy --cache-dir /var/cache/trivy image --download-java-db-only`. Чтобы обновить базы, пересоберите образ под тем же тегом; раннеры на политике загрузки `always` подхватят пересборку автоматически (раздел «Политика загрузки образа на раннере»). Раннеру архитектуры, отличной от `linux/amd64`, например `arm64`, тоже нужен образ, собранный под его архитектуру, например с помощью `docker buildx build --platform linux/arm64`.

Диагностика

Причину сбоя задачи Trivy или её отсутствия показывает пайплайн или лог задачи.

Сканирование не появляется в пайплайне

Проверьте, что:

- проект политик привязан к проекту или к группе выше него, и в его `policy.yml` есть действие `scan: trivy` (в редакторе политик оно называется «Trivy (встроенный)»);
- действие включает хотя бы один модуль значением `'true'`;

- правила политики подходят пайплайну: его ветке и его источнику;
- запрос на слияние с политикой принят в проекте политик;
- доступен GitLab Runner с executor `docker`.

Задача завершается с ошибкой «turns no module on»

Это задача `trivy_no_module`: действие `trivy` называет переменные модулей, но ни одна из них не равна `'true'`.

```
ERROR: this policy mandates a Trivy scan but turns no module on.
  Nothing was scanned. Open the policy and enable the dependency scan, the licence
scan, the secret search, the container image scan, the cluster scan, or any of them --
  in the editor, or by naming FE_SCANS_TRIVY_SCA / FE_SCANS_TRIVY_LICENSE /
FE_SCANS_TRIVY_SECRETS / FE_SCANS_TRIVY_CONTAINER / FE_SCANS_TRIVY_K8S in the policy.
```

Установите флажок у блока модуля в редакторе или задайте одной из этих переменных значение `'true'` в политике.

Задача trivy_no_image завершается с ошибкой

Политика включает модули Trivy и не называет «Образ сканирования»: она написана до того, как образ стал обязательным. Вместо задач Trivy в пайплайне одна задача `trivy_no_image`, и её лог называет `FE_SCANS_TRIVY_SCAN_IMAGE` и поле «Образ сканирования».

Укажите образ сканирования в политике (раздел «Общие настройки»).

Задача останавливается: проект указан без файла

Задача останавливается перед сканированием, потому что общее поле «Проект с trivy.yml» или «Проект с trivy-secret.yaml» заполнено, а его собственное поле файла пусто. Лог называет пустое поле.

Заполните поле файла либо очистите поле проекта.

Задача завершается с ошибкой «carries no vulnerability database»

Это задача `trivy_sca`, `trivy_container` или `trivy_k8s`, и «Образ сканирования» называет образ без встроенных баз, например публичный образ `aquasec/trivy`.

```
ERROR: this scan image carries no vulnerability database at /var/cache/trivy/db/
trivy.db.
```

Если отсутствует база Java, сообщение вместо этого называет `java-db/trivy-java.db`. Задача останавливается и базу не загружает. Назовите образ с базами по пути `/var/cache/trivy` (раздел «Собственный образ сканирования»).

Раннер отклоняет политику загрузки образа

Задача завершается с ошибкой ещё до запуска своего скрипта, с ошибкой вида `the configured PullPolicies (…) are not allowed by AllowedPullPolicies (…)`. Раннер не разрешает ни `always`, ни `if-not-present`. Разрешите обе политики на раннере (раздел «Политика загрузки образа на раннере»).

Сканирование не находит недавно опубликованную уязвимость

Базы в образе, который использовал раннер для сканирования, старше, чем уведомление об уязвимости. Проверьте `UpdatedAt` в выводе `trivy --version` в начале лога задачи. Если эта дата старая, а опубликованный образ новее, раннер не загружает образ при каждом запуске: разрешите политику загрузки `always` (раздел «Политика загрузки образа на раннере»). Если устарел сам опубликованный образ, укажите более новый образ в поле «Образ сканирования» либо соберите собственный (раздел «Собственный образ сканирования»).

Задача завершается с ошибкой «no image applies here»

Задача завершается с ошибкой, потому что для этого пайплайна не подошла ни одна строка поля «Список образов для сканирования».

```
ERROR: this policy mandates a container image scan and no image applies here.
```

Все строки были пропущены, и лог называет причину для каждой, либо поле пустое. Добавьте строку, которую этот пайплайн может заполнить, либо установите флажок «Не проваливать задачу, если по адресу нет образа» для веток, которые ничего не собирают.

Задача завершается с ошибкой «it does not exist»

Задача завершается с ошибкой, потому что образ или файл артефакта, названные в политике, не нашлись при запуске сканирования.

```
ERROR: this policy mandates a scan of <image> and it does not exist.
```

Хранилище образов ответило, что образ или тег неизвестны, либо файл `artifact:` не найден на диске. Если образ задан адресом, перенесите задачу сборки на этап раньше этапа сканирований или укажите адрес, куда проект отправляет образ. Для артефакта выполните требования из раздела «Образ, сохранённый как артефакт». Для пайплайна запроса на слияние добавьте задаче сборки `rules`.

Задача завершается с ошибкой UNAUTHORIZED или DENIED

Для образа в хранилище образов этой установки задача завершается с ошибкой `UNAUTHORIZED` и `DENIED: access forbidden`, потому что проект с образом не разрешает доступ токену задачи сканируемого проекта. Откройте «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD» в проекте с образом и добавьте туда сканируемый проект. Для образа в другом хранилище задайте `DOCKER_AUTH_CONFIG` либо `CS_REGISTRY_USER` и `CS_REGISTRY_PASSWORD` в переменных CI/

CD сканируемого проекта. Когда сканирование завершается с ошибкой по причине, отличной от отсутствующего образа, лог также показывает `ERROR: the scan itself did not complete, so this pipeline scanned no image.`

Задача завершается с ошибкой «private or link-local address»

Лог показывает `realm host "10.0.0.5" is a private or link-local address`, когда хранилище образов отправляет Trivy за токеном на IP-адрес из приватного диапазона или диапазона loopback, например `realm="http://10.0.0.5/jwt/auth"`. Trivy отклоняет такой адрес, и никакой флаг не отключает эту проверку. Хранилище образов должно указывать realm токена именем хоста: укажите в `external_url` установки доменное имя.

Задача завершается с ошибкой «could not be read»

Задача завершается с ошибкой, потому что не смогла прочитать названный проект с конфигурацией.

```
ERROR: the Trivy configuration project <project> could not be read.
```

У токена того, кто запустил пайплайн, нет доступа к названному проекту, либо проект не существует. Откройте доступ одним из способов из раздела «Файлы конфигурации».

Соседние сообщения: `ERROR: <VARIABLE> names "<file>", which <source> does not have.` — для отсутствующего файла, `... and that file is empty.` — для пустого файла, и `... which is not a path inside a repository.` — для абсолютного пути или пути, содержащего `..`.

Trivy отклоняет значение тайм-аута

Значение `--timeout`, записанное в поле «Дополнительные аргументы» модуля, не распознаётся Trivy как длительность. Trivy останавливается при запуске, и задача завершается с ошибкой. Указывайте значение как длительность, например `90s`, `15m` или `1h30m`.

Задача сканирования лицензий то проходит, то нет на неизменном коде

Лицензия перечислена в trivy.yml под категорией, отличной от встроенной, поэтому Trivy заново выбирает одну из двух категорий при каждом запуске. Исправление — в разделе «Категории лицензий в trivy.yml».

Лог задачи предлагает выполнить npm install

`[npm] Run "npm install" to collect the license information of packages` — подсказка, которую Trivy печатает всякий раз, когда не находит `node_modules`. Отчёт при этом может быть полным, если сам lock-файл несёт лицензии. Подробнее — в разделе «Лицензии неустановленных зависимостей».

Задача завершается с ошибкой «has no KUBECONFIG»

Задача `trivy_k8s` за несколько секунд завершается с ошибкой до запуска Trivy, потому что агент не разрешил доступ сканируемому проекту.

```
ERROR: this job has no KUBECONFIG: the agent group/project:agent has not authorised
this project.
```

Дальше сообщение перечисляет четыре условия из раздела «Доступ к кластеру». Перечислите проект или его группу в `ci_access` агента без `environments`. Проверьте, что проект агента находится в той же группе верхнего уровня.

Задача завершается с ошибкой «has no context»

У задачи есть kubeconfig, но в нём нет контекста агента, указанного в поле «Контекст агента», например из-за опечатки в имени агента.

```
ERROR: the kubeconfig of this job has no context group/project:agent.
  The agents that authorised this project are:
  group/project:agent-2
```

Укажите в поле «Контекст агента» одного из перечисленных агентов. Другой вариант — разрешите доступ проекту в `ci_access` нужного агента. В обоих случаях на вкладке «Тесты» один тест `Scanner error` с тем же текстом, а отчёт об уязвимостях получает отчёт незавершённого сканирования без несоответствий.

Сканирование кластеров завершается с ошибкой x509

Лог показывает `x509: certificate signed by unknown authority`. В «Настройках TLS» выбран вариант «Доверять тому, чему уже доверяет задача», а сертификат прокси агента подписан удостоверяющим центром, которому образ сканирования не доверяет. Выберите «Доверять удостоверяющему центру ниже» и вставьте сертификат удостоверяющего центра, который подписал сертификат прокси (раздел «Настройки TLS прокси агента»).

Образ сканирования кластеров отмечен «Could not scan»

Раннер не смог загрузить этот образ: в кластере нет секрета для его загрузки, хранилище образов доступно только из кластера, или исчерпан лимит Docker Hub (`TOOMANYREQUESTS`). Задайте `DOCKER_AUTH_CONFIG` в переменных CI/CD сканируемого проекта или откройте раннеру доступ к хранилищу (раздел «Учётные данные хранилищ образов кластеров»).

Список зависимостей и соответствие лицензий

Сканирование безопасности загружает в пайплайн Software Bill of Materials (SBOM) в формате CycloneDX. Две страницы проекта читают этот артефакт и показывают его содержимое:

Страница	Адрес	Что показывает
«Список зависимостей»	`/-/security/dependencies`	Компоненты SBOM: имя, версия, тип, лицензия, идентификатор пакета
«Соответствие лицензий»	`/-/security/licenses`	Лицензии этих компонентов и количество компонентов под каждой

Обе страницы доступны только для чтения и открываются из раздела «Безопасность» в боковом меню проекта.

Условия доступности

Обе страницы есть в разделе «Безопасность», пока выполнены все условия; иначе их адреса отвечают 404:

- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- в настройках проекта включена функция «Безопасность и соблюдение норм»;
- у пользователя в проекте роль Developer, Maintainer или Owner.

Откуда берутся данные

Обе страницы читают последний пайплайн **ветки по умолчанию** проекта в любом его статусе: у выполняющегося пайплайна артефакта ещё нет, а у упавшего остаются отчёты, которые задачи загрузили с `artifacts:when: always`.

Из этого пайплайна берутся все артефакты задач с типом отчёта `cyclonedx`. Такой отчёт пишет сканирование `codescoring`, если в политике включён его модуль «SCA»: задача `codescoring_sca`, которая формируется только в этом случае, публикует `gl-sbom.cdx.json` как `artifacts:reports:cyclonedx`. Сканирование `trivy` пишет такие артефакты, если в политике включён модуль зависимостей или лицензий: `trivy_sca` публикует `gl-sbom.cdx.json`, `trivy_license` — `gl-sbom-license.cdx.json`. Подробнее — на странице Сканирование с помощью Trivy. Задача из собственного `.gitlab-ci.yml` проекта, которая загружает отчёт того же типа, попадает на эти страницы так же.

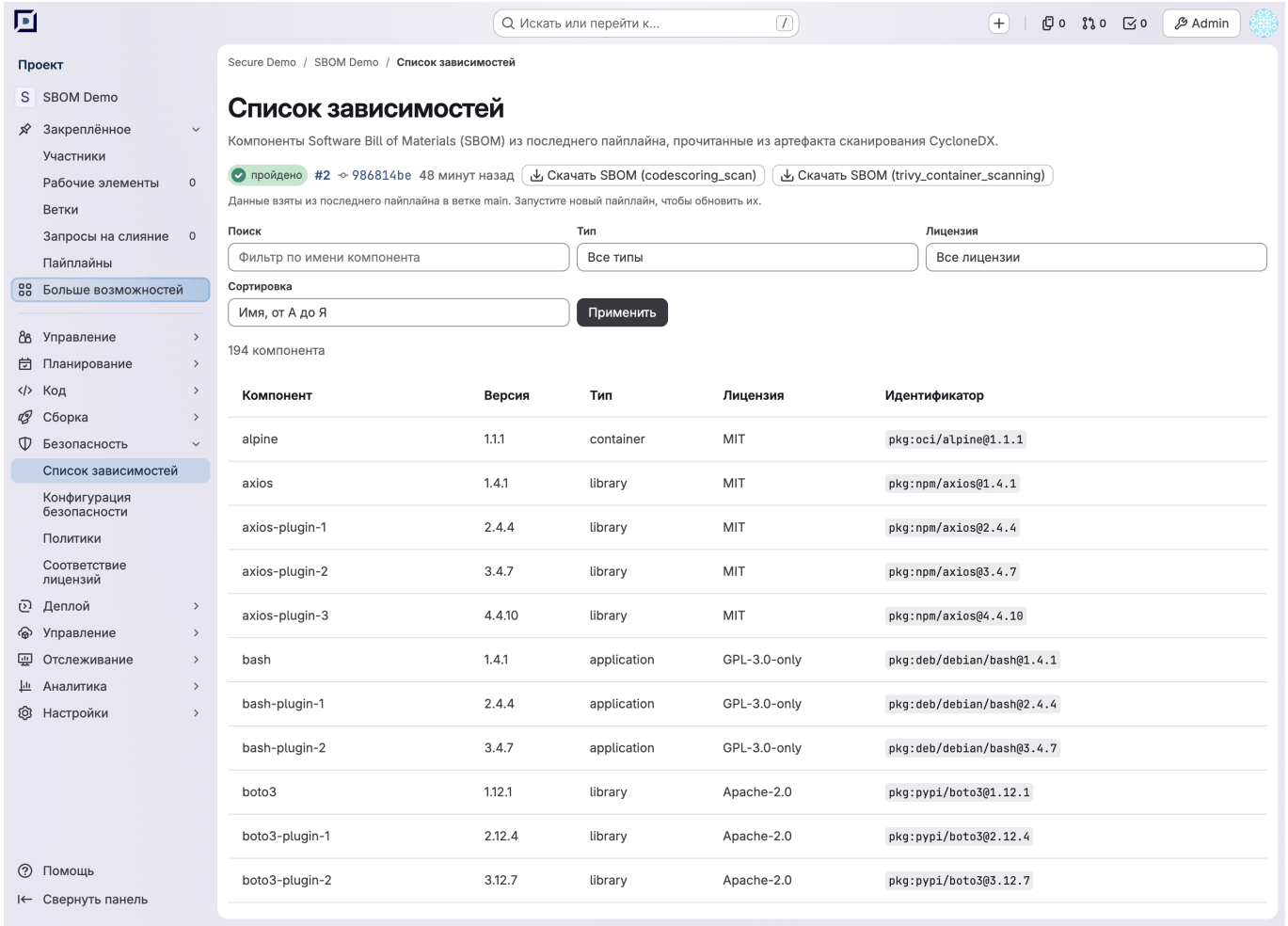
Компоненты из всех таких артефактов объединяются в одну таблицу. Компонент остаётся в ней один раз и опознаётся по идентификатору пакета, а если идентификатора нет — по имени и версии. Когда один и тот же компонент описан в нескольких артефактах, он сохраняет лицензии из всех них. Поэтому обе страницы показывают всё, что сообщили анализаторы пайплайна вместе.

В базе данных ничего не хранится: каждый запрос заново читает и разбирает артефакты. Чтобы обновить данные, запустите новый пайплайн в ветке по умолчанию.

Над таблицей обе страницы показывают, откуда взят SBOM: статус пайплайна, его номер, коммит, время запуска и кнопку скачивания для каждого артефакта CycloneDX — «Скачать SBOM» или «Скачать SBOM (`<имя задания>`)», если в пайплайне их несколько. Пользователю без доступа к артефактам задач проекта кнопка не показывается. Строка ниже называет ветку: «Данные взяты из последнего пайплайна в ветке main. Запустите новый пайплайн, чтобы обновить их.»

Для проекта, в ветке по умолчанию которого нет ни одного пайплайна, этого блока нет.

Список зависимостей



Столбцы таблицы:

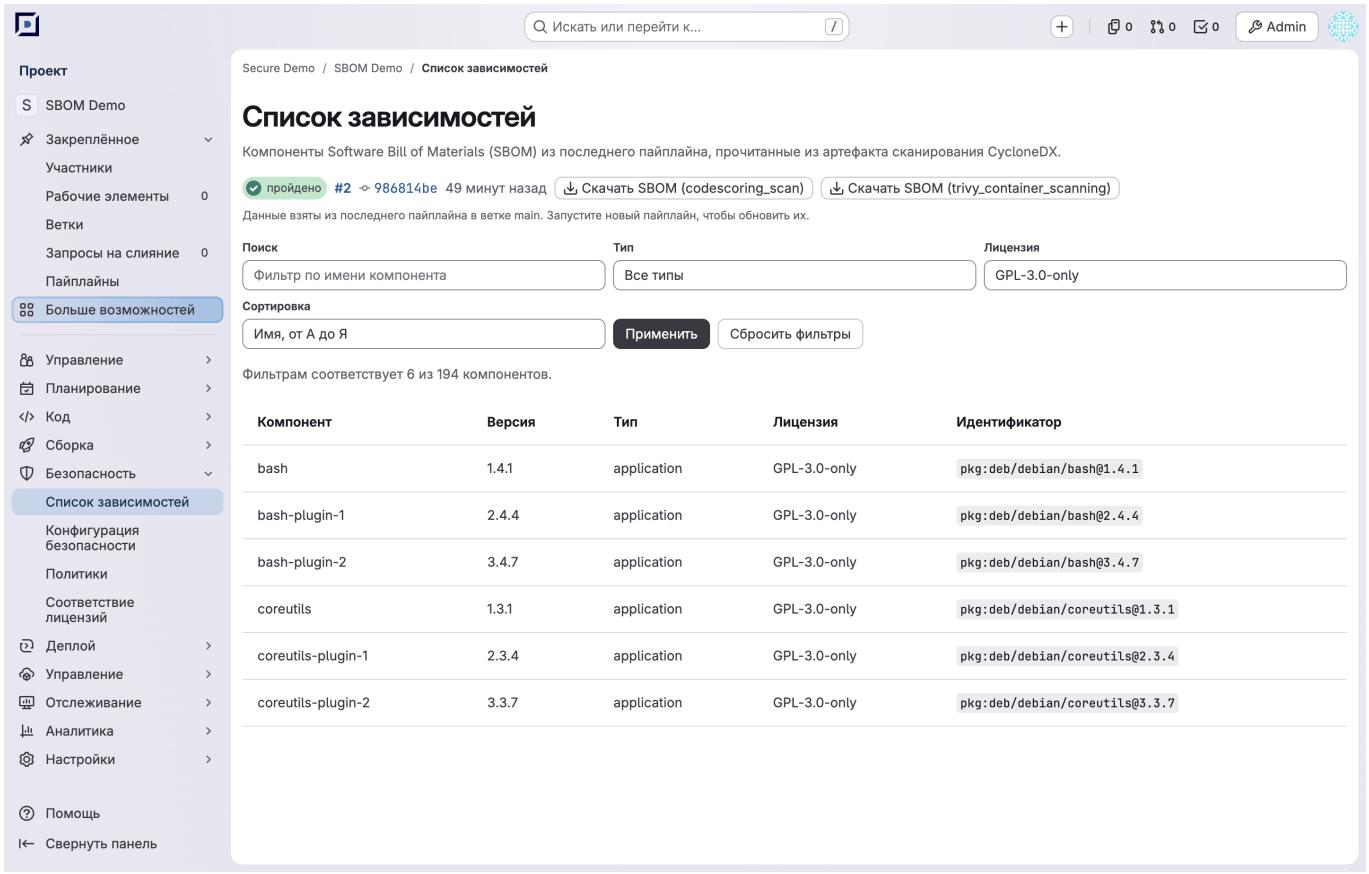
Столбец	Что содержит
«Компонент»	Имя компонента из SBOM
«Версия»	Версия компонента
«Тип»	Тип компонента по CycloneDX: `library`, `application`, `container` и другие
«Лицензия»	Лицензии, объявленные компонентом, через запятую; «Неизвестно», если он не объявил ни одной
«Идентификатор»	Идентификатор пакета в формате purl

На страницу выводится 50 строк. Над таблицей стоит количество компонентов в SBOM, а при заданном фильтре — «Фильтрам соответствует 6 из 194 компонентов.»

Элементы управления над таблицей:

Элемент	Что делает
«Поиск»	Оставляет компоненты, в имени которых есть введённый текст, без учёта регистра
«Тип»	Оставляет компоненты одного типа
«Лицензия»	Оставляет компоненты под одной лицензией; «Неизвестно» оставляет те, что не объявили ни одной
«Сортировка»	Задаёт порядок строк: «Имя, от А до Я» (по умолчанию), «Имя, от Я до А», «Тип», «Лицензия»

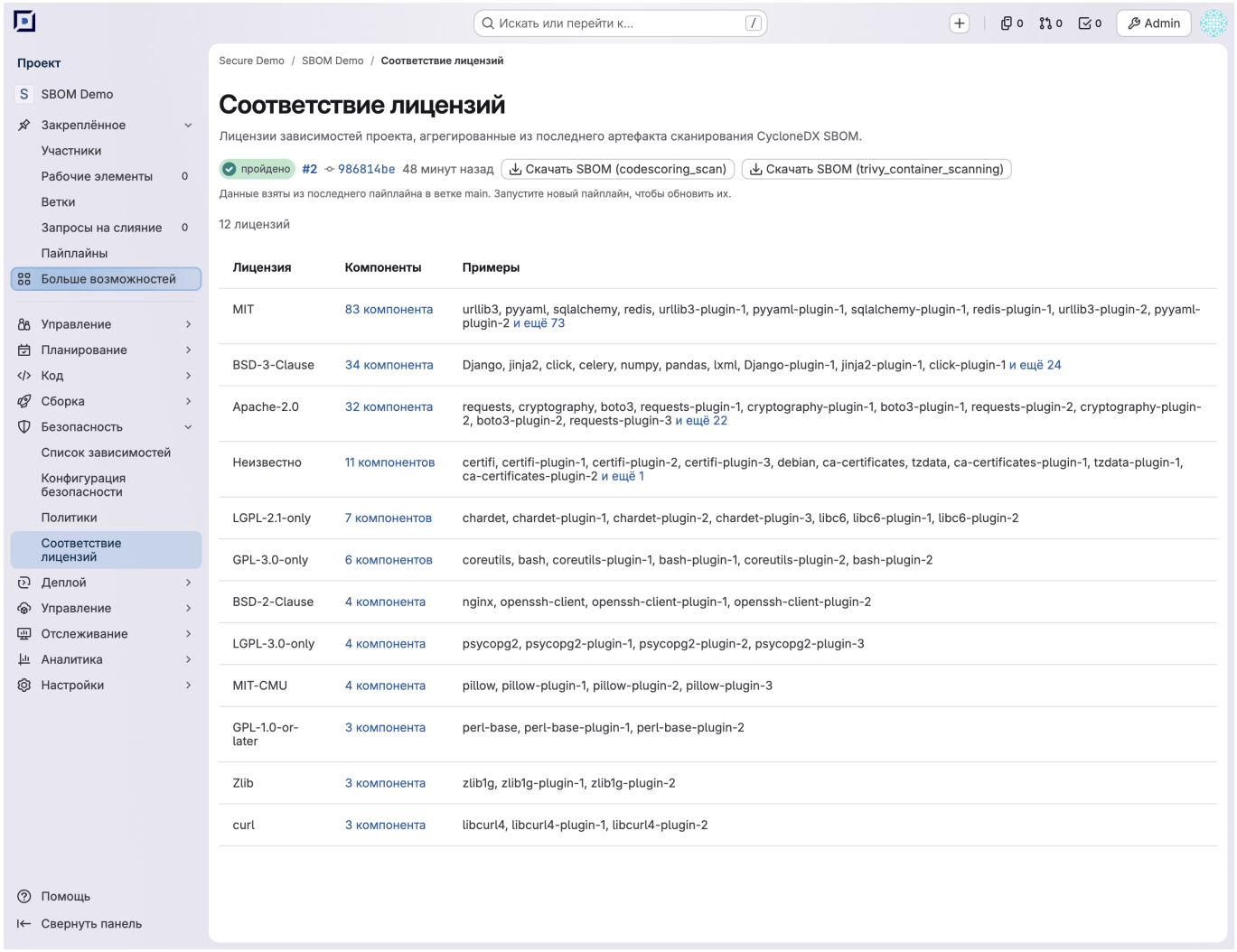
Кнопка «Применить» применяет выбранные значения. Кнопка «Сбросить фильтры» появляется, когда задан хотя бы один фильтр, и возвращает полный список. Списки «Тип» и «Лицензия» строятся по всему SBOM, поэтому сужение одного из них сохраняет варианты другого.



Выбранные значения хранятся в адресе, поэтому отфильтрованный список можно сохранить в закладки или отправить коллеге: `/-/security/dependencies?license=GPL-3.0-only&sort=name_desc`.

Фильтр, которому ничего не соответствует, показывает «Нет зависимостей, соответствующих фильтрам» и подсказку «Измените или сбросьте фильтры, чтобы увидеть больше компонентов.»

Соответствие лицензий



Столбцы таблицы:

Столбец	Что содержит
«Лицензия»	Идентификатор лицензии из SBOM; «Неизвестно» собирает компоненты, которые не объявили лицензию
«Компоненты»	Сколько компонентов под этой лицензией. Число ведёт в список зависимостей, отфильтрованный по ней
«Примеры»	До десяти имён компонентов; ссылка «и ещё N» ведёт в тот же отфильтрованный список

Строки упорядочены по количеству компонентов, от большего к меньшему, а при равном количестве — по алфавиту. Компонент, объявивший несколько лицензий, посчитан под каждой из них, поэтому сумма по строкам больше количества компонентов в SBOM.

«Неизвестно» на обеих страницах содержит одни и те же компоненты, поэтому число в строке и открываемый им отфильтрованный список совпадают.

Состояния отчёта

Что в пайплайне	Что показывают страницы
Артефакта CycloneDX нет	«Пока нет результатов SBOM» и «Запустите пайплайн с включённым сканером CodeScoring, чтобы увидеть здесь зависимости проекта.»
Артефакты есть, но ни один не удалось прочитать	«Не удалось прочитать отчёт SBOM» и «Не удалось прочитать последний артефакт сканирования. Проверьте логи задачи CodeScoring и перезапустите пайплайн.»
Хотя бы один артефакт прочитан	Таблицы. Для SBOM без компонентов — «Зависимости не найдены» или «Лицензии не найдены»

Один нечитаемый артефакт оставляет остальные на месте: страницы показывают то, что сообщили прочитанные. Блок с пайплайном и кнопками скачивания показывается во всех трёх состояниях, поэтому нечитаемый артефакт можно скачать и разобрать.

Диагностика

Страниц нет в разделе «Безопасность»

Проверьте условия из раздела «Условия доступности»: флаг функций, настройку проекта и свою роль в проекте.

«Пока нет результатов SBOM» после успешного пайплайна

Последний пайплайн ветки по умолчанию не загрузил ни одного артефакта с типом отчёта `cyclonedx`. Проверьте, что:

- в действии `codescoring` политики включён модуль «SCA»: задача `codescoring_sca` формируется только в этом случае, а действие `codescoring`, в котором не включён ни модуль «SCA», ни модуль «Секреты», схема политики отвергает, и такая политика целиком ничего не требует;
- пайплайн со сканированием запущен в ветке по умолчанию: пайплайн другой ветки или мерж-реквеста оставляет эти страницы пустыми;
- задача сканирования завершилась и загрузила артефакты — это видно на вкладке «Задания» пайплайна;

- путь в ключе `artifacts:reports:cyclonedx` задачи указывает на файл, который задача записала.

«Не удалось прочитать отчёт SBOM»

Скачайте артефакт из блока с пайплайном и откройте его: страницы читают документ CycloneDX в формате JSON. Задание `codescoring_sca` пишет формат, заданный переменной `FE_SCANS_CODESCORING_BOM_FORMAT`, по умолчанию `cyclonedx_v1_6_json`; другой формат загружается в тот же артефакт и здесь не читается.

Отчёт об уязвимостях

Отчёт об уязвимостях — страница проекта, на которой собраны несоответствия сканеров безопасности из пайплайнов проекта: SAST, обнаружение секретов, сканирование зависимостей, сканирование контейнеров и DAST. На этой странице можно отфильтровать несоответствия пайплайна, открыть подробности несоответствия и проследить, как меняется число несоответствий со временем.

Доступность

Пункт «Отчёт об уязвимостях» появляется в боковом меню проекта, если выполнены все условия:

- В настройках проекта включены функции «Безопасность и соблюдение норм» и «CI/CD».
- У вас есть роль «Разработчик», «Менеджер безопасности», «Мейнтейнер» или «Владелец» в проекте.

Изменять настройки отчёта могут только пользователи с ролью «Мейнтейнер» или «Владелец».

Открытие отчёта об уязвимостях

Чтобы открыть отчёт об уязвимостях:

1. Откройте проект.
2. В боковом меню слева выберите «Безопасность» → «Отчёт об уязвимостях».

На странице показан последний пайплайн, сформировавший отчёт безопасности. Несоответствия пайплайна появляются сразу после завершения его задач сканирования, поэтому у выполняющегося пайплайна результаты могут быть неполными.

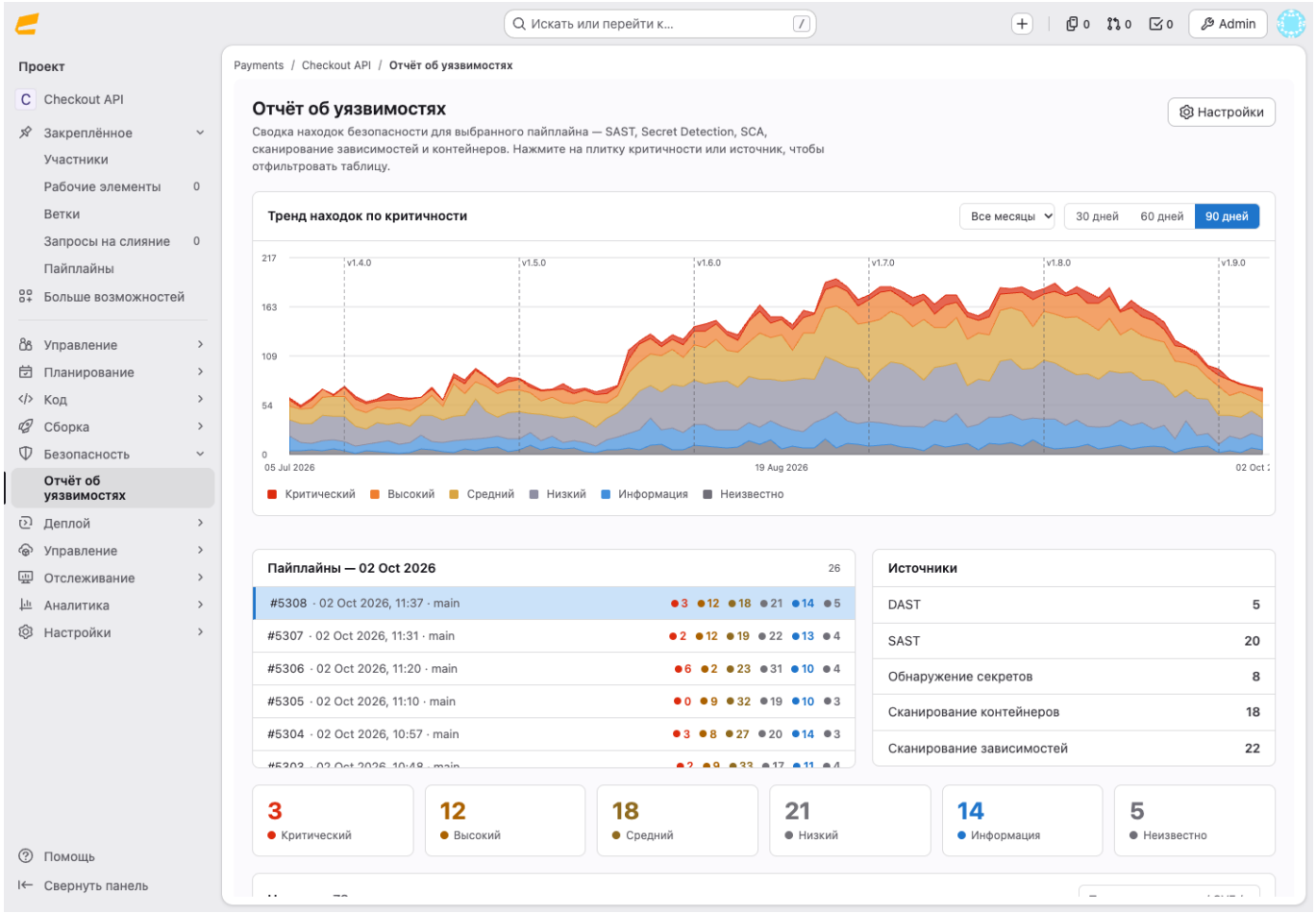


Таблица несоответствий

В таблице несоответствий перечислены несоответствия выбранного пайплайна. В столбце «Описание» показаны название несоответствия и его местоположение, например путь к файлу, затронутые пакеты или идентификатор CVE. У несоответствия могут быть значки:

- **×N** — о несоответствии сообщили N раз в этом пайплайне, например несколько сканеров;
- **N** — то же несоответствие встречается ещё в N других пайплайнах проекта.

На панели «Источники» перечислены типы отчётов пайплайна и число несоответствий каждого типа. Если сканеры сообщили об одном и том же несоответствии несколько раз, на плитке уровня критичности или у типа отчёта показаны два числа, например **12 / 15**: число несоответствий и число записей сканеров до объединения дубликатов.

Чтобы отфильтровать таблицу несоответствий:

- Выберите плитку уровня критичности над таблицей или тип отчёта на панели «Источники» либо в строке таблицы. Чтобы снять фильтр, выберите его повторно.
- Введите текст в поле «Поиск по описанию / CVE / пути…». Поиск идёт по названию, местоположению, затронутым пакетам, CVE или идентификатору, типу отчёта и инструменту несоответствия.

Чтобы снять все фильтры и очистить поле поиска, нажмите «× Сбросить фильтры».

Выбор пайплайна

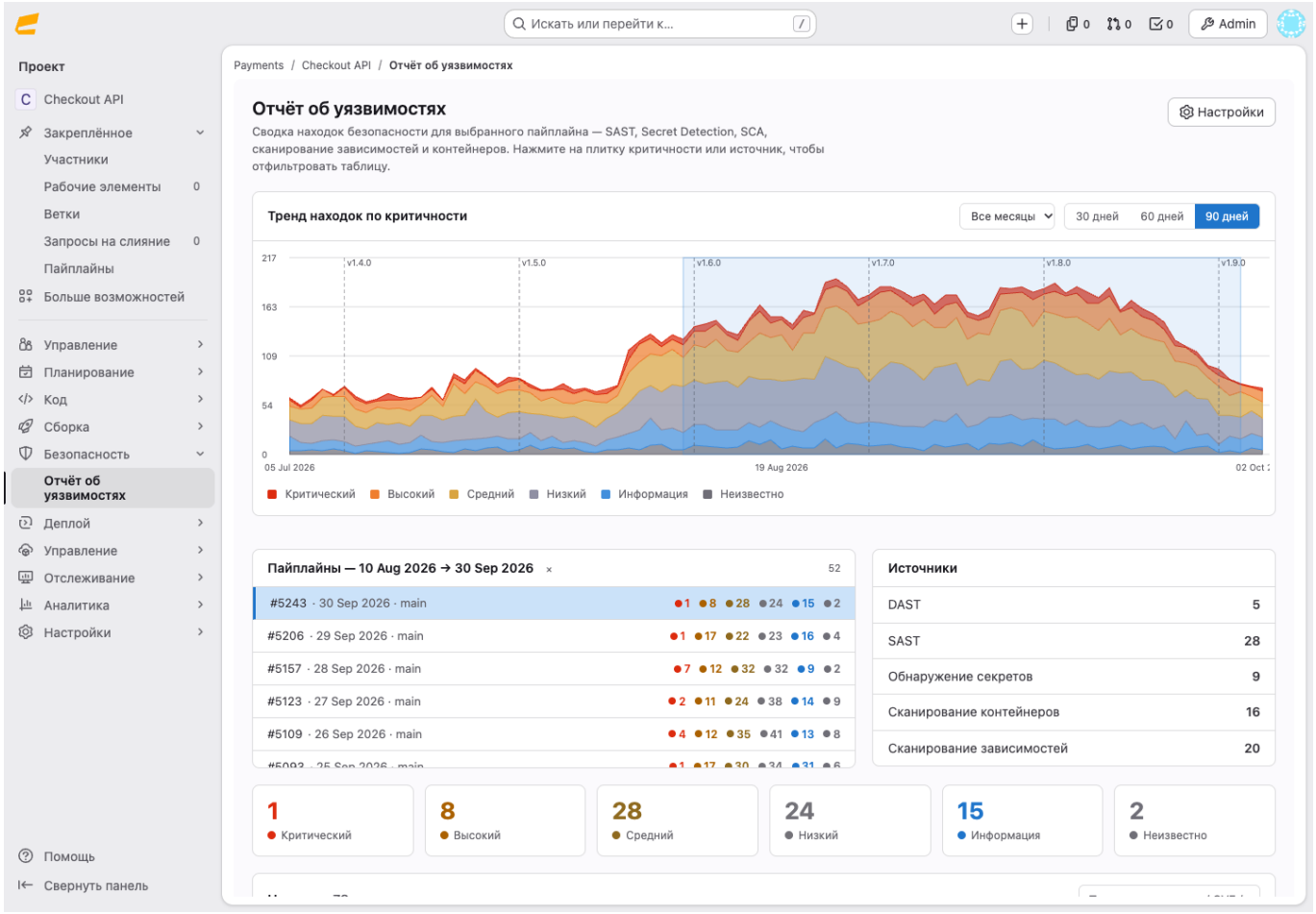
График «Тренд находок по критичности» показывает число несоответствий по дням с разбивкой по уровням критичности. Каждый день на графике соответствует последнему пайплайну этого дня в ветке по умолчанию, а если в ней пайплайнов не было — последнему пайплайну дня. Чтобы изменить период графика, выберите «30 дней», «60 дней» или «90 дней» (по умолчанию) либо месяц в списке «Все месяцы». Пунктирные вертикальные линии на графике отмечают релизы или этапы проекта — в зависимости от настроек отчёта.

На панели «Пайплайны» перечислены пайплайны выбранного дня, сформировавшие отчёт безопасности, от новых к старым. В каждой строке указаны номер пайплайна, время, ветка или тег и число несоответствий каждого уровня критичности. Чтобы посмотреть несоответствия другого пайплайна, выберите его строку.

Чтобы выбрать день или период:

- **Выбор дня**: щёлкните день на графике. Отчёт покажет последний пайплайн этого дня в ветке по умолчанию.
- **Выбор периода**: выделите период на графике, удерживая кнопку мыши. На панели «Пайплайны» появится по одной строке на каждый день периода.
- **Сброс выбора**: нажмите **×** рядом с заголовком «Пайплайны» или клавишу `Escape`.

Адрес страницы сохраняет выбранный день или период, поэтому им можно поделиться ссылкой.

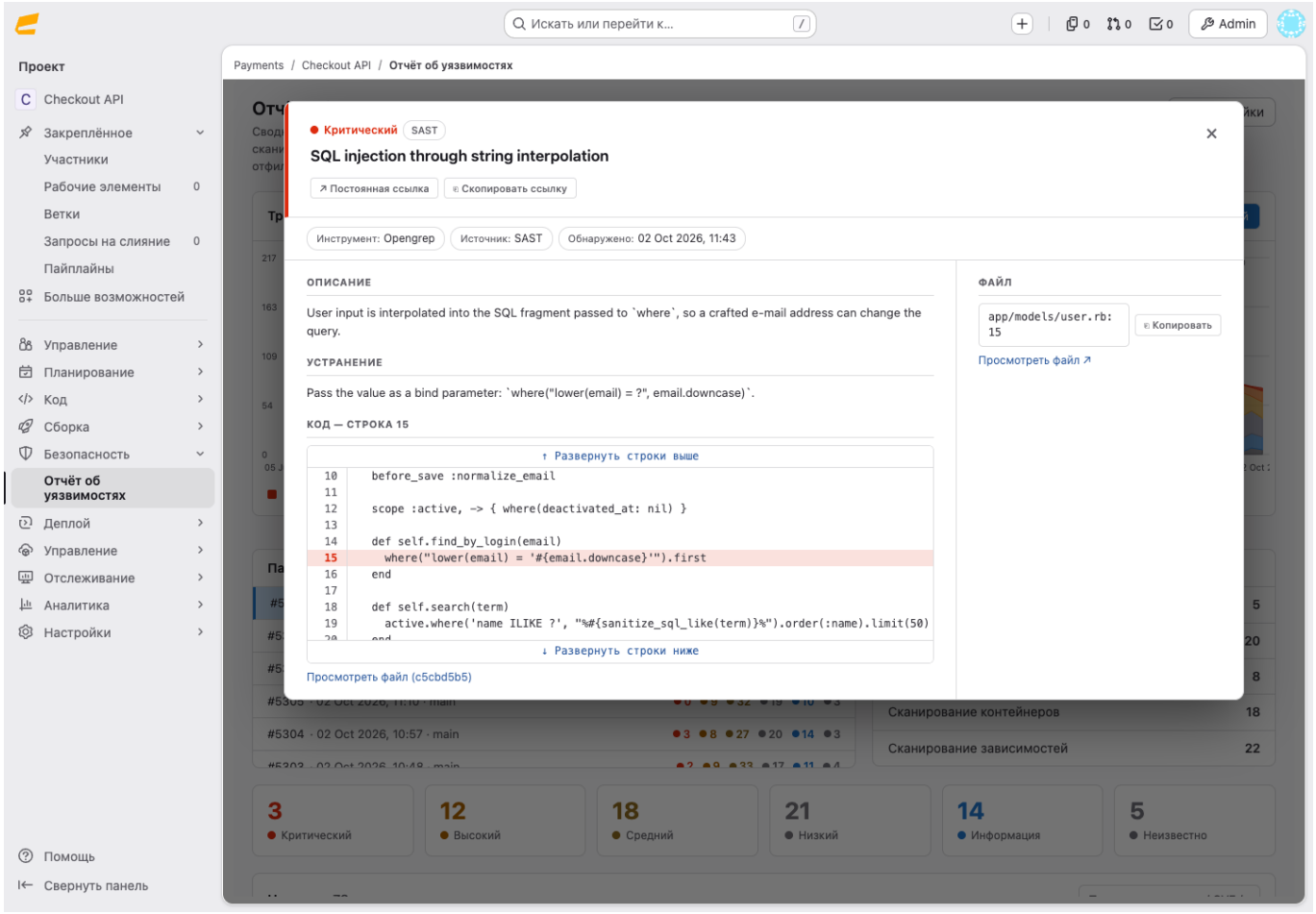


Подробности несоответствия

Чтобы открыть подробности несоответствия, выберите его строку в таблице несоответствий. В окне подробностей показаны данные, которые отчёт безопасности содержит о несоответствии, например файл или местоположение, затронутые пакеты, идентификаторы, ссылки, описание и способ устранения. Идентификатор CVE, для которого в отчёте нет ссылки, ведёт на запись в NVD (National Vulnerability Database).

В окне подробностей можно выполнить следующие действия:

- Чтобы скопировать путь к файлу, нажмите «Копировать» рядом с ним. Чтобы открыть файл, нажмите «Просмотреть файл».
- Чтобы показать больше кода вокруг несоответствия, нажмите «Развернуть строки выше» или «Развернуть строки ниже». Код показывается, если его показ включён в настройках отчёта.
- Чтобы поделиться несоответствием, нажмите «Постоянная ссылка» или «Скопировать ссылку». Если на графике не выбран день, ссылка открывает последний пайплайн, а он меняется после завершения новых пайплайнов.



Настройки отчёта об уязвимостях

Чтобы изменить настройки отчёта:

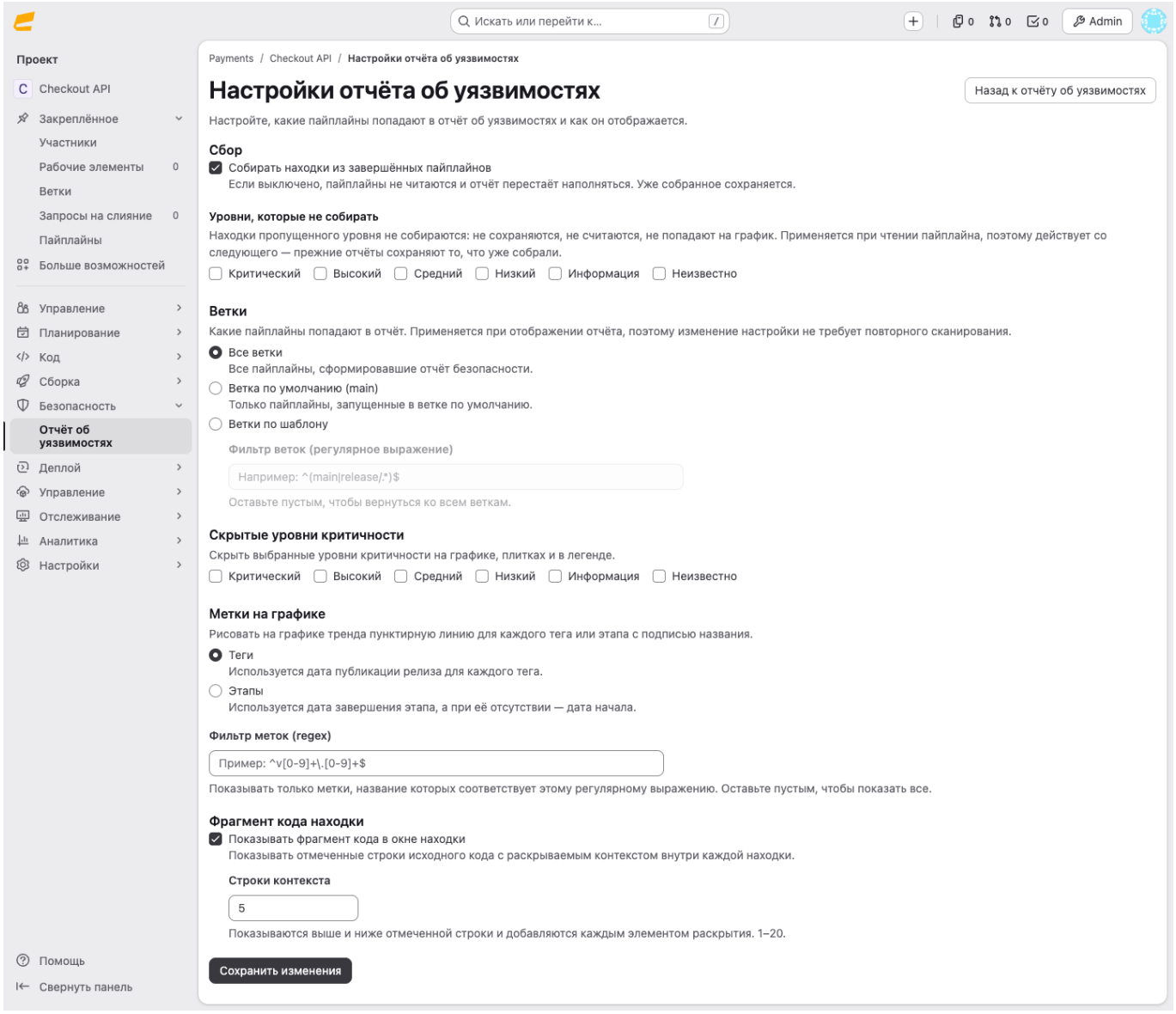
1. Откройте отчёт об уязвимостях.
2. В правом верхнем углу отчёта нажмите «Настройки». Откроется страница «Настройки отчёта об уязвимостях».
3. Измените настройки и нажмите «Сохранить изменения».

Чтобы вернуться к отчёту, нажмите «Назад к отчёту об уязвимостях».

Доступны следующие настройки:

Настройка	Описание	По умолчанию
«Собирать находки из завершённых пайплайнов»	Сбор несоответствий из новых пайплайнов. Если сбор выключен, отчёт сохраняет уже собранные несоответствия	Включено
«Уровни, которые не собирать»	Уровни критичности, несоответствия которых не собираются из новых пайплайнов. Уже собранные несоответствия остаются в отчёте	Не выбраны
«Ветки»	Ветки, пайплайны которых показывает отчёт: «Все ветки», «Ветка по умолчанию» или «Ветки по шаблону». Настройка действует и на уже собранные пайплайны	«Все ветки»
«Фильтр веток (регулярное выражение)»	Регулярное выражение RE2 для названий веток, используется с вариантом «Ветки по шаблону». Пустое выражение соответствует всем веткам. Некорректное выражение сохранить нельзя	Пусто
«Скрытые уровни критичности»	Уровни критичности, скрытые на графике, плитках и в легенде. Скрытые уровни также не учитываются в счётчиках на панели «Пайплайны». Несоответствия скрытого уровня остаются в таблице несоответствий с уровнем «Неизвестно». Если скрыт и уровень «Неизвестно», они	Не выбраны

Настройка	Описание	По умолчанию
	получают самый высокий уровень, который не скрыт	
«Метки на графике»	Что отмечают пунктирные линии на графике: «Теги» — релизы проекта по дате релиза, «Этапы» — этапы по дате завершения или, если её нет, по дате начала	«Теги»
«Фильтр меток (regex)»	Регулярное выражение RE2 для названий релизов или этапов. При пустом выражении показываются все метки. Некорректное выражение сохранить нельзя	Пусто
«Показывать фрагмент кода в окне находки»	Показ кода несоответствия в окне подробностей	Включено
«Строки контекста»	Число строк кода выше и ниже несоответствия, от 1 до 20	5



Интеграции

Вебхуки

Вебхуки — это событийно-ориентированный механизм интеграции с внешними системами. Они позволяют автоматически отправлять HTTP-запросы при возникновении определённых событий в Deckhouse Code:

- Поддержка широкого спектра событий: Push, Merge Request, Issue, Pipeline, Release и др.
- Настройка запросов: выбор метода (POST, PUT), формат JSON, настройка заголовков.
- Безопасность: Secret Token, SSL/TLS, фильтрация по событиям.
- Поддержка на уровне проектов, групп и всей инстанции.
- Интеграция с CI/CD, мониторингом, мессенджерами и системами трекинга.
- Механизм повторных попыток при сбоях соединения (Retry).

Вебхуки проектов доступны в Deckhouse Code.

Вебхуки групп

Чтобы добавить вебхук на уровне группы, откройте страницу группы и перейдите в «Настройки» → «Вебхуки». Далее выберите интересующие события. Вебхуки групп поддерживают все события из проектов, а также:

- События участников;
- События проектов;
- События подгрупп.

События участников срабатывают при создании, изменении или удалении участников группы или проекта.

Если у пользователя не указан публичный email, в теле запроса email будет отображаться как `"[УДАЛЕНО]"`.

Создание вебхуков групп

Заголовок запроса:

```
X-Gitlab-Event: Member Hook
```

Пример запроса:

```
{
  "created_at": "2025-07-02T15:23:25Z",
  "updated_at": "2025-07-02T15:35:51Z",
  "group_name": "agriculture",
  "group_path": "agriculture",
  "group_id": 1130,
  "user_username": "reported_user_barabara",
  "user_name": "Estella Gleason",
  "user_email": "[УДАЛЕНО]",
  "user_id": 58,
  "group_access": "Guest",
  "expires_at": "2025-07-09T00:00:00Z",
  "event_name": "user_add_to_group"
}
```

Изменение вебхуков групп

Заголовок запроса:

```
X-Gitlab-Event: Member Hook
```

Пример запроса:

```
{
  "created_at": "2025-07-02T15:23:25Z",
  "updated_at": "2025-07-02T15:36:21Z",
  "group_name": "agriculture",
  "group_path": "agriculture",
  "group_id": 1130,
  "user_username": "reported_user_barabara",
  "user_name": "Estella Gleason",
  "user_email": "[УДАЛЕНО]",
  "user_id": 58,
  "group_access": "Guest",
  "expires_at": null,
  "event_name": "user_update_for_group"
}
```

Удаление вебхуков групп

Заголовок запроса:

```
X-Gitlab-Event: Member Hook
```

Пример запроса:

```
{
  "created_at": "2025-07-02T15:23:25Z",
  "updated_at": "2025-07-02T15:36:21Z",
  "group_name": "agriculture",
  "group_path": "agriculture",
  "group_id": 1130,
  "user_username": "reported_user_barabara",
  "user_name": "Estella Gleason",
  "user_email": "[УДАЛЕНО]",
  "user_id": 58,
  "group_access": "Guest",
  "expires_at": null,
  "event_name": "user_remove_from_group"
}
```

События проекта

Срабатывает при создании или удалении проектов в группе и подгруппах.

Создание вебхуков проекта

Заголовок запроса:

```
X-Gitlab-Event: Project Hook
```

Пример запроса:


```
{
  "event_name": "project_create",
  "created_at": "2025-07-02T15:40:09Z",
  "updated_at": "2025-07-02T15:40:09Z",
  "name": "rspec",
  "path": "rspec",
  "path_with_namespace": "flant-development/agriculture/rspec",
  "project_id": 28,
  "project_namespace_id": 1130,
  "owners": [
    {
      "name": "Administrator",
      "email": "[УДАЛЕНО]"
    }
  ],
  "project_visibility": "private"
}
```


Удаление вебхуков проекта

Заголовок запроса:

```
X-Gitlab-Event: Project Hook
```

Пример запроса:

```
{
  "event_name": "project_destroy",
  "created_at": "2025-07-02T15:40:09Z",
  "updated_at": "2025-07-02T15:42:04Z",
  "name": "rspec",
  "path": "rspec",
  "path_with_namespace": "flant-development/agriculture/rspec",
  "project_id": 28,
  "project_namespace_id": 1130,
  "owners": [
    {
      "name": "Administrator",
      "email": "[REDACTED]"
    }
  ],
  "project_visibility": "private"
}
```


События подгрупп

Срабатывает при создании или удалении подгруппы.

Создание вебхуков подгрупп

Заголовок запроса:

```
X-Gitlab-Event: Subgroup Hook
```

Пример запроса:


```
{
  "created_at": "2025-07-02T15:44:02Z",
  "updated_at": "2025-07-02T15:44:02Z",
  "event_name": "subgroup_create",
  "name": "finances",
  "path": "finances",
  "full_path": "flant-development/finances",
  "group_id": 1659,
  "parent_group_id": 1123,
  "parent_name": "Flant development",
  "parent_path": "flant-development",
  "parent_full_path": "flant-development"
}
```


Удаление вебхуков подгрупп

Заголовок запроса:

```
X-Gitlab-Event: Subgroup Hook
```

Пример запроса:

```
{
  "created_at": "2025-07-02T15:44:02Z",
  "updated_at": "2025-07-02T15:44:02Z",
  "event_name": "subgroup_destroy",
  "name": "finances",
  "path": "finances",
  "full_path": "flant-development/finances",
  "group_id": 1659,
  "parent_group_id": 1123,
  "parent_name": "Flant development",
  "parent_path": "flant-development",
  "parent_full_path": "flant-development"
}
```

Интеграции со сканерами безопасности

Deckhouse Code добавляет собственные интеграции с инструментами, которые выполняют обязательные сканирования безопасности, и с системой, которая собирает их находки. Интеграция сканера хранит то, что нужно задаче сканирования и чего политика в репозитории хранить не должна: адрес сервера, токен, доверие сертификату и значения по умолчанию для каждого сканирования проекта. Интеграция DefectDojo хранит подключение к серверу, на который отправляются отчёты безопасности пайплайна.

Интеграция	Что обслуживает	Где настраивается
«CodeScoring»	Сканирование `codescoring`	Проект или группа
«Positive Technologies Application Inspector»	Сканирование `ptai`	Проект или группа
«DefectDojo»	Каждое сканирование, публикующее отчёт безопасности	Инстанс, группа или проект

Интеграция проекта или группы открывается из «Настройки» → «Интеграции». Интеграция группы действует для каждого проекта под ней, который не задал собственную. У Opengrep и Trivy, сканеров обязательных сканирований SAST и Trivy, настраивать нечего, и в этих списках их нет. Интеграции CodeScoring и PT AI доступны, пока включён флаг функций `fe_security_scan_policies`, а интеграция DefectDojo — пока включён флаг функций `defect_dojo_integration`; как администратор включает флаги, описано на странице Ранний доступ.

Само сканирование задаётся в политике выполнения сканирования: какие модули работают, что сканируется и что роняет пайплайн. Редактор политик предлагает сканирования PT AI и CodeScoring, только пока интеграция этого сканера активна для проекта или для группы выше него; недоступный вариант ведёт ссылкой на страницу интеграции. Переменные интеграции попадают в каждую задачу пайплайна; её токены попадают только в задачу сканирования, которую политика предписала для этого сканера. Переменная, заданная политикой, имеет приоритет над той же переменной из интеграции. Редактор политик и параметры SAST-сканирования описаны на странице SAST-сканирование.

Opengrep

Opengrep — сканер, которым выполняется обязательное SAST-сканирование `opengrep_sast`. Это консольная утилита, работающая внутри задачи: ей не нужен ни сервер, ни токен, ни доверие сертификату. Всё, что касается сканирования, включая образ, в котором оно выполняется, принадлежит политике, поэтому интеграции для настройки нет: «Opengrep» отсутствует в списках «Интеграции», на страницах настроек и в REST API, а задачи сканирования не получают переменных из записи интеграции.

Образ, в котором выполняется сканирование, сервер определяет при формировании задачи и записывает в неё как значение, а не как переменную. Сервер берёт:

- ссылку из поля «Образ сканера» действия `opengrep_sast`, если политика её называет;
- иначе образ из переменной окружения инстанса `FE_SCANS_OPENGREP_IMAGE`;
- иначе образ, зафиксированный этим релизом, — `opengrep-scan:1.30.0-fe2` из хранилища образов продукта.

Образ никогда не становится переменной CI/CD, поэтому собственные настройки проверяемого проекта не могут направить сканирование на другой сканер. Поле «Образ сканера» в политике показывает образ, который выполнится при пустом поле, а подсказка рядом называет версию Opengrep и набор правил из поставки продукта.

Правила, уровни правил, исключаемые категории, порог блокировки, игнорируемые пути и дополнительные параметры — это группы «Правила», «Уровни правил», «Исключаемые категории», «Порог блокировки», «Игнорируемые пути» и «Дополнительные параметры» того же действия. Сканирование, его задачи и эти группы описаны на странице SAST-сканирование.

Trivy

Trivy — сканер, которым выполняется обязательное сканирование `trivy`. Он работает внутри задачи и не обращается к собственному серверу, поэтому интеграции для настройки нет: «Trivy» отсутствует в списках «Интеграции», на страницах настроек и в REST API, а задачи сканирования не получают переменных из записи интеграции.

Всё, что касается сканирования, принадлежит политике: модули, их параметры и образ, в котором сканирование выполняется. Образ сервер записывает в задачу при её формировании: ссылку из поля «Образ сканирования» действия `trivy` или, если политика её не называет, образ,

зафиксированный этим релизом, — `trivy-with-db:0.74.0` из хранилища образов продукта. Учётные данные хранилища образов модуль образов контейнеров берёт из переменных CI/CD сканируемого проекта.

Сканирование, его модули и их параметры описаны на странице Сканирование с помощью Trivy.

CodeScoring

Полная настройка, модули SCA и «Секреты» и диагностика этой интеграции описаны на странице «Интеграция с CodeScoring».

Интеграция CodeScoring подключает проект к серверу CodeScoring и доставляет агент johnny в задачи сканирования `codescoring`. У сканирования два модуля, каждый включается в политике: анализ состава, который выполняет задача `codescoring_sca` и который публикует отчёт о сканировании зависимостей, отчёт о качестве кода, отчёт о тестах и CycloneDX SBOM `gl-sbom.cdx.json`; и поиск секретов, который выполняют задачи `codescoring_secrets_gitleaks`, `codescoring_secrets_trufflehog` и `codescoring_secrets_kingfisher`, по одной на движок, и который публикует отчёт об обнаружении секретов. Находки попадают в отчёт об уязвимостях, список зависимостей и DefectDojo наравне с находками любого другого сканера; SBOM читают страницы Список зависимостей и соответствие лицензий.

Интеграция настраивается на проекте или группе в «Настройки» → «Интеграции» → «CodeScoring». Редактор политик предлагает сканирование `codescoring`, пока интеграция активна для проекта или группы выше него. Форма содержит подключение, доверие и значения по умолчанию для задач сканирования; что и как сканирует агент, задаётся в политике:

Поле	Содержимое
«URL-адрес сервера»	Базовый URL сервера CodeScoring, например `https://codescoring.example.com`. Обязательное. Раздел «Подключение».
«Токен API»	Токен, с которым задачи аутентифицируются и по которому загружают агент johnny. Обязательное. Раздел «Подключение».
«Стадия сканирования»	Стадия политики, под которой сканирование отчитывается на сервере: build, dev, source, stage, test или prod. По умолчанию — build. Агент принимает одну стадию, поэтому сканирование отчитывается под одной. Раздел «Подключение».
«Доверие к сертификату»	Как задача сканирования решает, доверять ли серверу: «Доверять тому, чему уже доверяет задача» (по умолчанию), «Доверять удостоверяющему центру ниже», «Не проверять сертификат». Настройка относится к запросам, которые задача делает через curl, — загрузке агента — и к проверке соединения; сам агент johnny сертификат не проверяет и не даёт способа это включить. Раздел «Доверие сертификату».
«Удостоверяющий центр»	PEM-сертификат центра, подписавшего сертификат сервера. Читается только при варианте «Доверять удостоверяющему центру ниже» и тогда обязателен. Раздел «Доверие сертификату».
«Образ задачи сканирования»	Образ, в котором выполняются задачи сканирования CodeScoring, с хранилищем образов и тегом или дайджестом. В нём нужны curl, ca-certificates, jq, ruby и git; агент загружается в него во время выполнения. Пустое поле: `codescoring-scan:2026.08.30` из хранилища образов продукта, образ из поставки этого релиза. Раздел «Задачи сканирования».

Поле	Содержимое
«Формат SBOM»	Версия CycloneDX, в которой записывается SBOM: `cyclonedx_v1_4_json`, `cyclonedx_v1_5_json`, `cyclonedx_v1_6_json`, `cyclonedx_v1_6_ext_json` (расширение 1.6 от CodeScoring с его свойствами по ГОСТ) или `cyclonedx_v1_7_json`. Все версии перечисляют одни и те же компоненты. Пустое поле здесь и в политике: CycloneDX 1.6 JSON. Раздел «Задачи сканирования».

«Проверить настройки» запрашивает у сервера версию агента с токеном и настройкой доверия из формы, так что адрес, токен и сертификат проверяются разом.

Задачи получают `FE_SCANS_CODESCORING_URL`, `FE_SCANS_CODESCORING_TLS_MODE` и, если назван удостоверяющий центр, `FE_SCANS_CODESCORING_CA_CERT`; `FE_SCANS_CODESCORING_SCAN_STAGE` и `FE_SCANS_CODESCORING_BOM_FORMAT` — когда поля заполнены. `FE_SCANS_CODESCORING_TOKEN` маскируется и попадает только в предписанные политикой задачи CodeScoring. Каждая задача загружает агент с сервера при каждом запуске, поэтому версия агента — та, которую публикует сервер. Анализ состава отчитывается в проект CodeScoring, названный по slug пути проекта, и создаёт его, если такого нет.

«Образ задачи сканирования» читается при формировании задачи и записывается в неё как значение, а не как переменная; образ называет только интеграция группы, которая предписывает сканирование, — группы, к которой привязана политика, или группы выше неё, — а значение в собственной интеграции проверяемого проекта игнорируется. «Стадия сканирования» и «Формат SBOM» — значения по умолчанию для проекта: политика, называющая собственное значение, имеет приоритет, а список «Формат SBOM» в политике содержит вариант «Взять из интеграции», чтобы вернуть решение интеграции.

Группа «SCA» в политике задаёт режим и цель сканирования, игнорируемые пути, разрешение дерева зависимостей на сервере, стадию, формат SBOM и дополнительные аргументы агента; группа «Секреты» выбирает движки — gitleaks, trufflehog, kingfisher — и задаёт каждому конфигурацию и список исключений.

PT AI

Полная настройка, параметры сканирования и диагностика этой интеграции описаны на странице «Интеграция с PT AI».

Интеграция «Positive Technologies Application Inspector» обслуживает сканирование `ptai`. Его задача `ptai_scan` загружает исходный код коммита на сервер PT AI клиентом aictl, в проект, названный полным путём проекта с `:` вместо `/`, ждёт окончания сканирования и публикует отчёт о тестах и отчёт безопасности SAST, поэтому находки попадают в отчёт об уязвимостях. Белый и

чёрный ящик — части одного сканирования: модули белого ящика работают всегда, а адрес сайта добавляет к ним чёрный ящик. Порога по уровню находок нет: задача завершается с ошибкой, когда сканирование не удалось или политика безопасности проекта на сервере PT AI отклонила его.

Интеграция настраивается на проекте или группе в «Настройки» → «Интеграции» → «Positive Technologies Application Inspector» и хранит только подключение: «URL сервера», «API-токен», «Доверие к сертификату» с «Удостоверяющим центром», а также раздел «Джобы сканирования» с образом задачи и переменными логина и пароля реестра. Раздел «Джобы сканирования» читается из интеграции группы, которой принадлежит политика; подключение — из интеграции проверяемого проекта. Задача получает `FE_SCANS_PTAI_SERVER_URL`, `FE_SCANS_PTAI_TLS_MODE` и, если назван удостоверяющий центр, `FE_SCANS_PTAI_CA_CERT`; `FE_SCANS_PTAI_TOKEN` маскируется и попадает только в предписанную политикой задачу `ptai_scan`.

Что сканируется, задаёт политика: модули и языки, `.gitignore`, загрузка зависимостей, собственный конфигурационный файл и чёрный ящик со входом на сайт и прокси. Пароли и cookie политика называет ссылками `$NAME` на переменные CI/CD проверяемого проекта. Параметры интеграции в политике не принимаются.

DefectDojo

Полная настройка, сопоставление сущностей и диагностика этой интеграции описаны на странице «Интеграция с DefectDojo».

Интеграция DefectDojo отправляет отчёты безопасности пайплайна на сервер DefectDojo. Отчёты уходят в том виде, в каком их опубликовали задачи, в собственных форматах отчётов, через эндпоинт повторного импорта сервера:

Тип отчёта артефакта	Тип сканирования в DefectDojo
`sast`	GitLab SAST Report
`secret_detection`	GitLab Secret Detection Report
`dependency_scanning`	GitLab Dependency Scanning Report
`container_scanning`	GitLab Container Scan
`dast`	GitLab DAST Report

Обязательные сканирования публикуют эти типы — Opengrep и PT AI отчёт `sast`, Trivy `dependency_scanning`, `secret_detection` и `container_scanning`, CodeScoring `dependency_scanning` и `secret_detection`, — и задача из собственного `.gitlab-ci.yml` проекта, публикующая такой отчёт, выгружается так же. Отчёт PT AI не импортируется: PT AI записывает в него `scan.status: failure` даже для завершённого сканирования.

Интеграция настраивается на проекте или группе в «Настройки» → «Интеграции» → «DefectDojo», а для всего инстанса — в «Admin» → «Настройки» → «Интеграции» → «DefectDojo». Её «Параметры импорта по умолчанию» — это соответствие, которое использует каждая выгрузка:

Поле	Содержимое
«URL DefectDojo»	Базовый URL сервера DefectDojo, например `https://defectdojo.example.com`. Обязательное. Раздел «Соединение».
«API-токен»	Токен, с которым аутентифицируются выгрузки. Обязательное. Раздел «Соединение».
«Доверие сертификату»	Как инстанс решает, доверять ли серверу: «Доверять тому, чему уже доверяет этот экземпляр» (по умолчанию), «Доверять удостоверяющему центру ниже», «Не проверять сертификат». Соединение устанавливается с инстанса, поэтому проверку можно выключить; выгрузка тогда открыта для перехвата, а отчёты содержат всё, что нашли сканеры. Раздел «Доверие сертификату».
«Удостоверяющий центр»	PEM-сертификат центра, подписавшего сертификат сервера. Читается только при варианте «Доверять удостоверяющему центру ниже» и тогда обязателен. Раздел «Доверие сертификату».
«Название продукта»	Продукт DefectDojo, в который попадают отчёты. Пустое поле: полный путь проекта. Раздел «Параметры импорта по умолчанию».
«Тип продукта»	Тип продукта, в котором создаётся продукт, если его нет. Раздел «Параметры импорта по умолчанию».
«Название активности»	Активность DefectDojo, в которую попадают отчёты. Пустое поле: ветка или тег, на которых выполнялся пайплайн. Раздел «Параметры импорта по умолчанию».
«Минимальная критичность»	Находки ниже неё не попадают в импорт: `Info` (по умолчанию), `Low`, `Medium`, `High`, `Critical`. Раздел «Параметры импорта по умолчанию».

Поле	Содержимое
«Автоматически создавать контекст»	«Создать продукт и активность, если они не существуют». Включено по умолчанию. Раздел «Параметры импорта по умолчанию».
«Импортированные находки»	«Помечать импортированные находки как активные». Включено по умолчанию. Раздел «Параметры импорта по умолчанию».
«Подтверждённые находки»	«Помечать импортированные находки как подтверждённые». Выключено по умолчанию. Раздел «Параметры импорта по умолчанию».
«Закрывать старые находки»	«Закрывать находки, которые больше не фиксируются». Включено по умолчанию. Раздел «Параметры импорта по умолчанию».

При сохранении формы с собственными настройками сервер спрашивают, пригодно ли соответствие: отклонённый токен, отсутствующий продукт при выключенном «Автоматически создавать контекст», отсутствующий тип продукта при включённом и отсутствующая активность при выключенном отклоняются каждый с сообщением, называющим продукт или активность. Сервер, которого не удалось спросить, сохранение не блокирует. «Проверить настройки» — кнопка интеграции проекта — задаёт те же вопросы и сообщает также о сервере, которого не удалось спросить.

Выгрузка начинается, когда в проекте с активной интеграцией DefectDojo, собственной или унаследованной, завершается задача, опубликовавшая отчёт одного из типов выше. На пайплайн выполняется одна выгрузка; следующая завершившаяся в том же пайплайне задача повторяет её один раз. Каждый отчёт последнего запуска каждой задачи попадает в продукт и активность из формы, в тест, названный по задаче, с датой завершения задачи, поэтому повторный запуск сканирования обновляет тот же тест, и его находки не заводятся дважды. В активность записываются коммит, ветка, номер пайплайна, короткий коммит как версия и адрес проекта. Отчёт сканирования, которое не выполнилось, — заглушка, которую задача опубликовала, чтобы виджеты пайплайна что-то показывали, — не импортируется, потому что «Закрывать старые находки» иначе закрыло бы каждую находку последнего настоящего сканирования.

Ответ сервера 429 или 5xx и тайм-аут повторяются через 60, 120 и 240 секунд; выгрузка ждёт один отчёт до 600 секунд. Любой другой отказ окончателен, и подействовать на него может только проект: страница пайплайна показывает предупреждение «Предупреждения выгрузки отчётов о безопасности» с одной строкой на каждый отчёт, который не дошёл до сервера:

```
DefectDojo did not receive the report of job "opengrep_sast_junit" (GitLab SAST
Report): DefectDojo returned HTTP 400: Product "group/project" does not exist and no
product_type_name provided to create the new product in
```

Причина — собственное сообщение сервера, до 500 символов, либо одно из `blocked URL: …` для адреса, к которому инстансу запрещено обращаться, `certificate authority: …` для сертификата, который не удалось прочитать, и `the scan did not run, so its placeholder report was not imported`. Когда повторные попытки исчерпаны, страница показывает ещё одну строку: `DefectDojo upload of this pipeline's security reports failed and the retries are spent: …`. Строка об отчёте убирается, когда следующий запуск доставляет этот отчёт. Полный ответ сервера — в журнале приложения.

Интеграция с DefectDojo

Эта интеграция отправляет отчёты сканирования безопасности проекта в DefectDojo, систему управления уязвимостями, и поддерживает актуальность несоответствий в нём по мере выполнения пайплайнов.

Предварительные требования

Перед настройкой интеграции проверьте следующее:

- Администратор включил флаг функций `defect_dojo_integration` так, как описано на странице Ранний доступ.
- DefectDojo доступен из Deckhouse Code. Если его адрес находится в локальной сети, администратор установил флажок «Разрешить запросы к локальной сети от вебхуков и интеграций» или добавил адрес в поле «Локальные IP-адреса и доменные имена, к которым могут получить доступ хуки и интеграторы» в разделе «Admin» → «Настройки» → «Сеть» → «Исходящие запросы».
- У вас есть API-токен учётной записи DefectDojo, которой разрешено импортировать результаты сканирования, просматривать продукты и активности и изменять активности. Если установлен флажок «Автоматически создавать контекст», учётная запись также должна иметь право создавать продукты и активности.
- У вас есть роль «Мейнтейнер» или «Владелец» в проекте.
- В пайплайне есть задачи, которые формируют один из поддерживаемых отчётов безопасности.

Настройка интеграции с DefectDojo

Чтобы настроить интеграцию:

1. Откройте проект в Deckhouse Code.
2. Перейдите в раздел «Настройки» → «Интеграции» → «DefectDojo».
3. В группе «Включить интеграцию» установите флажок «Активные».

4. В блоке «Соединение» укажите в поле «URL DefectDojo» базовый URL инстанса DefectDojo, а в поле «API-токен» — токен для аутентификации в API DefectDojo.
5. В блоке «Доверие сертификату» выберите, как соединение решает, доверять ли серверу DefectDojo:
   - «Доверять тому, чему уже доверяет этот экземпляр» (по умолчанию);
   - «Доверять удостоверяющему центру ниже» — дополнительно доверять центру, чей PEM-сертификат вы вставите в поле «Удостоверяющий центр»;
   - «Не проверять сертификат» — выгрузка отправляется без проверки сертификата сервера и может быть перехвачена вместе со всеми несоответствиями из отчётов.
6. Заполните поля блока «Параметры импорта по умолчанию», которые применяются при импорте отчётов сканирования в DefectDojo:
   - «Название продукта» — продукт DefectDojo. Оставьте пустым, чтобы использовать полный путь проекта;
   - «Тип продукта» — тип продукта DefectDojo, используется при автоматическом создании продукта;
   - «Название активности» — активность DefectDojo. Оставьте пустым, чтобы использовать имя ветки;
   - «Минимальная критичность» — игнорировать несоответствия ниже указанной критичности при импорте (`Info` , `Low` , `Medium` , `High` , `Critical` — от низшей к высшей, по умолчанию `Info`);
   - «Автоматически создавать контекст» — создать продукт и активность, если они не существуют (по умолчанию флажок установлен);
   - «Импортированные находки» — помечать импортированные несоответствия как активные (по умолчанию флажок установлен);
   - «Подтверждённые находки» — помечать импортированные несоответствия как подтверждённые;
   - «Закрывать старые находки» — закрывать несоответствия, которые больше не фиксируются (по умолчанию флажок установлен).
7. Нажмите «Сохранить изменения».

Соединение

Подключитесь к вашему экземпляру DefectDojo.

Включить интеграцию

☑ Активные

URL DefectDojo

https://defectdojo.example.com

Базовый URL вашего экземпляра DefectDojo.

Введите новый API-токен

Оставьте пустым, чтобы использовать текущий API-токен

Доверие сертификату

Как этот экземпляр решает, доверять ли серверу DefectDojo.

Доверие сертификату

- ◉ Доверять тому, чему уже доверяет этот экземпляр
- ○ Доверять удостоверяющему центру ниже
- ○ Не проверять сертификат

Как этот экземпляр решает, доверять ли серверу DefectDojo. В отличие от скана, это соединение устанавливается отсюда, а не из задания, поэтому проверку можно отключить — но тогда выгрузка открыта для перехвата, а в отчётах всё, что нашли анализаторы.

Параметры импорта по умолчанию

Применяются при импорте отчётов сканирования в DefectDojo.

Название продукта

Продукт DefectDojo. Оставьте пустым, чтобы использовать полный путь проекта.

Тип продукта

Payments

Тип продукта DefectDojo, используется при автоматическом создании продукта.

Название активности

Активность DefectDojo. Оставьте пустым, чтобы использовать имя ветки.

Минимальная критичность

Low

Игнорировать находки ниже указанной критичности при импорте.

Автоматически создавать контекст

☑ Создать продукт и активность, если они не существуют

Импортированные находки

☑ Помечать импортированные находки как активные

Подтверждённые находки

☐ Помечать импортированные находки как подтверждённые

Закрывать старые находки

☑ Закрывать находки, которые больше не фиксируются

Сохранить изменения | Проверить настройки | Отмена

Интеграцию можно также настроить на уровне группы или инстанса; проекты этой группы или инстанса наследуют эти настройки, если для них не заданы собственные.

Проверка подключения

Чтобы проверить заполненные в форме настройки до сохранения, нажмите «Проверить настройки». Кнопка доступна только в настройках интеграции проекта. Deckhouse Code отправляет настройки в DefectDojo и проверяет следующее:

- DefectDojo принимает API-токен.
- Продукт существует. Продукт — это значение «Название продукта» или, если оно пустое, полный путь проекта. Если продукта нет, проверка проходит, только когда установлен флажок «Автоматически создавать контекст» и заполнено поле «Тип продукта».
- Активность из поля «Название активности» существует в продукте. Это проверяется, только если поле «Название активности» заполнено, а флажок «Автоматически создавать контекст» снят.

Если все проверки пройдены, выводится сообщение «Соединение успешно.». Иначе выводится сообщение «Ошибка подключения. Проверьте настройки интеграции.» и причина.

При нажатии «Сохранить изменения» в собственных настройках интеграции проекта Deckhouse Code выполняет те же проверки, если интеграция активна и изменились URL, токен, продукт, тип продукта, активность или флажок «Автоматически создавать контекст». Если проверка не пройдена, настройки не сохраняются. Если DefectDojo недоступен или отвечает не так, как ожидается, настройки сохраняются без проверки.

Работа интеграции

Пока интеграция активна, Deckhouse Code отправляет в DefectDojo отчёты безопасности из пайплайнов проекта.

Автоматический импорт результатов сканирования

Когда в пайплайне проекта завершается задача сканирования безопасности, Deckhouse Code импортирует отчёты пайплайна в DefectDojo.

Поддерживаются результаты следующих типов сканирования:

- статическое тестирование безопасности приложений (SAST);
- обнаружение секретов;
- сканирование зависимостей;
- сканирование контейнеров;
- динамическое тестирование безопасности приложений (DAST).

Отчёты отправляет сам Deckhouse Code, поэтому в пайплайне не нужны ни отдельная задача, ни учётные данные DefectDojo в CI-переменных.

Сопоставление сущностей в DefectDojo

По умолчанию Deckhouse Code использует следующие соответствия:

- **Продукт** (Product) — значение «Название продукта» или, если оно пустое, полный путь проекта;
- **Активность** (Engagement) — значение «Название активности» или, если оно пустое, ветка или тег, на которых выполнялся пайплайн. Для пайплайна запроса на слияние это `refs/merge-requests/<IID>/head`, где `<IID>` — номер запроса на слияние;
- **Тест** (Test) — имя CI-задачи, сформировавшей отчёт.

У каждой задачи сканирования свой тест, поэтому сканеры не закрывают несоответствия друг друга. При повторном запуске задачи сканирования импортируется только отчёт её последнего запуска. Если переименовать задачу сканирования, её отчёты перестают импортироваться в тест со старым именем.

Параметры импорта

При импорте результатов Deckhouse Code передаёт в DefectDojo значения из блока «Параметры импорта по умолчанию», а также дату завершения задачи сканирования, идентификатор пайплайна, SHA коммита и его короткую форму в качестве версии, имя ветки или тега и URL проекта. После импорта Deckhouse Code записывает те же SHA коммита, версию, ветку или тег, идентификатор пайплайна и URL проекта в саму активность DefectDojo.

Диагностика

О проблемах интеграции сообщают форма настроек, результат кнопки «Проверить настройки» и страница пайплайна.

Настройки не сохраняются

Если проверка в DefectDojo не пройдена, форма показывает причину и не сохраняет настройки:

Сообщение	Причина	Решение
`DefectDojo отклонил API-токен.`	DefectDojo ответил на токен кодом `401` или `403`	Введите действующий API-токен
`В DefectDojo нет продукта с названием <NAME>. Создайте его в DefectDojo или включите автоматическое создание контекста.`	Продукта нет, флажок «Автоматически создавать контекст» снят	Создайте продукт в DefectDojo или установите флажок «Автоматически создавать контекст»
`Для создания продукта <NAME> в DefectDojo нужно указать тип продукта.`	Продукта нет, поле «Тип продукта» пустое	Заполните поле «Тип продукта»
`В этом продукте DefectDojo нет активности с названием <NAME>. Создайте её в DefectDojo или включите автоматическое создание контекста.`	Активности нет, флажок «Автоматически создавать контекст» снят	Создайте активность в DefectDojo или установите флажок «Автоматически создавать контекст»

Импорт завершается ошибкой

Отчёт, который не попал в DefectDojo, отображается на странице пайплайна в предупреждении «Предупреждения выгрузки отчётов о безопасности» строкой `DefectDojo did not receive the report of job "<JOB_NAME>" (<SCAN_TYPE>): <REASON>`. Строки выводятся на английском языке. `<SCAN_TYPE>` — тип сканирования DefectDojo, например `GitLab SAST Report`, а `<REASON>` — одна из причин:

Причина	Что означает	Решение
`DefectDojo returned HTTP <CODE>` или `DefectDojo returned HTTP <CODE>: <DETAIL>`	DefectDojo отклонил отчёт	Устраните то, что названо в пояснении, например права токена или отсутствующий продукт, и перезапустите задачу сканирования
`blocked URL: <MESSAGE>`	Deckhouse Code не отправляет запросы на этот адрес	Проверьте поле «URL DefectDojo»; для адреса в локальной сети выполните требования из раздела «Предварительные требования»
`certificate authority: The certificate authority could not be read.`	В поле «Удостоверяющий центр» нет читаемого PEM-сертификата	Вставьте PEM-сертификат заново
`the scan did not run, so its placeholder report was not imported`	Сканер задачи не запустился	Исправьте задачу сканирования и перезапустите её

Если DefectDojo недоступен, не отвечает за время тайм-аута или отвечает кодом `429` или `5xx`, выгрузка повторяется до трёх раз с увеличивающейся задержкой. Если все повторы не удались, в предупреждении появляется одна строка для всего пайплайна: `DefectDojo upload of this pipeline's security reports failed and the retries are spent: <REASON>`.

Защита API-токена

Используйте отдельную учётную запись DefectDojo только с правами, перечисленными в разделе «Предварительные требования». Сохранённый токен в форме не отображается. Чтобы заменить его, введите новый токен в поле «Введите новый API-токен»; если оставить поле пустым, сохраняется текущий API-токен.

Интеграция с CodeScoring

CodeScoring — самостоятельно разворачиваемая платформа для анализа состава ПО (SCA, Software Composition Analysis) и поиска секретов. Интеграция с CodeScoring подключает проект к серверу CodeScoring, а политика выполнения сканирования добавляет задачи сканирования CodeScoring в пайплайны проекта и задаёт, что они сканируют.

Предварительные требования

Чтобы сканирование могло запуститься, убедитесь, что:

- сервер CodeScoring доступен раннеру, и у вас есть для него токен API. Версия сервера — 2026.20.0 или выше, а агента, которого он отдаёт, — 2026.27.0 или выше: задачи сканирования привязывают результат к просканированной ссылке, и на более старых версиях сканирование завершается с ошибкой вместо того, чтобы записать результат не в ту версию проекта (см. раздел «Куда записывается результат сканирования» ниже);
- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- интеграция с CodeScoring настроена и активна для проекта, или унаследована от группы выше него;
- проект политик безопасности привязан к проекту или к группе выше него;
- доступен GitLab Runner с Docker executor, и он может загрузить образ задачи сканирования (образ по умолчанию или тот, что администратор задал в интеграции).

Развёртывание самого сервера CodeScoring не входит в эту интеграцию; порядок описан в руководстве по установке от вендора.

Интеграция с CodeScoring

Настройте интеграцию на уровне проекта или группы, в разделе «Настройки» → «Интеграции» → «CodeScoring». Форма задаёт подключение, доверие сертификату и настройки по умолчанию для задач сканирования. Опции сканирования задаются в политике выполнения сканирования.

Подключение

В разделе «Подключение» есть следующие поля:

Поле	Что задаёт
«URL-адрес сервера»	Базовый URL-адрес инстанса CodeScoring. Обязательное
«Токен API»	Токен API для аутентификации и загрузки агента johnny. Обязательное
«Стадия сканирования»	Стадия, для которой ваш сервер CodeScoring хранит политики: `build`, `dev`, `source`, `stage`, `test` или `prod`. По умолчанию — «build». Используется, только если политика не задаёт свою стадию

Доверие сертификату

В разделе «Доверие сертификату» есть следующие поля:

Поле	Что задаёт
«Доверие к сертификату»	Как задача сканирования решает, доверять ли серверу CodeScoring: «Доверять тому, чему уже доверяет задача», «Доверять удостоверяющему центру ниже» или «Не проверять сертификат»
«Удостоверяющий центр»	PEM-сертификат центра, подписавшего сертификат сервера. Показывается и используется, только если для «Доверие к сертификату» выбрано «Доверять удостоверяющему центру ниже»

⚠ Настройка «Доверие к сертификату» действует на скачивание агента johnny в задаче сканирования и на проверку «Проверить настройки».

Задачи сканирования

В разделе «Задачи сканирования» есть следующие поля:

Поле	Что задаёт
«Образ задачи сканирования»	Необязательно. Образ, в котором выполняются задачи сканирования CodeScoring, с хранилищем образов и тегом или дайджестом. Действует только значение, заданное в интеграции на уровне группы; значение в интеграции проекта игнорируется. Если оставить пустым, задачи используют образ по умолчанию. Для образа в закрытом хранилище образов укажите в политике переменные CI/CD с логином и паролем хранилища образов: Учётные данные хранилища образов для сканера
«Формат SBOM»	Необязательно. В какой версии CycloneDX записывается перечень состава: `cyclonedx_v1_4_json`, `cyclonedx_v1_5_json`, `cyclonedx_v1_6_json`, `cyclonedx_v1_6_ext_json` (собственное расширение CodeScoring версии 1.6 с ГОСТ-свойствами) или `cyclonedx_v1_7_json`. Все версии перечисляют одни и те же компоненты. Формат, заданный в политике выполнения сканирования, имеет приоритет. Если не задано ни здесь, ни в политике, задачи сканирования пишут CycloneDX 1.6 JSON

Собственный образ задачи сканирования должен содержать curl, ca-certificates, jq, ruby и git, а также исполняемый файл каждого поискового движка секретов, который включает политика (gitleaks, trufflehog, kingfisher). Задача, в образе которой нет одной из этих команд, завершается с ошибкой и называет её.

Нажмите «Сохранить изменения», чтобы сохранить форму. Нажмите «Проверить настройки», чтобы одним запросом проверить URL-адрес, токен и доверие сертификату на сервере.

Включение сканирования

Добавьте действие `codescoring` в файл `.deckhouse_code/security-policies/policy.yml` проекта политик безопасности:

```
scan_execution_policy:
  - name: CodeScoring everywhere
    enabled: true
    rules:
      - type: pipeline
        branches: ["*"]
    actions:
      - scan: codescoring
        variables:
          FE_SCANS_CODESCORING_SCA: "true"
          FE_SCANS_CODESCORING_SECRETS: "true"
          FE_SCANS_CODESCORING_SECRETS_GITLEAKS: "true"
```

Действие должно включать хотя бы один модуль в `variables`: `FE_SCANS_CODESCORING_SCA` для SCA или `FE_SCANS_CODESCORING_SECRETS` вместе хотя бы с одной из переменных `FE_SCANS_CODESCORING_SECRETS_GITLEAKS`, `FE_SCANS_CODESCORING_SECRETS_TRUFFLEHOG` и `FE_SCANS_CODESCORING_SECRETS_KINGFISHER` для «Секретов». Политика, которая не включает ни одного модуля, не проходит проверку и пропускается без сообщения в пайплайне.

Затем привяжите проект политик: для проекта — в разделе «Безопасность» → «Политики», для группы — в разделе «Настройки» → «Политика безопасности». Опции сканирования можно также редактировать в редакторе политик, который показывает SCA и «Секреты» как два модуля, включаемых независимо друг от друга.

Редактор политик предлагает `codescoring` только тогда, когда для покрываемого проекта (или группы выше него) настроена активная интеграция с CodeScoring; до тех пор редактор показывает «Для использования этого сканирования в политиках безопасности требуется активная интеграция с CodeScoring.» со ссылкой «Настроить интеграцию с CodeScoring».

После применения политики каждый подходящий под неё пайплайн получает задачу `codescoring_sca`, если включён модуль SCA, и по одной задаче `codescoring_secrets_<движок>` на каждый поисковый движок секретов, выбранный в модуле «Секреты». Все эти задачи выполняются на этапе пайплайна, который называет политика, по умолчанию — `fe-security-scanner` (раздел «Этап задач сканирования» на странице Сканирование с помощью Trivy). Модуль, который политика не включила, не даёт ни одной задачи.

SCA

Анализ состава: от чего зависит репозиторий, какие из этих зависимостей известны как уязвимые и какой перечень их описывает.

У модуля SCA есть следующие поля:

Поле	Что задаёт
«Стадия сканирования»	Стадия, для которой применяются политики сервера CodeScoring. Редактор политик подставляет `build` при включении SCA, и значение из политики имеет приоритет над полем «Стадия сканирования» интеграции
«Режим сканирования»	Что сканировать: каталог с исходниками (по умолчанию), отдельный файл, образ контейнера, готовый документ CycloneDX или манифесты только одной технологии
«Цель сканирования»	Необязательно. Каталог, ссылка на образ или путь к SBOM, в зависимости от режима. По умолчанию — весь чекаут
«Исключения»	Необязательно. По одному шаблону в строке. Убирает из результатов тестовые фикстуры и вендорные деревья
«Разрешать зависимости на сервере»	Флажок. Разрешает дерево зависимостей на сервере CodeScoring, а не на раннере, поэтому пакетный менеджер в образе задачи сканирования не нужен. Пока флажок установлен, несоответствия SCA не попадают в Deckhouse Code: отчёт об уязвимостях, виджет Code Quality, вкладка «Тесты» и DefectDojo остаются пустыми, записывается только SBOM. Задача всё равно завершается с ошибкой, если несоответствие блокируют политики сервера CodeScoring
«Формат SBOM»	То же значение, что и в интеграции; заданное здесь значение имеет приоритет над значением интеграции
«Дополнительные аргументы агента»	Необязательно. Передаётся агенту как написано, для опций, которых эта форма не называет, например аргументов конкретного пакетного менеджера. Поля выше всегда имеют приоритет над той же опцией, указанной здесь

Если включить SCA, оставив все поля по умолчанию, сканируется весь чекаут как каталог с исходниками, на стадии `build`, и пишется CycloneDX 1.6 JSON. В основной ветке проекта сканирование также фиксируется как версия проекта по умолчанию на сервере CodeScoring.

Секреты

Учётные данные, попавшие в репозиторий. Выберите один или несколько поисковых движков: gitleaks, trufflehog, kingfisher. Каждый выбранный движок выполняется в отдельной задаче.

Включение любого движка открывает одно общее поле:

Поле	Что задаёт
«Режим сканирования»	«В рабочем дереве (по умолчанию)» ищет то, что есть в ветке сейчас; «В истории репозитория» ищет ещё и то, что один коммит добавил, а следующий убрал. Поиск по истории идёт дольше и может найти секрет, который из файлов уже удалён

Поиск секретов идёт по каталогу из поля «Цель сканирования» модуля SCA или по всему чекауту, если поле пустое.

Дальше у каждого движка открываются собственные элементы:

Поле	Показывается для	Что задаёт
«Конфигурация»	gitleaks (TOML), trufflehog (YAML); не предлагается для kingfisher, чьи правила встроены	Конфигурация движка, записанная прямо в политику
«Игнорировать или исключить»	Для каждого движка	Записи, которые движок должен игнорировать или исключить, записанные прямо в политику
«Игнорировать комментарии gitleaks:allow»	Только для gitleaks	Флажок включён по умолчанию: движок игнорирует комментарий `gitleaks:allow` в строке и всё равно сканирует её. Снимите флажок, чтобы такие комментарии снова пропускали строку, как обычно делает gitleaks
«Дополнительные аргументы агента»	Для каждого движка	Аргументы, которые передаются агенту раньше всего, что задаёт эта форма, поэтому поля выше всегда имеют приоритет над тем же флагом, указанным здесь

Отчёты и куда они попадают

Каждая задача выгружает следующие отчёты:

Задача	Отчёты
`codescoring_sca`	`dependency_scanning`, `codequality`, `cyclonedx` (SBOM) и `junit`
`codescoring_secrets_<движок>` (по одной задаче на каждый выбранный движок)	`secret_detection` и `junit`

Отчёты отображаются в следующих местах:

Где	Что там видно
Вкладка «Тесты» пайплайна	По одному набору JUnit на каждую выполнившуюся задачу: `codescoring_sca` и по одному на каждый выбранный поисковый движок секретов
Виджет Code Quality в запросе на слияние	Несоответствия из `codescoring_sca`
Отчёт об уязвимостях	Несоответствия анализа зависимостей из `codescoring_sca` и несоответствия поиска секретов из каждой задачи `codescoring_secrets_<движок>`, отражённые так же, как их отражает любое другое сканирование `secret_detection`. Секрет, который покрывают и поиск CodeScoring, и обязательное сканирование на секреты, может появиться там дважды
Страница «Список зависимостей»	SBOM, который пишет `codescoring_sca`
Страница «Соответствие лицензий»	Лицензии компонентов из того же SBOM
DefectDojo	Все несоответствия выше, если эта интеграция настроена

Куда записывается результат сканирования

Deckhouse Code определяет, какую ссылку и какой коммит покрывает сканирование, и передаёт их агенту, поэтому результат записывается против того кода, который просканировали: ветки, тега или исходной ветки запроса на слияние, к которому относится пайплайн. Сканируемый проект не может задать это в своих настройках CI/CD.

Каждая задача проверяет привязку до обращения к серверу и останавливается с ненулевым кодом возврата, если ссылки нет, коммит пуст или коммит не совпадает с `HEAD` рабочей копии. Сканирование, которое не может сказать, какой код прочитало, завершается с ошибкой, а не записывает результат в угаданную версию.

Пайплайн ветки по умолчанию проекта дополнительно отмечает свой результат версией по умолчанию на сервере.

Два ограничения сервера, о которых стоит знать при чтении его версий; измерено на 2026.27.2:

- ветка и тег с одинаковым коротким именем попадают в одну версию проекта, потому что версия ключуется этим коротким именем; полная ссылка остаётся в самой записи анализа;
- поиск секретов записывает свой анализ в версию по умолчанию, сохраняя ссылку в записи анализа, тогда как анализ состава для той же ссылки записывается в версию этой ссылки.

Результаты, собранные до того, как сервер обновился до 2026.20.0, остаются в той версии, в которую попали: сервер не переносит анализы между версиями, поэтому версии по ссылкам наполняет повторное сканирование проекта.

Блокировка

`codescoring_sca` завершается с ошибкой, если агент вышел с ненулевым кодом: он нашёл что-то, что блокируют политики сервера CodeScoring, не смог выполниться или не дал результата. Собственного порога критичности здесь нет: что считать достаточно серьёзным, решают политики сервера CodeScoring, и вкладка «Тесты» отмечает как проваленные именно те несоответствия, которые блокируют эти политики.

Каждая задача `codescoring_secrets_<движок>` завершается с ошибкой при любом найденном секрете. Чтобы пропустить проверенное несоответствие, добавьте его в список «Игнорировать или исключить» этого движка. Чтобы только сообщать о секретах, не проваливая задачу, задайте `FE_SECURITY_FAIL_ON: report_only` в `variables` действия.

Любая задача также завершается с ошибкой, если агент johnny не удалось скачать.

Диагностика

Сканирование не появляется в пайплайне

Проверьте, что:

- интеграция с CodeScoring активна для проекта, или унаследована от группы выше него, и в ней заданы URL-адрес сервера и токен;
- проект политик безопасности привязан, и его `policy.yml` содержит действие `codescoring`;
- действие `codescoring` включает хотя бы один модуль в `variables`;
- доступен GitLab Runner с Docker executor, и он может загрузить образ задачи сканирования.

«WARNING: the CodeScoring agent (johnny) could not be downloaded»

Это сообщение появляется в логе задачи `codescoring_sca`: задача не смогла скачать агент, ничего не просканировала и завершается с ошибкой. Задача `codescoring_secrets_<движок>` в той же ситуации пишет «ERROR: the secret search could not run – the johnny agent was not delivered.». Обычно причина одна из следующих:

- Сервер CodeScoring не публикует агент для платформы раннера, например раннер на arm64 обращается к серверу, который публикует агент только под amd64. Попросите администратора задать образ под amd64 в поле «Образ задачи сканирования».
- Неверный URL-адрес сервера или токен API.
- Задача не доверяет сертификату сервера. Настройте «Доверие к сертификату» в интеграции или передайте связку CA через CI/CD-переменную типа «Файл» `CODESCORING_SSL_FILE`.

Задача поиска секретов проваливается при каждом запуске

Это ожидаемое поведение. Причина и способ пропустить конкретное проверенное несоответствие описаны в разделе «Блокировка» выше.

Результаты не появляются на страницах «Список зависимостей» или «Соответствие лицензий»

Проверьте, что `codescoring_sca` выполнилась и создала артефакт `cyclonedx` в формате `gl-sbom.cdx.json`, которого ожидают эти страницы. Порядок диагностики описан в разделе «Диагностика» страницы «Список зависимостей и соответствие лицензий».

Интеграция с Ягой

Яга может служить внешней системой управления задачами проекта. Deckhouse Code превращает коды задач в ссылки на Ягу, добавляет комментарии к задачам и может проверять коммиты и запросы на слияние по этим задачам.

Предварительные требования

Администратор включил флаг функций `fe_jaga_integration` так, как описано на странице Ранний доступ.

Для настройки интеграции в проекте нужна роль «Мейнтейнер» или «Владелец» в этом проекте. Для настройки в группе нужна роль «Владелец»; настройки инстанса изменяет администратор.

Используйте учётную запись Яги с доступом к проектам, задачи которых будет обрабатывать интеграция. Действия в Яге выполняются от имени этой учётной записи. Интеграция не ограничивает коды задач конкретными проектами Яги, поэтому предоставьте учётной записи доступ только к проектам, задачи которых требуется изменять.

В проекте может быть активна только одна внешняя система управления задачами: Deckhouse Code не сохраняет интеграцию с Ягой в проекте, где активна другая система. Активная система, унаследованная от группы или инстанса, тоже мешает сохранению. Перед настройкой Яги откройте эту систему в проекте, выберите «Использовать пользовательские настройки» и отключите её.

Если Яга находится в локальной сети, попросите администратора разрешить обращения к её адресу в разделе «Admin» → «Настройки» → «Сеть» → «Исходящие запросы».

Настройка интеграции

1. Откройте проект и перейдите в раздел «Настройки» → «Интеграции» → «Jaga».
2. В разделе «Включить интеграцию» установите флажок «Активные».
3. В поле «Web URL» укажите базовый адрес Яги, например `https://jaga.example.com`. В полях «Электронная почта» и «Пароль» введите данные учётной записи Яги. Для обращений к API Deckhouse Code использует адрес `<Web URL>/backend`.

4. В разделе «Упоминания» выберите источники комментариев в Яге: «Коммиты», «Запросы на слияние» и «Задачи GitLab». В поле «Подробность комментария» выберите заголовок или полный текст.
5. В разделе «Задачи Jaga» заполните поля «Регулярное выражение задачи Jaga» или «Префикс задачи Jaga», если требуется другой шаблон кодов. По умолчанию код задачи распознаётся, если он начинается с заглавной латинской буквы, за которой следуют одна или несколько заглавных латинских букв, цифр или подчёркиваний, затем дефис и число, например `JAGA-1`.
6. При необходимости настройте разделы «Умные коммиты» и «Закрытие задач» для изменения задач через коммиты.
7. Нажмите «Сохранить изменения». На странице группы или инстанса подтвердите сохранение кнопкой «Сохранить» в окне «Сохранить настройки?». На странице проекта или группы нажмите «Проверить настройки», чтобы проверить адрес и учётные данные.

Интеграцию также можно настроить для группы или инстанса. Проекты наследуют эти настройки и могут задать собственные. При смене «Web URL» сохранённый пароль удаляется: введите его заново перед сохранением. При использовании `http://` учётные данные и токены передаются без шифрования.

По умолчанию в разделе «Упоминания» выбраны все три источника, а в поле «Подробность комментария» — «Заголовок коммита или запроса на слияние». Умные коммиты и переходы Jaga выключены. При включении умных коммитов команда `#comment` и команда статуса включены, `#time` выключена, а в поле «Вид работы» указано `develop`.

Ссылки и упоминания задач

Код задачи вида `JAGA-1` в задаче, запросе на слияние, комментарии или сообщении коммита становится ссылкой на `<Web URL>/browse/JAGA-1`. В боковом меню проекта появляется ссылка «Jaga» на адрес из поля «Web URL».

Для источников, выбранных в разделе «Упоминания», Deckhouse Code добавляет к задаче Яги комментарий со ссылкой на объект в Deckhouse Code. Несуществующая задача пропускается. Повторное упоминание того же объекта не создаёт новый комментарий. Комментарии добавляются от имени учётной записи Яги из настроек интеграции.

Комментарии об упоминании, умные коммиты и закрытие задачи коммитом, отправленным в ветку по умолчанию, находят код, только если в тексте есть код вида `JAGA-1` или другая ссылка на объект, например `#1` или URL. Код вида `JAGA-1` — это заглавная латинская буква, за которой могут следовать заглавные латинские буквы, цифры или подчёркивания, затем дефис и число. Код другого вида, заданный собственным регулярным выражением, в тексте без такой ссылки молча игнорируется. Ссылки, созданные из кодов задач, правила отправки изменений, проверка при слиянии и закрытие через запрос на слияние используют заданный шаблон напрямую.

Умные коммиты

В разделе «Умные коммиты» установите флажок «Выполнять команды из коммитов, у автора которых есть учётная запись Jaga», затем включите нужные команды. Электронная почта автора, записанная в коммите, должна соответствовать учётной записи Яги. Действия выполняет учётная запись из настроек интеграции.

Поместите код задачи и команды в одну строку сообщения коммита:

```
JAGA-1 #comment Исправлена форма входа #на-проверке #time 1h 30m
```

- `#comment` добавляет следующий за командой текст в комментарий к задаче.
- `#на-проверке` переводит задачу в доступный статус «На проверке». Подставьте название статуса из своего процесса в Яге, заменив пробелы дефисами; регистр не учитывается. Командой статуса считается только первое `#слово` в коммите, кроме `#comment` и `#time`; следующие игнорируются.
- `#time` списывает рабочее время, если команда включена и указан «Вид работы». Единицы времени: `w`, `d`, `h` и `m`; день равен восьми часам, неделя — пяти дням. Запись больше 40 часов, например `6d` или `1w 1h`, игнорируется, время по ней не списывается. Яга записывает время на учётную запись интеграции.

Команды выполняются при отправке коммита, в том числе в ветки, отличные от ветки по умолчанию. Обрабатываются только последние 100 коммитов одной отправки; это относится и к комментариям об упоминании, и к закрытию задач коммитом. Если у автора коммита нет учётной записи Яги, команды пропускаются. После выполнения команд комментарий добавляется и при снятом флажке «Коммиты» в разделе «Упоминания».

Если задачу нельзя перевести в названный статус — Яга отклоняет смену, из текущего статуса задачи нет перехода в него или Яга не отвечает, — записи `#time` и комментарий этого коммита тоже пропускаются. Они выполняются, когда Deckhouse Code обработает коммит в следующий раз, например при отправке коммита в новую ветку или при попадании в ветку по умолчанию. Повторная отправка того же коммита в ту же ветку его не обрабатывает. Если команда статуса не совпадает ни с одним статусом проекта Яги, например `#wip`, она остаётся обычным текстом, а `#comment` и `#time` выполняются.

> ⚠ Если Яга не смогла добавить комментарий после списания времени командой `#time`, повторная обработка коммита может списать время ещё раз. Проверьте записи времени в задаче Яги перед повторной обработкой коммита, например перед отправкой коммита в другую ветку или слиянием.

Закрытие задач Яги

В разделе «Закрытие задач» установите флажок «Включить переходы Jaga», чтобы менять статус задачи, когда закрывающий коммит или запрос на слияние попадает в ветку по умолчанию.

Добавьте запись вида `Closes JAGA-1` или `Fixes JAGA-1` в сообщение коммита или, для запроса на слияние, в его заголовок, описание или сообщения его коммитов. Если задан префикс задачи, включите его в запись.

Выберите «Перевести в «Готово»» для перехода в первый доступный статус категории «Готово». Выберите «Перевести по целевым статусам» и перечислите названия статусов в порядке процесса через запятую или точку с запятой (подсказка под полем называет только запятую) для прохождения нескольких доступных переходов. Если Яга отклоняет переход, задача остаётся в последнем достигнутом статусе. После перехода в категорию «Готово» Deckhouse Code добавляет комментарий со ссылкой на закрывающий коммит.

Правила отправки изменений

Настройте проверки Яги в разделе «Настройки» → «Репозиторий» → «Push-правила» на уровне проекта, группы или инстанса, как описано на странице Правила отправки изменений. Интеграция с Ягой должна быть активна на этом уровне либо унаследована. По умолчанию правила выключены:

- «Отклонять коммиты с несуществующей задачей Jaga» требует, чтобы каждый новый коммит называл хотя бы одну существующую задачу.
- «Отклонять коммиты, если email автора не принадлежит исполнителю задачи Jaga» требует, чтобы каждая существующая задача, названная в коммите, была назначена на электронную почту автора коммита напрямую или через группу Яги. Коммит отклоняется, если у названной задачи нет исполнителей или ни у одного пользователя Яги нет такой почты.
- «Статусы задач Jaga, в которых разрешён push» требует, чтобы каждая существующая задача, названная в коммите, находилась в одном из перечисленных статусов. Поле принимает названия статусов через запятую или точку с запятой (подсказка под полем называет только запятую); регистр не учитывается. Как и у каждого правила отправки изменений, у него есть список со значениями «Включен» и «Отключен». Правило действует, только если выбрано «Включен» и в поле указан хотя бы один статус; пустое поле оставляет правило выключенным.

При любом включённом правиле Яги каждый новый коммит в отправляемых ветках и тегах должен называть хотя бы одну задачу, которая существует в Яге. Код, для которого в Яге нет задачи, не приводит к отклонению, если коммит называет и существующую задачу. Коммиты из веб-редактора и коммиты слияния тоже проверяются. Первый коммит, созданный через веб-интерфейс для заполнения пустого репозитория, например начальный README, не проверяется.

Отправка изменений также отклоняется, если:

- в ней названо более 100 различных кодов задач;
- Яга недоступна или отвечает ошибкой — пока Яга не начнёт отвечать без ошибок;
- Яга отклоняет учётные данные интеграции — до исправления учётных данных или правил;
- правило, унаследованное от группы или инстанса, осталось включённым после отключения интеграции в проекте.

Проверка исполнителя использует электронную почту автора, записанную в коммите. Создатель коммита может указать её самостоятельно; чтобы установить, кто подписал коммит, включите также правило «Коммит должен быть подписан GPG».

Правила отправки изменений, включая правила Яги, не действуют в проекте с pull-зеркалированием: не проверяются ни обновления зеркала, ни прямые отправки изменений в такой проект.

Проверка при слиянии

В проекте откройте «Настройки» → «Запросы на слияние» → «Проверки слияния» и установите флажок «В заголовке или описании должна быть указана задача Jaga». Интеграция должна быть активна в проекте или унаследована. Проверка ищет код задачи в заголовке или описании запроса на слияние без обращения к Яге.

Флажок недоступен, пока интеграция с Ягой не настроена. Если интеграцию позже отключить, флажок остаётся установленным, но проверка больше не блокирует слияние.

Если правило отправки изменений требует код задачи в каждом коммите, укажите его в заголовке запроса на слияние. Сообщение коммита слияния по умолчанию содержит заголовок и проходит проверку при отправке изменений. Если в проекте задан шаблон сообщения коммита слияния, включите в него `%{title}`, а при редактировании сообщения перед слиянием сохраните код задачи. Если включено «Объединять коммиты при слиянии», проверяется и объединённый коммит: его сообщение по умолчанию берётся из шаблона сообщения squash-коммита, а без шаблона — из заголовка запроса на слияние. Включите `%{title}` и в этот шаблон.

Интеграция с PT AI

PT Application Inspector (PT AI) — анализатор защищённости приложений от Positive Technologies: статический анализ кода, поиск уязвимых компонентов, конфигурации и секретов (белый ящик) и сканирование работающего сайта (чёрный ящик). Интеграция с PT AI подключает проект к серверу PT AI Enterprise Server, а политика выполнения сканирования добавляет в пайплайны проекта задачу `ptai_scan` и задаёт, что она сканирует. Задача загружает исходный код коммита на сервер клиентом aictl, ждёт окончания сканирования, читает вердикт политики безопасности PT AI и публикует отчёты.

Предварительные требования

Чтобы сканирование могло запуститься, убедитесь, что:

- администратор включил флаг функций `fe_security_scan_policies` так, как описано на странице Ранний доступ;
- сервер PT AI Enterprise Server версии от 5.0 до 7.0 доступен раннеру, и у вас есть для него токен доступа (см. Токен доступа PT AI);
- интеграция с PT AI настроена и активна для проекта или унаследована от группы выше него;
- проект политик безопасности привязан к проекту или к группе выше него;

- доступен GitLab Runner с Docker executor, и он может загрузить образ задачи сканирования.

Токен доступа PT AI

1. Войдите в веб-интерфейс PT AI Enterprise Server под учётной записью администратора.
2. Нажмите значок шестерёнки в левом нижнем углу, чтобы открыть раздел администрирования.
3. Откройте **Настройка системы** → **Токены доступа**.
4. Нажмите **Создать токен**.
5. Задайте название токена, например `gitlab-ci-ptai`, выберите разрешение **Для легкого агента и плагинов CI/CD** и срок действия.
6. Подтвердите создание и скопируйте токен.

После закрытия окна значение токена в PT AI больше не показывается. Потерянный токен отзовите и создайте новый.

Интеграция с PT AI

Настройте интеграцию на уровне проекта или группы в разделе «Настройки» → «Интеграции» → «Positive Technologies Application Inspector». Форма задаёт подключение к серверу, доверие его сертификату и образ задачи сканирования. Что сканируется, чёрный ящик и вход на сайт задаются в политике выполнения сканирования.

Подключение

Поле	Что задаёт
«URL сервера»	Базовый URL сервера PT AI, например `https://ptai.example.com`. Обязательное
«API-токен»	Токен доступа PT AI, с которым задача входит на сервер. Обязательное

Настройка TLS

Поле	Что задаёт
«Доверие к сертификату»	Как задача сканирования решает, доверять ли серверу PT AI: «Доверять сертификатам образа джобы», «Доверять удостоверяющему центру ниже» или «Не проверять сертификат»
«Удостоверяющий центр»	PEM-сертификат центра, подписавшего сертификат сервера. Используется, только если выбрано «Доверять удостоверяющему центру ниже»

Задача передаёт удостоверяющий центр клиенту aictl отдельным параметром. Хранилище сертификатов образа для остальных команд задачи не меняется.

Образ задачи сканирования

Поле	Что задаёт
«Образ джобы сканирования»	Образ, в котором выполняется задача `ptai_scan`: полный адрес с тегом или дайджестом. Обязательное. Новая интеграция заполняется образом из поставки продукта, `registry.flant.com/deckhouse/code/gitlab-custom/ptai:1.2.2`
«Переменная логина реестра»	Необязательно. Имя переменной CI/CD проекта или его группы с логином для реестра образа, например `$SCANNER_REGISTRY_USER`. Обычная переменная, не файловая
«Переменная пароля реестра»	Необязательно. Имя переменной CI/CD с паролем или токеном для этого реестра. Пароль не может содержать символы `"` и `\`

Собственный образ должен содержать aictl, ruby и mktemp, а если политика читает конфигурационный файл из другого проекта, то и curl. Задача, в образе которой нет одной из этих команд, завершается с ошибкой и называет её.

Поля раздела «Джобы сканирования» читаются только из интеграции группы, которой принадлежит политика, или ближайшей группы выше неё. Значения в интеграции проверяемого

проекта ни на что не влияют. Адрес сервера, токен и доверие сертификату задача берёт из интеграции проверяемого проекта: собственной или унаследованной.

Нажмите «Проверить настройки», чтобы войти на сервер с токеном из формы и запросить версию PT AI. При ошибке форма показывает код ответа сервера или ошибку соединения. Нажмите «Сохранить изменения», чтобы сохранить форму.

Включение сканирования

Добавьте действие `ptai` в файл `.deckhouse_code/security-policies/policy.yml` проекта политик безопасности:

```
scan_execution_policy:
  - name: PT AI everywhere
    enabled: true
    rules:
      - type: pipeline
        branches: ["*"]
    actions:
      - scan: ptai
        variables:
          FE_SCANS_PTAI_MODULES: Configuration,Components,PatternMatching,StaticCodeAn
alysis
          FE_SCANS_PTAI_LANGUAGES: Java,Python
```

Затем привяжите проект политик: для проекта — в разделе «Безопасность» → «Политики», для группы — в разделе «Настройки» → «Политика безопасности».

Сканирование можно также включить в редакторе политик флажком у сканирования PT AI. Редактор предлагает его, только когда для проекта или группы выше него настроена активная интеграция с PT AI; до тех пор редактор показывает «Для использования этого сканирования в политиках безопасности требуется активная интеграция с PT AI.» со ссылкой на настройку интеграции. Настройки сканирования в редакторе разбиты на блоки «Сканирование» и «Чёрный ящик и автопроверка»; «Вход на сайт» и «Прокси-сервер» находятся внутри блока «Чёрный ящик и автопроверка».

Политика не может задавать параметры интеграции. Действие `ptai`, в котором названы `FE_SCANS_PTAI_SERVER_URL`, `FE_SCANS_PTAI_TOKEN`, `FE_SCANS_PTAI_TLS_MODE`, `FE_SCANS_PTAI_CA_CERT`, `FE_SCANS_PTAI_IMAGE`, `FE_SCANS_REGISTRY_LOGIN_VARIABLE` или `FE_SCANS_REGISTRY_PASSWORD_VARIABLE`, не сохраняется. Порог `FE_SECURITY_FAIL_ON` для `ptai` тоже не принимается: задачу роняет политика безопасности на сервере PT AI (см. Когда задача завершается с ошибкой).

Опции политики проверяются при сохранении и ещё раз при формировании задачи, поэтому политика, записанная в репозиторий без редактора, с недопустимым значением роняет задачу до обращения к серверу:

```
ERROR: this scan cannot run the configuration the policy asks for: <причины>
```

После применения политики каждый подходящий под неё пайплайн получает задачу `ptai_scan` на этапе пайплайна, который называет политика, по умолчанию — `fe-security-scanner`.

Сканирование

Блок «Сканирование» обязателен: в каждом сканировании работает хотя бы один модуль белого ящика, а чёрный ящик добавляется к нему.

Поле	Переменная	Что задаёт
«Модули анализа»	`FE_SCANS_PTAI_MODULES`	Модули белого ящика, через запятую. По умолчанию — `Configuration`, `Components`, `PatternMatching`, `StaticCodeAnalysis`: с ними сервер PT AI создаёт новый проект
«Языки приложения»	`FE_SCANS_PTAI_LANGUAGES`	Языки проекта, через запятую. Обязательно, как только заданы модули, адрес сайта или «Не загружать зависимости»
«Не сканировать файлы и папки, указанные в .gitignore»	`FE_SCANS_PTAI_SKIP_GITIGNORE`	`true` исключает из сканирования результаты сборки и каталоги со сторонним кодом, если репозиторий их игнорирует
«Не загружать зависимости»	`FE_SCANS_PTAI_SKIP_DEPENDENCY_DOWNLOAD`	`true` запрещает ядрам анализа загружать пакеты проекта для всех языков. Включите, если у сервера PT AI нет доступа к реестрам пакетов: иначе загрузка роняет сканирование
«Конфигурационный файл»	`FE_SCANS_PTAI_SETTINGS_FILE`	Путь к собственному файлу настроек проекта (см. Конфигурационный файл)
«Проект с конфигурационным файлом»	`FE_SCANS_PTAI_SETTINGS_PROJECT`	Репозиторий с этим файлом, в виде `group/project`

Модули в редакторе называются так же, как в документации PT AI:

Значение в политике	Название в редакторе
`StaticCodeAnalysis`	«Статический анализ кода»
`PatternMatching`	«Поиск по шаблонам»
`Components`	«Поиск уязвимых компонентов»
`SoftwareCompositionAnalysis`	«Поиск уязвимых компонентов и их использования»
`Configuration`	«Поиск уязвимостей конфигурации»
`MaliciousCodeDetection`	«Поиск вредоносного кода»
`SecretDetection`	«Поиск секретов в коде»

`BlackBox` в списке нет: модуль добавляется сам, когда задан адрес сайта. Если в политике нет ни одного модуля белого ящика, задача берёт модули по умолчанию. `SecretDetection` сервер принимает только вместе со `StaticCodeAnalysis` или `PatternMatching`; без них сканирование завершается ошибкой `INVALID_MODULES_FOR_SECRET_DETECTION`.

Языки: `Java`, `CSharp` (CSharp (Windows, Linux)), `CSharpWindows` (CSharp (Windows)), `Php`, `JavaScript`, `Python`, `ObjectiveC`, `Swift`, `CAndCPlusPlus`, `Go`, `Kotlin`, `Sql`, `Ruby`, `Solidity`, `Scala`, `OneC`.

Задача загружает на сервер рабочее дерево коммита без каталога `.git`, каталогов `reports` и `security-reports` и файла `.aiproj.json` в корне репозитория. Подмодули и объекты LFS попадают в загрузку так, как их выгрузил раннер; это настраивается переменными CI/CD `GIT_SUBMODULE_STRATEGY` и `GIT_LFS_SKIP_SMUDGE` проекта.

Чёрный ящик

Блок «Чёрный ящик и автопроверка» сканирует работающее приложение в той же задаче, что и исходный код. Сайт открывает сервер PT AI, поэтому адрес должен быть доступен ему; внутренние адреса допускаются.

Поле	Переменная	Что задаёт
«Адрес сайта»	`FE_SCANS_PTAI_BLACKBOX_SITE`	URL сайта, `http` или `https`. Заполнение поля включает чёрный ящик. Любая другая опция чёрного ящика без адреса сайта не сохраняется
«Режим поиска»	`FE_SCANS_PTAI_BLACKBOX_LEVEL`	Насколько подробно исследуется сайт: `Fast`, `Normal` или `Full`. По умолчанию — `Normal`
«Область сканирования»	`FE_SCANS_PTAI_BLACKBOX_SCOPE`	Насколько далеко от адреса уходит сканирование: `Domain` («Домен»), `Folder` («Папка») или `Path` («Путь»). По умолчанию — `Path`
«Проверка безопасности HTTPS-соединения»	`FE_SCANS_PTAI_BLACKBOX_SSL_CHECK`	Проверяет ли сервер PT AI сертификат сайта: `true` (по умолчанию) или `false`. Для сайта с сертификатом частного удостоверяющего центра добавьте этот центр в хранилище сертификатов сервера PT AI
«Автопроверка уязвимостей по завершении сканирования»	`FE_SCANS_PTAI_BLACKBOX_AUTOCHECK`	`true` запускает эксплойты для находок статического анализа на работающем сайте, чтобы подтвердить или опровергнуть каждую. Редактор включает её для нового блока
«Дополнительные HTTP-заголовки»	`FE_SCANS_PTAI_BLACKBOX_HEADERS`	По одному в строке, в виде `Name: value`
«Белый список адресов»	`FE_SCANS_PTAI_BLACKBOX_ALLOWLIST`	По одному в строке. Адрес без суффикса — шаблон; `адрес|ExactMatch` — точное

Поле	Переменная	Что задаёт
		совпадение, `адрес|RegExp` — регулярное выражение
«Чёрный список адресов»	`FE_SCANS_PTAI_BLACKBOX_DENYLIST`	В том же виде. Сюда добавляют ссылку выхода и всё, по чему сканирование переходить не должно

Политику читает каждый проект, который она охватывает, поэтому секреты в ней не хранятся. Поля с паролем, cookie и паролем прокси принимают ссылку на переменную CI/CD проверяемого проекта или его группы: `$NAME` или `${NAME}`. Раннер подставляет значение в задаче. Политика с литеральным значением не сохраняется; имена переменных, начинающиеся с `FE_`, `CI_` и `GITLAB_`, не принимаются. Заголовки `Authorization`, `Proxy-Authorization`, `Cookie`, `X-Api-Key` и `X-Auth-Token` задаются так же: значением заголовка должна быть вся ссылка.

```
FE_SCANS_PTAI_BLACKBOX_HEADERS: |
  X-Trace: on
  Authorization: $PTAI_SITE_AUTHORIZATION
```

`Bearer $TOKEN` не принимается: храните `Bearer` вместе с токеном в переменной `PTAI_SITE_AUTHORIZATION`.

Если переменная, на которую ссылается политика, не существует или пуста, задача завершается с ошибкой, например:

```
ERROR: the CI/CD variable the security policy names for the sign-in password holds no value.
```

Вход на сайт

Без входа сканирование видит только страницы, открытые анонимному посетителю. Редактор показывает только поля выбранного типа аутентификации.

Поле	Переменная	Для типов	Что задаёт
«Аутентификация»	`FE_SCANS_PTAI_BLACKBOX_AUTH_TYPE`	все	`none` («Без аутентификации», по умолчанию), `form` («Form-аутентификация»), `http` («HTTP-аутентификация»), `cookie` («Cookie-аутентификация»)
«Логин»	`FE_SCANS_PTAI_BLACKBOX_AUTH_LOGIN`	`form`, `http`	Логин на сайте. Обязательное
«Переменная пароля»	`FE_SCANS_PTAI_BLACKBOX_AUTH_PASSWORD`	`form`, `http`	Ссылка на маскируемую переменную CI/CD с паролем. Обязательное
«Переменная cookie»	`FE_SCANS_PTAI_BLACKBOX_AUTH_COOKIE`	`cookie`	Ссылка на маскируемую переменную CI/CD с сессионным cookie. Обязательное
«Адрес проверки»	`FE_SCANS_PTAI_BLACKBOX_AUTH_VALIDATION_ADDRESS`	`http`, `cookie`	Страница, доступная только вошедшему пользователю. Обязательное
«Искомая строка»	`FE_SCANS_PTAI_BLACKBOX_AUTH_VALIDATION_TEMPLATE`	`form`, `cookie`	Текст, который есть на странице при выполненном входе и которого нет без него. Обязательное для `cookie`
«Адрес формы»	`FE_SCANS_PTAI_BLACKBOX_AUTH_FORM_ADDRESS`	`form`	URL страницы с формой входа
		`form`	`Auto` («Автоматически», по

Поле	Переменная	Для типов	Что задаёт
«Обнаружение формы»	`FE_SCANS_PTAI_BLACKBOX_AUTH_FORM_DETECTION`		умолчанию) или `Manual` («По заданным параметрам») для формы, которую сканирование не находит само
«XPath формы», «Идентификатор для логина», «Идентификатор для пароля»	`FE_SCANS_PTAI_BLACKBOX_AUTH_FORM_XPATH`, `FE_SCANS_PTAI_BLACKBOX_AUTH_LOGIN_KEY`, `FE_SCANS_PTAI_BLACKBOX_AUTH_PASSWORD_KEY`	`form`	XPath формы и имена её полей логина и пароля, при `Manual`

Пример действия со входом HTTP-аутентификацией:

```
actions:
  - scan: ptai
    variables:
      FE_SCANS_PTAI_LANGUAGES: Java
      FE_SCANS_PTAI_BLACKBOX_SITE: https://staging.example.com
      FE_SCANS_PTAI_BLACKBOX_AUTH_TYPE: http
      FE_SCANS_PTAI_BLACKBOX_AUTH_LOGIN: scanner
      FE_SCANS_PTAI_BLACKBOX_AUTH_PASSWORD: $PTAI_SITE_PASSWORD
      FE_SCANS_PTAI_BLACKBOX_AUTH_VALIDATION_ADDRESS: https://staging.example.com/
profile
```

Чтобы пример работал:

1. Откройте проект или группу и перейдите в «Настройки» → «CI/CD» → «Переменные».
2. Создайте переменную `PTAI_SITE_PASSWORD` со значением пароля сайта и сделайте её маскируемой.

Если вход не удался, сканирование проходит только по открытым страницам, а задача печатает ошибки сканирования, которые вернул сервер, в том числе `AuthFailed`.

Прокси-сервер

Поле	Переменная	Что задаёт
«Адрес»	`FE_SCANS_PTAI_BLACKBOX_PROXY_HOST`	Имя хоста или IP-адрес прокси, не URL. Заполнение поля включает прокси
«Порт»	`FE_SCANS_PTAI_BLACKBOX_PROXY_PORT`	Число от 1 до 65535
«Тип»	`FE_SCANS_PTAI_BLACKBOX_PROXY_TYPE`	`Http` (по умолчанию), `Socks4` или `Socks5`
«Логин»	`FE_SCANS_PTAI_BLACKBOX_PROXY_LOGIN`	Логин на прокси. С ним обязательна переменная пароля
«Переменная пароля»	`FE_SCANS_PTAI_BLACKBOX_PROXY_PASSWORD`	Ссылка на маскируемую переменную CI/CD с паролем прокси

Конфигурационный файл

Для настройки, которую PT AI принимает, а политика не предлагает, задайте собственный файл настроек проекта PT AI (формат `.aiproj`). Задача загружает его на сервер как есть вместо файла, который она собирает из политики. Пока поле «Конфигурационный файл» заполнено, редактор показывает остальные настройки блоков «Сканирование» и «Чёрный ящик и автопроверка» неактивными: задача их не читает, и при сохранении они не проверяются.

```
actions:
  - scan: ptai
    variables:
      FE_SCANS_PTAI_SETTINGS_FILE: .aiproj.json
      FE_SCANS_PTAI_SETTINGS_PROJECT: security/ptai-settings
```

«Конфигурационный файл» — путь от корня репозитория: относительный, без сегмента `..`, без `/` в конце, только символы ASCII. Имя файла может быть любым.

«Проект с конфигурационным файлом» определяет, откуда берётся файл:

- пустое поле — из проверяемого репозитория, на собираемом коммите. Такой файл проект может изменить сам, в том числе в ветке запроса на слияние;
- `group/project` — с ветки по умолчанию этого проекта, токеном задачи. Проверяемые проекты изменить такой файл не могут.

Чтобы задача читала файл из другого проекта:

1. Откройте проект с файлом.
2. Перейдите в «Настройки» → «CI/CD» → «Права доступа токена заданий CI/CD».
3. Добавьте проверяемые проекты или их группу.

Задача завершается с ошибкой до сканирования, если файла нет по указанному пути, файл пустой, проект не пускает задачи проверяемого проекта (HTTP 401 или 403) или файла нет на ветке по умолчанию (HTTP 404). Пароли и cookie, записанные в файл, видит каждый, у кого есть доступ к репозиторию.

Файл `.aiproj.json` из корня проверяемого репозитория задача на сервер не загружает. Чтобы использовать его, назовите его в поле «Конфигурационный файл».

Проект PT AI

Задача сама создаёт на сервере проект и ветку, если их нет. Проект называется полным путём проекта Deckhouse Code, в котором `/` заменён на `:`: `group/subgroup/app` пишет в проект `group:subgroup:app`. Имя длиннее 255 символов обрезается до 240, и в конец добавляется `-p<PROJECT_ID>`, где `<PROJECT_ID>` — ID проекта. Ветка называется именем ветки или тега пайплайна. Точное имя задача пишет в журнал:

```
PT AI: scanning into project group:subgroup:app.
```

Имена проектов в PT AI не зависят от регистра. Переименование или перенос проекта в Deckhouse Code меняет путь: следующее сканирование создаёт новый проект PT AI, а прежний остаётся на сервере со своей историей. Проект можно создать заранее в веб-интерфейсе PT AI, чтобы до первого сканирования настроить для него политику безопасности; назовите его по тому же правилу.

Каждое сканирование заменяет настройки проекта на сервере настройками из политики или конфигурационного файла, наложенными на настройки по умолчанию сервера: модули, языки и настройки языков, флажки блока «Сканирование» и чёрный ящик целиком. Изменение этих настроек в веб-интерфейсе PT AI действует до следующего сканирования. Задача не меняет политику безопасности проекта, флажок «Применять политику безопасности», приоритет проекта и предпочитаемые агенты: они задаются в веб-интерфейсе PT AI.

Очередь сканирований

Сканирует агент PT AI на стороне сервера; задача ждёт результата и всё это время занимает раннер. Время задачи складывается из ожидания в очереди сервера и самого сканирования, поэтому задайте проекту тайм-аут задач с запасом.

Задачи `ptai_scan` одной ветки одного проекта выполняются по очереди: следующая ждёт в статусе `waiting_for_resource`, не занимая раннер и не расходуя тайм-аут. Пайплайн ветки и пайплайн запроса на слияние из неё ждут друг друга. Если ветку уже сканирует кто-то другой,

например из веб-интерфейса PT AI, задача повторяет попытку раз в 30 секунд, пока ветка не освободится или не истечёт тайм-аут задачи.

Отменённая задача, в том числе отменённая новым пайплайном, останавливает своё сканирование на сервере. Чтобы при нескольких отправках подряд сканировался только последний коммит, включите в проекте «Настройки» → «CI/CD» → «Пайплайны» → «Автоматическая отмена избыточных пайплайнов». Задача, остановленная по тайм-ауту, сканирование не останавливает: уберите его из очереди в веб-интерфейсе PT AI на вкладке **Агенты сканирования**.

Когда задача завершается с ошибкой

Порога по уровню находок у сканирования PT AI нет: все находки попадают в отчёты, а задачу роняет политика безопасности проекта на сервере PT AI. После завершённого сканирования задача читает её вердикт:

- `Rejected` — политика нарушена, задача завершается с ошибкой «the PT AI security policy rejected the scan»;
- `Confirmed` — политика соблюдена;
- `None` — проверка политики у проекта выключена.

Сервер записывает вердикт через несколько секунд после сканирования, поэтому при включённой проверке задача ждёт его до 120 секунд. Вердикт, который не удалось получить или прочитать, роняет задачу с сообщением «the PT AI security policy verdict could not be read». Проверку политики включают и правила задают в веб-интерфейсе PT AI: «Настройка» → «Политика безопасности» проекта.

Задача также завершается с ошибкой, если:

- интеграции нет, она неактивна или в ней нет адреса или токена: «ERROR: the PT AI integration is missing or inactive: the job got no server address or token.»;
- первое обращение к серверу не удалось: задача печатает сообщение aictl и «Check the server address, the token and the certificate trust in the PT AI integration.»;
- сервер отклонил настройки сканирования: «ERROR: the PT AI server refused the settings aictl does not send: …» с ответом сервера;
- сканирование завершилось с ошибкой или отчёт не получен: «Failing job: the PT AI scan did not finish or produced no readable report.»; причину, которую назвал сервер, задача печатает после строки «PT AI: the server reports for this scan:».

Отчёты

Задача публикует артефакты:

- `security-reports/gitlab/ptai.json` — отчёт безопасности SAST, отчёт PT AI в формате GitLab без изменений; его находки попадают в [отчёт об уязвимостях];
- `security-reports/junit/ptai.xml` — отчёт о тестах для вкладки «Тесты» пайплайна;

- `reports/ptai.json` и `reports/ptai.sarif` — собственный JSON-отчёт PT AI и SARIF, как их отдал сервер.

На вкладке «Тесты» каждая находка — отдельный тест. Набор тестов — модуль PT AI, имя теста — тип находки и место, например `Cross-Site Scripting [js/server.js:28]`; у находки чёрного ящика вместо места адрес, у уязвимого компонента — компонент, версия, CVE и манифест. Уровень находки — свойство `severity` теста: Critical, High, Medium, Low или Info. Подробности теста содержат CWE и OWASP, код и точку входа, CVE с оценкой CVSS, текущее и рекомендуемое значение конфигурации, вызовы вредоносного кода, адрес и эксплойт чёрного ящика, признак новой находки и ссылку на сканирование в веб-интерфейсе PT AI. Если политика безопасности PT AI отклонила сканирование, первым идёт отдельный тест с её вердиктом. Модуль, код, CVE и адрес чёрного ящика есть только в отчёте о тестах.

Отчёты не маскируются: эксплойт находки чёрного ящика в `reports/ptai.json` и на вкладке «Тесты» содержит строку запроса, заголовок `Host` и тело запроса.

Диагностика

Сканирование не появляется в пайплайне

Проверьте, что:

- интеграция с PT AI активна для проекта или унаследована от группы выше него;
- проект политик привязан к проекту или группе выше него, и правило политики подходит под пайплайн;
- в действии `ptai` политики нет параметров интеграции и порога `FE_SECURITY_FAIL_ON`.

Проверка настроек или первое обращение к серверу не проходит

Проверьте URL сервера, токен (без пробелов и переносов строк, не отозван, срок не истёк, разрешение **Для легкого агента и плагинов CI/CD**) и «Доверие к сертификату». «Проверить настройки» показывает код ответа сервера, например HTTP 401 для неверного токена.

Проекта нет в PT AI

Ищите полный путь проекта с `:` вместо `/`. Точное имя задача пишет в журнал строкой «PT AI: scanning into project …».

«cannot downgrade 1.11 -> 1.10»

Конфигурационный файл из поля «Конфигурационный файл» содержит `Version` 1.11, а сервер — PT AI 6.0. Укажите в файле версию 1.9 или 1.10.

Отчёт не получен после повторного сканирования

Сканирование на сервере в состоянии `Done`, а задача завершается с ошибкой «the PT AI scan did not finish or produced no readable report» и печатает код `REPORT_GENERATE_FAILED`.

Запустите полное сканирование: измените настройки сканирования в политике, и следующий запуск задачи просканирует ветку полностью.

Расширенный поиск

Поиск в Deckhouse Code помогает быстро находить нужную информацию по проектам, группам или всему инстансу. Результаты сортируются по релевантности и позволяют сразу перейти к исходному объекту.

Расширенный поиск на базе OpenSearch позволяет:

- находить фрагменты кода во всех доступных проектах;
- отслеживать использование устаревших функций и библиотек;
- искать комментарии в задачах и запросах на слияние;
- находить коммиты по сообщению или SHA;
- искать содержимое wiki-страниц.

Использование расширенного поиска

- Администратор должен включить расширенный поиск (OpenSearch).

Чтобы выполнить поиск:

1. В верхней панели выберите «Поиск».
2. Введите поисковый запрос.
3. Нажмите «Enter».

Расширенный поиск также доступен в контексте проекта или группы.

При включённом OpenSearch Deckhouse Code использует его как бэкенд для расширенных областей поиска (задачи, запросы на слияние, код и другие). Параметры REST API, фильтры и формат ответа описаны в разделе «API поиска».

Области поиска

Области описывают тип данных, по которым выполняется поиск.

Базовый поиск

Следующие области доступны в базовом поиске (без OpenSearch):

Область	Глобальный	Группа	Проект
Код	✗	✗	✓
Комментарии	✗	✗	✓
Коммиты	✗	✗	✓
Задачи	✓	✓	✓
Запросы на слияние	✓	✓	✓
Этапы (milestones)	✓	✓	✓
Проекты	✓	✓	✗
Пользователи	✓	✓	✓
Wiki	✗	✗	✓

Расширенный поиск

При включённом OpenSearch доступны следующие области:

Область	Глобальный	Группа	Проект
Код	✓	✓	✓
Комментарии	✓	✓	✓
Коммиты	✓	✓	✓
Задачи	✓	✓	✓
Запросы на слияние	✓	✓	✓
Этапы (milestones)	✓	✓	✓
Проекты	✓	✓	✗
Пользователи	✓	✓	✓
Wiki	✓	✓	✓

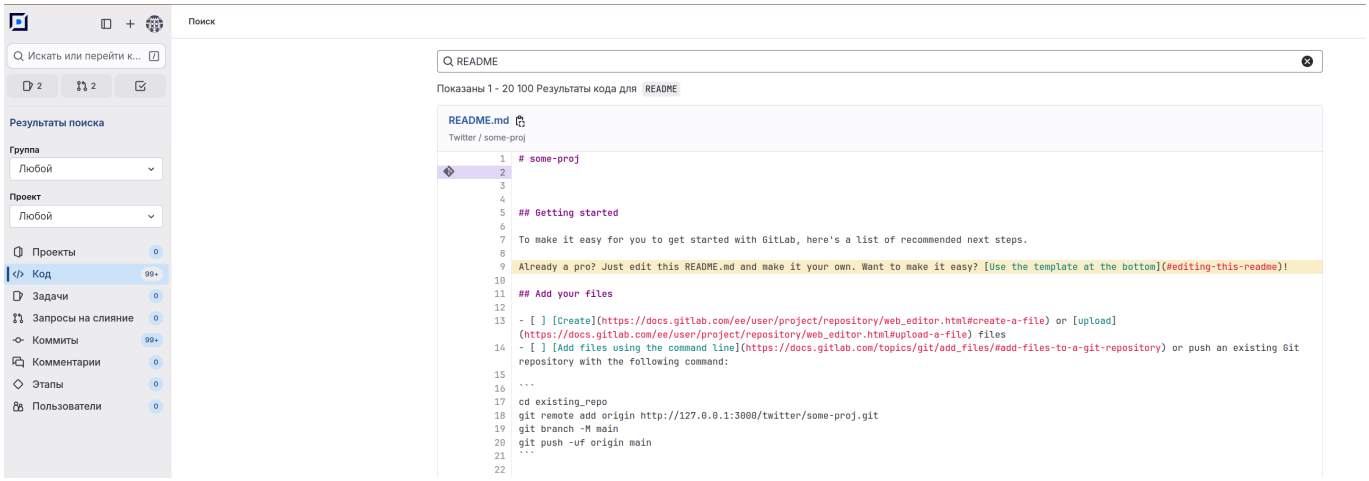
Поиск по коду, коммитам, wiki и комментариям при включённом OpenSearch выполняется через OpenSearch и учитывает **матрицу доступа**. Пользователи видят только те объекты, к которым у них есть права на чтение. Поиск по задачам, запросам на слияние и другим сущностям выполняется через базу данных.

Таблицы выше описывают области поиска в веб-интерфейсе. Набор областей в REST API отличается: поиск по коду, коммитам, комментариям и wiki доступен там только на уровне проекта. Подробнее — в разделе «Области поиска в API».

Использование поиска

Общий порядок работы с поиском в Deckhouse Code:

1. Нажмите «Поиск» в верхней панели.
2. Введите поисковый запрос.
3. Нажмите «Enter» — результаты появятся на странице поиска.
4. Используйте фильтры для уточнения результатов по группе, проекту или типу объекта.



Глобальный поиск

Позволяет искать по всем проектам и группам инстанса.

1. В левом меню выберите «Поиск».
2. Введите запрос и нажмите «Enter».

Поиск в проекте

1. Перейдите в нужный проект.
2. В левом меню выберите «Поиск».
3. Введите запрос и нажмите «Enter».

Поиск по группе

1. Перейдите в нужную группу.
2. В левом меню выберите «Поиск».
3. Введите запрос и нажмите «Enter».

Дополнительные возможности

- Поиск поддерживает автодополнение по проектам, группам и пользователям.
- При включённом расширенном поиске автодополнение также работает по сообщениям коммитов, именам файлов, коду, задачам и запросам на слияние.
- При поиске можно быстро перейти к нужному коммиту по его SHA.

Синтаксис

Расширенный поиск поддерживает расширенный синтаксис запросов: точные и нечёткие совпадения, логические операторы и фильтры.

Синтаксис	Описание	Пример
`"`	Точный поиск	`"gem sidekiq"`
`~`	Нечёткий поиск	`J~ Doe`
`|`	Или	`display | banner`
`+`	И	`display +banner`
`-`	Исключение	`display -banner`
`*`	Частичное совпадение	`bug error 50*`
`\`	Экранирование	`\*md`
`#`	ID задачи (в комментариях)	`#23456`
`!`	ID запроса на слияние (в комментариях)	`!23456`

Поиск по коду

Синтаксис	Описание	Пример
`filename:`	Имя файла	`filename:*spec.rb`
`path:`	Путь в репозитории (полное или частичное совпадение)	`path:spec/workers/`
`extension:`	Расширение файла без точки	`extension:js`
`blob:`	Git object ID	`blob:998707*`

В интерфейсе поиска по коду также доступен фильтр по языку программирования.

Примеры

Запрос	Описание
`rails -filename:gemfile.lock`	Находит `rails` во всех файлах, кроме `gemfile.lock`
`RSpec.describe Resolvers -*builder`	Находит `RSpec.describe Resolvers`, исключая совпадения, начинающиеся с `builder`
`bug | (display +banner)`	Находит `bug` или одновременно `display` и `banner`
`helper -extension:yml -extension:js`	Находит `helper` во всех файлах, кроме `.yml` и `.js`
`helper path:lib/git`	Находит `helper` в файлах с путём `lib/git*` (например, `spec/lib/gitlab`)

Настройки индексации

Настройки проекта

Maintainer проекта может перейти в «Настройки» → «Поиск».

Regex для веток

Если на уровне инстанса включён режим **Разрешить регулярное выражение для веток на уровне проекта**, maintainer может указать regex для дополнительных веток. Ветка по умолчанию индексируется всегда.

Пример regex: `(feature|hotfix)/.*`

> ⚠ Изменение regex запускает полную переиндексацию проекта.

Переиндексация code и wiki

- **Переиндексировать код** — полная переиндексация кода репозитория.
- **Переиндексировать вики** — полная переиндексация wiki (если wiki-репозиторий существует).

Бейдж «Индекс актуален» показывает, завершена ли индексация для текущего состояния репозитория.

Настройки группы

Owner группы может перейти в «Настройки» → «Поиск».

Доступна переиндексация wiki группы: статус индекса и кнопка «Переиндексировать вики».

API поиска

Search REST API позволяет выполнять поиск по инстансу Deckhouse Code, отдельной группе или проекту.

Эндпоинты

Для поиска доступны следующие эндпоинты:

- `GET /api/v4/search` — поиск по инстансу Deckhouse Code;
- `GET /api/v4/groups/:id/search` (или `/api/v4/groups/:id/-/search`) — поиск по группе;
- `GET /api/v4/projects/:id/search` (или `/api/v4/projects/:id/-/search`) — поиск по проекту.

Все эндпоинты требуют аутентификации.

Области поиска в API

Область поиска задаётся обязательным параметром `scope`. Поддерживаемые значения зависят от эндпоинта:

Значение `scope`	Инстанс	Группа	Проект	Бэкенд при включённом OpenSearch
`projects`				PostgreSQL
`users`				PostgreSQL
`snippet_titles`				PostgreSQL
`issues`				OpenSearch (`advanced`)
`work_items`				OpenSearch (`advanced`)
`merge_requests`				OpenSearch (`advanced`)
`milestones`				OpenSearch (`advanced`)
`notes`				OpenSearch (`advanced`)
`wiki_blobs`				OpenSearch (`advanced`)
`commits`				OpenSearch (`advanced`)
`blobs`				OpenSearch (`advanced`)

В заголовке ответа `X-Search-Type` возвращается фактически использованный тип поиска.

Набор областей в API отличается от областей поиска в веб-интерфейсе: значения `blobs`, `commits`, `notes` и `wiki_blobs` поддерживаются только эндпоинтом проекта, тогда как в интерфейсе поиск по этим областям доступен и глобально, и по группе.

Параметры запроса

Общие параметры

Параметр	Тип	Обязательный	Эндпоинты	Примечание
`search`	string	Да	Все	Поисковый запрос
`scope`	string	Да	Все	Область поиска. Доступные значения описаны в таблице выше
`confidential`	boolean	Нет	Все	Передаётся в службу поиска
`include_archived`	boolean	Нет	Инстанс, группа	Параметр недоступен для поиска по проекту
`page` / `per_page`	integer	Нет	Все	Постраничный вывод со смещением (offset)
`ref`	string	Нет	Проект	Ветка или тег для поиска в проекте
`state`	string	Нет	Все	Состояние объекта: `all`, `opened`, `closed`, `merged`
`type`	array[string]	Нет	Все	Фильтр типа work item (фактически применяется при `scope=work_items`)

Дополнительные параметры

Поддержка дополнительных параметров зависит от выбранной области поиска. Если параметр передан с неподдерживаемым значением `scope`, API возвращает ответ `400` с сообщением `<PARAM_NAME> is supported only for <SCOPE_LIST>`.

Параметр	Тип	Применяется к `scope`	Ограничения
`author_username`	string	`merge_requests`	Фильтр по автору
`exclude_forks`	boolean	`work_items`, `issues`	Только в этих `scope`
`fields`	array[string]	`work_items`, `issues`	Поддерживается только значение `title`. Для других значений API возвращает `400`
`label_name`	array[string]	`work_items`, `issues`, `merge_requests`	Поддерживаются значения через запятую
`language`	array[string]	`blobs`	Поддерживаются значения через запятую
`not_author_username`	string	`merge_requests`	Исключение по автору
`not_source_branch`	string	`merge_requests`	Исключающий фильтр
`not_target_branch`	string	`merge_requests`	Исключающий фильтр
`num_context_lines`	integer	`blobs`	Поддерживается диапазон `0..20`
`source_branch`	string	`merge_requests`	Точный фильтр по исходной ветке
`target_branch`	string	`merge_requests`	Точный фильтр по целевой ветке

Заголовки ответа

API может возвращать следующие заголовки:

- `X-Search-Type` — фактически использованный тип поиска;
- `X-Search-Aggregations` — присутствует только когда OpenSearch включён и для выбранной области поиска доступны агрегаты.

Состав агрегатов зависит от значения `scope`:

Значение `scope`	Агрегаты
`blobs`	`language`
`work_items`, `issues`	`work_item_type_ids`, `labels`
`merge_requests`	`labels`

Тело ответа

Эндпоинт возвращает JSON-массив объектов, тип которых зависит от выбранной области поиска:

Значение `scope`	Тип объекта
`issues`	`IssueBasic`
`work_items`	`WorkItem`
`merge_requests`	`MergeRequestBasic`
`milestones`	`Milestone`
`notes`	`Note`
`commits`	`Commit`
`blobs`	`Blob`
`wiki_blobs`	`Blob`
`projects`	`BasicProjectDetails`
`users`	`UserBasic`
`snippet_titles`	`Snippet`

Примеры запросов

Поиск по инстансу: issues/work items с метками и полями

```
curl --request GET \
  --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
  --url "https://code.example.com/api/v4/search?
scope=issues&search=deploy&fields=title&label_name=team%3Aplatform&exclude_forks=true"
```

Поиск по группе: запросы на слияние с фильтрами

```
curl --request GET \
  --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
  --url "https://code.example.com/api/v4/groups/my-group/-/search?
scope=merge_requests&search=release&source_branch=release%2F1.2&not_author_username=bo
t"
```

Поиск по проекту: blobs с контекстными строками

```
curl --request GET \
  --header "PRIVATE-TOKEN: <ACCESS_TOKEN>" \
  --url "https://code.example.com/api/v4/projects/my-group%2Fmy-project/-/search?
scope=blobs&search=deploy&num_context_lines=5&language=Ruby"
```

Справка по Deckhouse Code

REST API

Обзор

> Справка по REST API сгенерирована автоматически из спецификации OpenAPI 3.0 Deckhouse Code и представлена только на английском языке.

Операции сгруппированы по ресурсам, каждой группе отведена отдельная страница. Для каждой операции приведены параметры, тело запроса и ответы; после операций описаны схемы возвращаемых объектов.

Базовый адрес

Пути операций в справке начинаются с `/api/v4`. Отправляйте запросы на адрес Deckhouse Code, дополненный путём операции. Если используется шаблон публичных доменов `%s.example.com`, список проектов доступен по адресу `https://code.example.com/api/v4/projects`.

Аутентификация

Запрос аутентифицируется токеном, переданным одним из способов:

- в заголовке `Authorization: Bearer <TOKEN>` — для токена доступа или токена OAuth 2.0;
- в заголовке `PRIVATE-TOKEN: <TOKEN>` — для токена доступа;
- в параметре строки запроса `private_token=<TOKEN>` — для токена доступа.

Токены доступа — это персональные токены, токены групп и токены проектов.

Пример запроса с персональным токеном доступа:

```
curl --header "PRIVATE-TOKEN: <TOKEN>" "https://code.example.com/api/v4/projects"
```

OAuth 2.0

Приложение, зарегистрированное в Deckhouse Code, получает токен OAuth 2.0 по схеме с кодом авторизации (authorization code):

- пользователь перенаправляется на `https://code.example.com/oauth/authorize`;
- приложение обменивает код авторизации на токен по адресу `https://code.example.com/oauth/token`;
- приложение обновляет токен по тому же адресу, `https://code.example.com/oauth/token`.

Области, которые запрашивает приложение, ограничивают права токена:

Область	Доступ
`api`	Чтение и запись через API, включая все группы и проекты, хранилище образов контейнеров, прокси зависимостей и реестр пакетов
`read_api`	Чтение через API, включая все группы и проекты, хранилище образов контейнеров и реестр пакетов
`read_user`	Чтение профиля пользователя через эндпоинт `/user` (имя пользователя, публичный email и полное имя) и доступ к эндпоинтам `/users`, работающим только на чтение
`create_runner`	Создание раннеров
`manage_runner`	Управление раннерами
`k8s_proxy`	Вызовы Kubernetes API через агент для Kubernetes
`self_rotate`	Ротация токена самим токеном
`mcp`	Чтение и запись через сервер Model Context Protocol (MCP) для запуска инструментов
`mcp_orbit`	Доступ к MCP-серверу Orbit Knowledge Graph для запросов к графу и получения его схемы

Приложениями управляют операции из раздела Applications.

Постраничный вывод

Операции, которые возвращают списки, принимают параметр `page` (номер страницы, по умолчанию `1`) и параметр `per_page` (число элементов на странице, по умолчанию `20`). Если операция ограничивает `per_page`, ограничение указано в таблице параметров операции.

Пример запроса второй страницы из 50 проектов:

```
curl --header "PRIVATE-TOKEN: <TOKEN>" "https://code.example.com/api/v4/projects?page=2&per_page=50"
```

Файл спецификации

Файл спецификации OpenAPI содержит те же операции и схемы в формате JSON. Используйте его, чтобы импортировать операции в API-клиент, например Postman, или сгенерировать клиентскую библиотеку с помощью OpenAPI Generator.

Access requests

Use this API to interact with access requests for groups and projects.

List all access requests for a group

`GET /api/v4/groups/{id}/access_requests`

Lists all access requests for a specified group that are viewable by the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

Request access to a group

`POST /api/v4/groups/{id}/access_requests`

Requests access to a specified group for the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

Deny an access request

`DELETE /api/v4/groups/{id}/access_requests/{user_id}`

Denies an access request for a specified user in a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the user requesting access

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Approves an access request for the given user

`PUT /api/v4/groups/{id}/access_requests/{user_id}/approve`

This feature was introduced in GitLab 8.11.

Parameters

Name	Type	Description
`id` Path, required	String	The ID or URL-encoded path of the group owned by the authenticated user
`user_id` Path, required	Integer	The user ID of the access requester

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	A valid access level (defaults: `30` , the Developer role) Default: `30`

Responses

Code	Description	Schema
`200`	successful operation	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

List all access requests for a project

`GET /api/v4/projects/{id}/access_requests`

Lists all access requests for a specified project that are viewable by the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

Request access to a project

`POST /api/v4/projects/{id}/access_requests`

Requests access to a specified project for the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

Deny an access request

`DELETE /api/v4/projects/{id}/access_requests/{user_id}`

Denies an access request for a specified user in a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`user_id` Path, required	Integer	ID of the user requesting access

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Approves an access request for the given user

`PUT /api/v4/projects/{id}/access_requests/{user_id}/approve`

This feature was introduced in GitLab 8.11.

Parameters

Name	Type	Description
`id` Path, required	String	The ID or URL-encoded path of the project owned by the authenticated user
`user_id` Path, required	Integer	The user ID of the access requester

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	A valid access level (defaults: `30`, the Developer role) Default: `30`

Responses

Code	Description	Schema
`200`	successful operation	APIEntitiesAccessRequester
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAccessRequester

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`requested_at`	String	—
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

Access tokens

Use this API to interact with access tokens including:

- Personal access tokens.
- Group access tokens.
- Project access tokens.
- Impersonation tokens.

List all group access tokens

`GET /api/v4/groups/{id}/access_tokens`

Lists all group access tokens for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesResourceAccessToken
`400`	Bad Request	—
`404`	Not Found	—

Create a group access token

`POST /api/v4/groups/{id}/access_tokens`

Creates a group access token for a specified group. You cannot create a token with an access level greater than your account. For example, a user with the Maintainer role cannot create a group access token with the Owner role. You must use a personal access token with this endpoint. You cannot authenticate with a group access token.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group Example: `2`

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	The access level of the token in the group Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50` Default: `40`
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Date when the access token expires. If omitted, the token does not expire, which is accepted only when token expiry is not enforced Example: `2026-02-14`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	List of scopes available to the token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Rotate a group access token

`POST /api/v4/groups/{id}/access_tokens/self/rotate`

Rotates a group access token by passing it to the API in a header.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`405`	Method not allowed	—

Retrieve details on a group access token

`GET /api/v4/groups/{id}/access_tokens/{token_id}`

Retrieves details on a specified group access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`token_id` Path, required	String	ID of the token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a group access token

`DELETE /api/v4/groups/{id}/access_tokens/{token_id}`

Revokes a specified group access token.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`token_id` Path, required	String	ID of the token

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not found	—

Rotate a group access token

`POST /api/v4/groups/{id}/access_tokens/{token_id}/rotate`

Rotates a group access token. This immediately revokes the previous token and creates a token. Generally, this endpoint rotates a specific group access token by authenticating with a personal access token. You can also use a group access token to rotate itself. If you attempt to use this endpoint to rotate a token that was previously revoked, any active tokens from the same token family are revoked. This feature was introduced in GitLab 16.0.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`token_id` Path, required	String	ID of the token

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all personal access tokens for a group service account

`GET /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens`

Lists all personal access tokens for a specified group service account

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`user_id` Path, required	Integer	ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	Forbidden	—
`404`	404 Group Not Found	—

Create a personal access token for a group service account

`POST /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens`

Creates a personal access token for a specified group service account. Available only for group Owners and administrators. This feature was introduced in GitLab 16.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the service account

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	Array of approved scopes. For a list of possible values, see access token scopes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a personal access token for a group service account

`DELETE /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}`

Revokes a specified personal access token for a group service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the personal access token
`user_id` Path, required	Integer	ID of the service account

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Rotate a personal access token for a group service account

`POST /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}/rotate`

Rotates a specified personal access token for a group service account. This revokes the existing token and creates a token with the same name, description, and scopes.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the personal access token
`user_id` Path, required	Integer	ID of the service account

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. Introduced in GitLab 17.9. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all personal access tokens

`GET /api/v4/personal_access_tokens`

Lists all personal access tokens accessible by the authenticated user. For administrators, returns all personal access tokens in the instance. For non-administrators, returns all of their personal access tokens.

Parameters

Name	Type	Description
`user_id` Query	Integer	Filter personal access tokens by user ID. Non-administrators can only filter their own tokens Example: `2`
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`

Name	Type	Description
`search` Query	String	Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	Unauthorized	—

Retrieve a personal access token

`GET /api/v4/personal_access_tokens/self`

Retrieves a specified personal access token by passing it to the API in a header.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`401`	Unauthorized	—
`404`	Not found	—

Revoke a personal access token

`DELETE /api/v4/personal_access_tokens/self`

Revokes a personal access token by passing it to the API in a header.

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—

List all token associations

`GET /api/v4/personal_access_tokens/self/associations`

Lists all groups and projects accessible by the personal access token used to authenticate the request. Generally, this includes any groups or projects that the user is a member of.

Parameters

Name	Type	Description
`min_access_level` Query	Integer	Return groups and projects where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Rotate a personal access token

`POST /api/v4/personal_access_tokens/self/rotate`

Rotates a personal access token by passing it to the API in a header

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`405`	Method not allowed	—

Retrieve a personal access token

`GET /api/v4/personal_access_tokens/{id}`

Retrieves details for a specified personal access token. Administrators can retrieve details on any token. Non-administrators can only retrieve details on their own tokens.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the personal access token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Revoke a personal access token

`DELETE /api/v4/personal_access_tokens/{id}`

Revokes a specified personal access token. Administrators can revoke tokens for any user. Non-administrators can only revoke their own tokens.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the personal access token

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Rotate a personal access token

`POST /api/v4/personal_access_tokens/{id}/rotate`

Rotates a specified personal access token. This revokes the previous token and creates a token that expires after one week. Administrators can revoke tokens for any user. Non-administrators can only revoke their own tokens.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the personal access token

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all project access tokens

`GET /api/v4/projects/{id}/access_tokens`

Lists all project access tokens for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesResourceAccessToken
`400`	Bad Request	—
`404`	Not Found	—

Create a project access token

`POST /api/v4/projects/{id}/access_tokens`

Creates a project access token for a specified project. You cannot create a token with an access level greater than your account. For example, a user with the Maintainer role cannot create a project access token with the Owner role. You must use a personal access token with this endpoint. You cannot authenticate with a project access token.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `2`

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	The access level of the token in the project Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50` Default: `40`
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Date when the access token expires. If omitted, the token does not expire, which is accepted only when token expiry is not enforced Example: `2026-02-14`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	List of scopes available to the token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Rotate a project access token

`POST /api/v4/projects/{id}/access_tokens/self/rotate`

Rotates a project access token by passing it to the API in a header.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`405`	Method not allowed	—

Retrieve details on a project access token

`GET /api/v4/projects/{id}/access_tokens/{token_id}`

Retrieves details on a specified project access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`token_id` Path, required	String	ID of the token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a project access token

`DELETE /api/v4/projects/{id}/access_tokens/{token_id}`

Revokes a specified project access token.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`token_id` Path, required	String	ID of the token

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not found	—

Rotate a project access token

`POST /api/v4/projects/{id}/access_tokens/{token_id}/rotate`

Rotates a project access token. This immediately revokes the previous token and creates a token. Generally, this endpoint rotates a specific project access token by authenticating with a personal access token. You can also use a project access token to rotate itself. If you attempt to use this endpoint to rotate a token that was previously revoked, any active tokens from the same token family are revoked. This feature was introduced in GitLab 16.0.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`token_id` Path, required	String	ID of the token

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all personal access tokens for a project service account

`GET /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens`

Lists all personal access tokens for a specified project service account

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`user_id` Path, required	Integer	The ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	Forbidden	—
`404`	404 Project Not Found	—

Create a personal access token for a project service account

`POST /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens`

Creates a personal access token for a specified project service account. Available only for project Owners, Maintainers, and administrators.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`user_id` Path, required	Integer	The ID of the service account

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	The array of scopes of the personal access token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a personal access token for a project service account

`DELETE /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}`

Revokes a specified personal access token for a project service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`token_id` Path, required	Integer	The ID of the personal access token
`user_id` Path, required	Integer	The ID of the service account

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Rotate a personal access token for a project service account

`POST /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}/rotate`

Rotates a specified personal access token for a project service account. This revokes the existing token and creates a token with the same name, description, and scopes.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`token_id` Path, required	Integer	The ID of the personal access token
`user_id` Path, required	Integer	The ID of the service account

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	The expiration date of the token Example: `2021-01-31`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Create a personal access token

`POST /api/v4/user/personal_access_tokens`

Creates a personal access token for the currently authenticated user. For security purposes, the token is limited to the `k8s_proxy` and `self_rotate` scope. Token values are included with the response, but cannot be retrieved later. This feature was introduced in GitLab 16.5.

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`granular_scopes`	Array of objects	List of granular scopes to assign to the token. Exactly one of `scopes`, `granular_scopes` must be provided
`granular_scopes[].access` Required	String	Access to configure for the granular scope Allowed values: `personal_projects`, `all_memberships`, `selected_memberships`, `user`, `instance` Minimum length: `1`
`granular_scopes[].group_ids`	Array of integers	IDs of groups to associate with the granular scope, when access is `selected_memberships`
`granular_scopes[].permissions` Required	Array of strings	List of permissions for the granular scope
`granular_scopes[].project_ids`	Array of integers	IDs of projects to associate with the granular scope, when access is `selected_memberships`
`name` Required	String	Name of the access token Example: `My token`

Property	Type	Description
`scopes`	Array of strings	Array of approved scopes. For a list of possible values, see access token scopes. Exactly one of `scopes`, `granular_scopes` must be provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—

List all impersonation tokens for a user

`GET /api/v4/users/{user_id}/impersonation_tokens`

Lists all impersonation tokens for a specified user. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`state` Query	String	Filter tokens by state Allowed values: `all`, `active`, `inactive` Default: `all`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntities ImpersonationToken
`400`	Bad Request	—
`404`	Not Found	—

Create an impersonation token

`POST /api/v4/users/{user_id}/impersonation_tokens`

Creates an impersonation token. These tokens are used to act on behalf of a user and can perform API calls as well as Git read and write actions. These tokens are not visible to the associated user on their profile settings page. Token values are included with the response, but cannot be retrieved later. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the impersonation token. Limited to 255 characters
`expires_at`	String (date)	Date when the impersonation token expires. If omitted, the token does not expire, which is accepted only when token expiry is not enforced
`granular_scopes`	Array of objects	List of granular scopes to assign to the token. Mutually exclusive with `scopes`
`granular_scopes[].access` Required	String	Access to configure for the granular scope Allowed values: `personal_projects`, `all_memberships`, `selected_memberships`, `user`, `instance` Minimum length: `1`
`granular_scopes[].group_ids`	Array of integers	IDs of groups to associate with the granular scope, when access is `selected_memberships`
`granular_scopes[].permissions` Required	Array of strings	List of permissions for the granular scope
`granular_scopes[].project_ids`	Array of integers	IDs of projects to associate with the granular scope, when access is `selected_memberships`
`name` Required	String	Name of the impersonation token
`scopes`	Array of strings	Array of approved scopes. For a list of possible values, see access token scopes. Mutually

Property	Type	Description
		exclusive with `granular_scopes`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesImpersonationTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an impersonation token for a user

`GET /api/v4/users/{user_id}/impersonation_tokens/{impersonation_token_id}`

Retrieves an impersonation token for a specified user. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user
`impersonation_token_id` Path, required	Integer	ID of the impersonation token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesImpersonationToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke an impersonation token

`DELETE /api/v4/users/{user_id}/impersonation_tokens/{impersonation_token_id}`

Revokes an impersonation token for a specified user. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user
`impersonation_token_id` Path, required	Integer	ID of the impersonation token

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Create a personal access token for a user

`POST /api/v4/users/{user_id}/personal_access_tokens`

Creates a personal access token for a user. Token values are included with the response, but cannot be retrieved later. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	Array of approved scopes. For a list of possible values, see access token scopes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesImpersonationToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`impersonation`	String	—
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`user_id`	Integer (int64)	Example: `3`

APIEntitiesImpersonationTokenWithToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`impersonation`	String	—
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`token`	String	—
`user_id`	Integer (int64)	Example: `3`

APIEntitiesPersonalAccessToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`user_id`	Integer (int64)	Example: `3`

APIEntitiesPersonalAccessTokenGranularScope

Property	Type	Description
`access`	String	Example: `personal_projects`
`group_id`	Integer (int64)	Example: `5`
`permissions`	Array	Example: `["read_job"]`
`project_id`	Integer (int64)	Example: `3`

APIEntitiesPersonalAccessTokenWithToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`token`	String	—
`user_id`	Integer (int64)	Example: `3`

APIEntitiesResourceAccessToken

Property	Type	Description
`access_level`	Integer	Allowed values: `10`, `20`, `30`, `40`, `50` Example: `40`
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`resource_id`	Integer	Example: `1234`
`resource_type`	String	Allowed values: `project`, `group` Example: `project`
`revoked`	Boolean	—

Property	Type	Description
`scopes`	Array	Example: `["api"]`
`user_id`	Integer (int64)	Example: `3`

APIEntitiesResourceAccessTokenWithToken

Property	Type	Description
`access_level`	Integer	Allowed values: `10`, `20`, `30`, `40`, `50` Example: `40`
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`resource_id`	Integer	Example: `1234`
`resource_type`	String	Allowed values: `project`, `group` Example: `project`
`revoked`	Boolean	—

Property	Type	Description
`scopes`	Array	Example: `["api"]`
`token`	String	—
`user_id`	Integer (int64)	Example: `3`

Admin opensearch

Operations related to admin opensearch.

Get Sidekiq queue statistics for OpenSearch indexing: notes instance reindex worker, and repository indexer jobs with force: true only

`GET /api/v4/admin/opensearch/indexing_queue_stats`

Responses

Code	Description	Schema
`200`	OK	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Synchronously recreate OpenSearch index(es) and enqueue background reindex jobs

`POST /api/v4/admin/opensearch/recreate_indices`

schema_class must be one of: `recreate_all`, `Search::Opensearch::IndicesSchema::Code`, `Search::Opensearch::IndicesSchema::Wiki`, `Search::Opensearch::IndicesSchema::Note`, `Search::Opensearch::IndicesSchema::Milestone`, `Search::Opensearch::IndicesSchema::WorkItem`, `Search::Opensearch::IndicesSchema::MergeRequest`.

Request body (`application/json`)

Property	Type	Description
`schema_class` Required	String	One of: `recreate_all`, `Search::Opensearch::IndicesSchema::Code`, `Search::Opensearch::IndicesSchema::Wiki`, `Search::Opensearch::IndicesSchema::Note`, `Search::Opensearch::IndicesSchema::Milestone`, `Search::Opensearch::IndicesSchema::WorkItem`, `Search::Opensearch::IndicesSchema::MergeRequest` Allowed values: `recreate_all`, `Search::Opensearch::IndicesSchema::Code`, `Search::Opensearch::IndicesSchema::Wiki`, `Search::Opensearch::IndicesSchema::Note`, `Search::Opensearch::IndicesSchema::Milestone`, `Search::Opensearch::IndicesSchema::WorkItem`, `Search::Opensearch::IndicesSchema::MergeRequest` Minimum length: `1`

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	400 Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Agents

Use this API to retrieve the GitLab agents for Kubernetes that a CI/CD job is authorized to access.

List all GitLab agents for Kubernetes by job token

`GET /api/v4/job/allowed_agents`

Lists all GitLab agents for Kubernetes with a specified `CI_JOB_TOKEN`.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJob
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Ai agents

Operations related to ai agents.

Report the state of an AI agent task

`POST /api/v4/projects/{id}/ai_agents/tasks/{task_id}/report`

Called by the agent pod: the task is active, done or failed. Available only to the agent account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `12`
`task_id` Path, required	Integer	The task id the pod was handed Example: `7`

Request body (`application/json`)

Property	Type	Description
`error`	String	Why the task failed Example: `OpenCode timed out`
`result_url`	String	The merge request or note the task produced
`retry`	Boolean	false once the task must not be repeated after an interruption
`status` Required	String	active, done or failed Allowed values: `active` , `done` , `failed` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAiAgentTask
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAiAgentTask

Property	Type	Description
`id`	Integer	Example: `7`
`lease_expires_at`	String (date-time)	Example: `2026-10-10T12:00:00Z`
`state`	String	Allowed values: `pending`, `running`, `done`, `failed` Example: `running`

Alert management

Use this API to interact with metric images for alerts.

Additional endpoints are available with the GraphQL API.

List all metric images

`GET /api/v4/projects/{id}/alert_management_alerts/{alert_iid}/metric_images`

Lists all metric images for a specified alert.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `17`
`alert_iid` Path, required	Integer	The IID of the Alert Example: `23`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMetric Image
`400`	Bad Request	—
`404`	Not found	—

Upload a metric image

`POST /api/v4/projects/{id}/alert_management_alerts/{alert_iid}/metric_images`

Uploads a metric image for a specified alert.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `17`
`alert_iid` Path, required	Integer	The IID of the Alert Example: `23`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The image file to be uploaded
`url`	String	The url to view more metric info Example: `https://example.com/metric`
`url_text`	String	A description of the image or URL Example: `An example metric`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMetricImage
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize metric image file upload

`POST /api/v4/projects/{id}/alert_management_alerts/{alert_iid}/metric_images/authorize`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `17`
`alert_iid` Path, required	Integer	The IID of the Alert Example: `23`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Update a metric image

`PUT /api/v4/projects/{id}/alert_management_alerts/{alert_iid}/metric_images/{metric_image_id}`

Updates a specified metric image for an alert.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `17`
`alert_iid` Path, required	Integer	The IID of the Alert Example: `23`
`metric_image_id` Path, required	Integer	The ID of metric image Example: `42`

Request body (`multipart/form-data`)

Property	Type	Description
`url`	String	The url to view more metric info Example: `https://example.com/metric`
`url_text`	String	A description of the image or URL Example: `An example metric`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMetricImage
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable entity	—

Delete a metric image

`DELETE /api/v4/projects/{id}/alert_management_alerts/{alert_iid}/metric_images/{metric_image_id}`

Deletes a specified metric image for an alert.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `17`
`alert_iid` Path, required	Integer	The IID of the Alert Example: `23`
`metric_image_id` Path, required	Integer	The ID of metric image Example: `42`

Responses

Code	Description	Schema
`204`	No Content	APIEntitiesMetricImage
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable entity	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMetricImage

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-11-13T00:06:18.084Z`
`file_path`	String	Example: `/uploads/-/system/alert_metric_image/file/23/file.png`
`filename`	String	Example: `file.png`
`id`	Integer (int64)	Example: `23`
`url`	String	Example: `https://example.com/metric`
`url_text`	String	Example: `An example metric`

Applications

Use this API to manage instance-wide OAuth applications that:

- Use GitLab as an authentication provider.
- Allow access to GitLab resources on a user's behalf.

> You cannot use this API to manage group applications or individual user applications.

Get applications

`GET /api/v4/applications`

List all registered applications

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesApplication
`401`	Unauthorized	—
`403`	Forbidden	—

List all applications

`GET /api/v4/user/applications`

Lists all applications owned by the authenticated user.

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesApplication
`401`	Unauthorized	—
`403`	Forbidden	—

Create an application

`POST /api/v4/user/applications`

Creates a new OAuth application for the authenticated user. This feature was introduced in GitLab 19.0

Request body (`application/json`)

Property	Type	Description
`confidential`	Boolean	If `true` , the application can securely store client credentials, such as the client secret. Non-confidential applications (such as native mobile apps and Single Page Apps) might expose client credentials Default: `true`
`name` Required	String	Name of the application Example: `MyApplication`
`redirect_uri` Required	String	Redirect URI of the application Example: `https://redirect.uri`
`scopes` Required	String	Scopes available to the application. Separate multiple scopes with a space

Responses

Code	Description	Schema
`201`	Created	APIEntitiesApplicationWithSecret
`400`	Bad Request	—

Retrieve an application

`GET /api/v4/user/applications/{id}`

Retrieves details of a specific application owned by the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the application. Differs from `application_id`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplication
`400`	Bad Request	—
`404`	Not Found	—

Update an application

`PUT /api/v4/user/applications/{id}`

Updates an existing application owned by the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the application. Differs from `application_id`

Request body (`application/json`)

Property	Type	Description
`name`	String	Name of the application
`scopes`	String	Scopes available to the application. Separate multiple scopes with a space

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplication
`400`	Bad Request	—
`404`	Not Found	—

Delete an application

`DELETE /api/v4/user/applications/{id}`

Deletes a specified application owned by the authenticated user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the application. Differs from `application_id`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesApplication

Property	Type	Description
`application_id`	String	Example: `5832fc6e14300a0d962240a8144466eef4ee93ef0d218477e55f11cf12fc3737`
`application_name`	String	Example: `MyApplication`
`callback_url`	String	Example: `https://redirect.uri`
`confidential`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `1`
`scopes`	Array	Example: `["api","read_user"]`

APIEntitiesApplicationWithSecret

Property	Type	Description
`application_id`	String	Example: `5832fc6e14300a0d962240a8144466eef4ee93ef0d218477e55f11cf12fc3737`
`application_name`	String	Example: `MyApplication`
`callback_url`	String	Example: `https://redirect.uri`
`confidential`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `1`
`scopes`	Array	Example: `["api","read_user"]`
`secret`	String	Example: `ee1dd64b6adc89cf7e2c23099301ccc2c61b441064e9324d963c46902a85ec34`

Attestations

Use this API to retrieve and download provenance attestations for a project.

> The availability of these endpoints is controlled by a feature flag. They are available for testing, but not ready for production use.

Retrieve an attestation bundle

`GET /api/v4/projects/{id}/attestations/{attestation_iid}/download`

This feature was introduced in GitLab 18.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`attestation_iid` Path, required	String or integer	The iid of the attestation

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Artifact SHA-256 not found	—

List all attestations for a project

`GET /api/v4/projects/{id}/attestations/{subject_digest}`

Lists all attestations for a specified project and artifact hash. This feature was introduced in GitLab 18.7.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`subject_digest` Path, required	String	The SHA-256 hash of the artifact

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSupplyChainAttestation
`400`	Bad Request	—
`404`	Artifact SHA-256 not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesSupplyChainAttestation

Property	Type	Description
`build_id`	Integer (int64)	—
`created_at`	String (date-time)	Example: `2025-09-17T02:26:10.898Z`
`download_url`	String	—
`expire_at`	String (date-time)	Example: `2025-09-17T02:26:10.898Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `14`
`predicate_kind`	String	Example: `provenance`
`predicate_type`	String	Example: `https://slsa.dev/provenance/v1`
`project_id`	Integer (int64)	—
`status`	String	Example: `success`
`subject_digest`	String	Example: `5db1fee4b5703808c48078a76768b155b421b210c0761cd6a5d223f4d99f1eaa`
`updated_at`	String (date-time)	Example: `2025-09-17T02:26:10.898Z`

Audit events

Use this API to list and retrieve the audit events recorded for an instance or a group.

Get a list of audit events with optional filters and sorting

`POST /api/v4/admin/audit_events/search`

Note: The date range must be within a single calendar month. If created_after and created_before span multiple months, the created_before date will be adjusted to the end of the month of created_after.

Request body (`application/json`)

Property	Type	Description
`created_after`	String	Start date (inclusive) in ISO8601. Defaults to the start of the current month
`created_before`	String	End date (inclusive) in ISO8601. Defaults to the end of the current month
`entity_types`	Array of strings	Comma-separated list of entity types to filter (e.g., “User, Project”)
`page`	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page`	Integer	Number of items to list per page Default: `20` Example: `20`
`q`	String	Free-text search in custom messages
`sort`	String	Sort by creation date. Default is created_desc Allowed values: `created_asc` , `created_desc`

Responses

Code	Description	Schema
`201`	Created	Array of APIEntitiesAuditEvent
`400`	Bad Request	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAuditEvent

Property	Type	Description
`author_name`	String	—
`created_at`	String	—
`details`	String	—
`entity_id`	String	—
`entity_type`	String	—
`id`	String	—
`target_id`	String	—
`target_type`	String	—

Authn

Operations related to authn.

Return the OIDC ID-token claims for the caller identified by an IAM OAuth token

`GET /api/v4/iam/userinfo`

Authenticates exclusively via an IAM-issued OAuth JWT and returns the subset of OIDC claims documented as “Included in ID Token” in the OpenID Connect provider docs

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—

Avatars

Use this API to retrieve and upload avatars for users, groups, and projects.

Return avatar url for a user

`GET /api/v4/avatar`

Parameters

Name	Type	Description
`email` Query, required	String	Public email address of the user
`size` Query	Integer	Single pixel dimension for the avatar image. Only used for avatar lookups at Gravatar or a configured Libravatar server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAvatar
`400`	Bad Request	—

Download a group avatar

`GET /api/v4/groups/{id}/avatar`

Downloads a group avatar image.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`404`	Not Found	—

Download a project avatar

`GET /api/v4/projects/{id}/avatar`

Downloads a project avatar. You can access this endpoint without authentication if the project is publicly accessible. This feature was introduced in GitLab 16.9.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`404`	Not Found	—

Upload an avatar

`PUT /api/v4/user/avatar`

Uploads an avatar for the currently authenticated user. This feature was introduced in GitLab 17.0.

Request body (`multipart/form-data`)

Property	Type	Description
`avatar` Required	String (binary)	The avatar file (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAvatar
`400`	Bad Request	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAvatar

Property	Type	Description
`avatar_url`	String	—

Award emoji

Use this API to manage emoji reactions.

GitLab objects that accept emoji reactions are called awardables. You can react with emoji on the following resources:

- Epics
- Issues
- Merge requests
- Snippets
- Comments

List all emoji reactions for an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/award_emoji`

Lists all emoji reactions for a specified issue. This endpoint can be accessed without authentication if the issue is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	ID (`iid` for merge requests/ issues, `id` for snippets) of an awardable
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/award_emoji`

Adds an emoji reaction to an issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from an issue. This endpoint can be accessed without authentication if the issue is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from an issue

`DELETE /api/v4/projects/{id}/issues/{issue_iid}/award_emoji/{award_id}`

Deletes a specified emoji reaction from an issue. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all emoji reactions for an issue comment

`GET /api/v4/projects/{id}/issues/{issue_iid}/notes/{note_id}/award_emoji`

Lists all emoji reactions for a specified comment on an issue. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`note_id` Path, required	Integer	The ID of a note
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to an issue comment

`POST /api/v4/projects/{id}/issues/{issue_iid}/notes/{note_id}/award_emoji`

Adds an emoji reaction to a comment on an issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from an issue comment

`GET /api/v4/projects/{id}/issues/{issue_iid}/notes/{note_id}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from a comment on an issue. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from an issue comment

`DELETE /api/v4/projects/{id}/issues/{issue_iid}/notes/{note_id}/award_emoji/{award_id}`

Deletes a specified emoji reaction from a comment on an issue. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`issue_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all emoji reactions for a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/award_emoji`

Lists all emoji reactions for a specified merge request. This endpoint can be accessed without authentication if the merge request is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	ID (`iid` for merge requests/ issues, `id` for snippets) of an awardable
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/award_emoji`

Adds an emoji reaction to a merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from a merge request. This endpoint can be accessed without authentication if the merge request is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from a merge request

`DELETE /api/v4/projects/{id}/merge_requests/{merge_request_iid}/award_emoji/{award_id}`

Deletes a specified emoji reaction from a merge request. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all emoji reactions for a merge request comment

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/notes/{note_id}/award_emoji`

Lists all emoji reactions for a specified comment on a merge request. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`note_id` Path, required	Integer	The ID of a note
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to a merge request comment

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/notes/{note_id}/award_emoji`

Adds an emoji reaction to a comment on a merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from a merge request comment

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/notes/{note_id}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from a comment on a merge request. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from a merge request comment

`DELETE /api/v4/projects/{id}/merge_requests/{merge_request_iid}/notes/{note_id}/award_emoji/{award_id}`

Deletes a specified emoji reaction from a comment on a merge request. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all emoji reactions for a snippet

`GET /api/v4/projects/{id}/snippets/{snippet_id}/award_emoji`

Lists all emoji reactions for a specified snippet. This endpoint can be accessed without authentication if the snippet is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID (`iid` for merge requests/ issues, `id` for snippets) of an awardable
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to a snippet

`POST /api/v4/projects/{id}/snippets/{snippet_id}/award_emoji`

Adds an emoji reaction to a snippet.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from a snippet

`GET /api/v4/projects/{id}/snippets/{snippet_id}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from a snippet. This endpoint can be accessed without authentication if the snippet is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from a snippet

`DELETE /api/v4/projects/{id}/snippets/{snippet_id}/award_emoji/{award_id}`

Deletes a specified emoji reaction from a snippet. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all emoji reactions for a snippet comment

`GET /api/v4/projects/{id}/snippets/{snippet_id}/notes/{note_id}/award_emoji`

Lists all emoji reactions for a specified comment on a snippet. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`note_id` Path, required	Integer	The ID of a note
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Add an emoji reaction to a snippet comment

`POST /api/v4/projects/{id}/snippets/{snippet_id}/notes/{note_id}/award_emoji`

Adds an emoji reaction to a comment on a snippet.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet
`note_id` Path, required	Integer	The ID of a note

Request body (`application/json`)

Property	Type	Description
`name` Required	String	Name of the emoji without colons

Responses

Code	Description	Schema
`201`	Created	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an emoji reaction from a snippet comment

`GET /api/v4/projects/{id}/snippets/{snippet_id}/notes/{note_id}/award_emoji/{award_id}`

Retrieves a specified emoji reaction from a comment on a snippet. This endpoint can be accessed without authentication if the comment is publicly accessible.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of the emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAwardEmoji
`400`	Bad Request	—
`404`	Not Found	—

Delete an emoji reaction from a snippet comment

`DELETE /api/v4/projects/{id}/snippets/{snippet_id}/notes/{note_id}/award_emoji/{award_id}`

Deletes a specified emoji reaction from a comment on a snippet. Only an administrator or the user who added the reaction can delete it.

Parameters

Name	Type	Description
`award_id` Path, required	Integer	ID of an emoji reaction
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`snippet_id` Path, required	Integer	ID of the snippet
`note_id` Path, required	Integer	The ID of a note

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAwardEmoji

Property	Type	Description
`awardable_id`	Integer (int64)	Example: `42`
`awardable_type`	String	Example: `Issue`
`created_at`	String (date-time)	Example: `2019-01-10T13:39:08Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `lizard`
`updated_at`	String (date-time)	Example: `2020-06-28T10:52:04Z`
`url`	String	Example: `https://example.com/emojis/example.gif`
`user`	APIEntitiesUserBasic	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Badges

Use this API to manage badges for projects and groups.

Placeholders

Badges support placeholders that are replaced in real time in both the link and image URL. The following placeholders are available:

- `%{project_path}` : replaced by the project path.
- `%{project_title}` : replaced by the project title.
- `%{project_name}` : replaced by the project name.
- `%{project_id}` : replaced by the project ID.
- `%{project_namespace}` : replaced by the project's namespace full path.
- `%{group_name}` : replaced by the project's top-level group name.
- `%{gitlab_server}` : replaced by the project's server name.
- `%{gitlab_pages_domain}` : replaced by the domain name hosting GitLab Pages.
- `%{default_branch}` : replaced by the project default branch.

- `%{commit_sha}` : replaced by the project's last commit SHA.
- `%{latest_tag}` : replaced by the project's last tag.

Group badge endpoints operate outside a project's context, so placeholder values are taken from the group's earliest-created project. If no projects exist in the group, the URL is returned unchanged.

List all badges for a group

`GET /api/v4/groups/{id}/badges`

Lists all badges for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`name` Query	String	Name of the badge

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Create a badge for a group

`POST /api/v4/groups/{id}/badges`

Creates a badge for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`image_url` Required	String	URL of the badge image
`link_url` Required	String	URL of the badge link
`name`	String	Name of the badge

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a badge preview for a group

`GET /api/v4/groups/{id}/badges/render`

Previews the final `link_url` and `image_url` for a specified group after resolving the placeholder interpolation.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`link_url` Query, required	String	URL of the badge link
`image_url` Query, required	String	URL of the badge image

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicBadgeDetails
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a badge for a group

`GET /api/v4/groups/{id}/badges/{badge_id}`

Retrieves a specified badge for a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`badge_id` Path, required	Integer	ID of the badge

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Update a badge for a group

`PUT /api/v4/groups/{id}/badges/{badge_id}`

Updates a specified badge for a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`badge_id` Path, required	Integer	ID of the badge

Request body (`application/json`)

Property	Type	Description
`image_url`	String	URL of the badge image
`link_url`	String	URL of the badge link
`name`	String	Name of the badge

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Delete a badge from a group

`DELETE /api/v4/groups/{id}/badges/{badge_id}`

Deletes a specified badge from a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`badge_id` Path, required	Integer	ID of the badge

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

List all badges for a project

`GET /api/v4/projects/{id}/badges`

Lists all badges for a specified project, including group badges.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`name` Query	String	Name of the badge

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Create a badge for a project

`POST /api/v4/projects/{id}/badges`

Creates a badge for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`image_url` Required	String	URL of the badge image
`link_url` Required	String	URL of the badge link
`name`	String	Name of the badge

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a badge preview for a project

`GET /api/v4/projects/{id}/badges/render`

Previews the final `link_url` and `image_url` for a specified project after resolving the placeholder interpolation.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`link_url` Query, required	String	URL of the badge link
`image_url` Query, required	String	URL of the badge image

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicBadgeDetails
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a badge for a project

`GET /api/v4/projects/{id}/badges/{badge_id}`

Retrieves a specified badge for a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`badge_id` Path, required	Integer	ID of the badge

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Update a badge for a project

`PUT /api/v4/projects/{id}/badges/{badge_id}`

Updates a specified badge for a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`badge_id` Path, required	Integer	ID of the badge

Request body (`application/json`)

Property	Type	Description
`image_url`	String	URL of the badge image
`link_url`	String	URL of the badge link
`name`	String	Name of the badge

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBadge
`400`	Bad Request	—
`404`	Not Found	—

Delete a badge from a project

`DELETE /api/v4/projects/{id}/badges/{badge_id}`

Deletes a specified badge from a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`badge_id` Path, required	Integer	ID of the badge

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBadge

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`image_url`	String	Example: `https://example.gitlab.com`
`kind`	String	Example: `project`
`link_url`	String	Example: `https://example.gitlab.com`
`name`	String	Example: `Pipeline Status`
`rendered_image_url`	String	Example: `https://example.gitlab.com`
`rendered_link_url`	String	Example: `https://example.gitlab.com`

APIEntitiesBasicBadgeDetails

Property	Type	Description
`image_url`	String	Example: `https://example.gitlab.com`
`link_url`	String	Example: `https://example.gitlab.com`
`name`	String	Example: `Pipeline Status`
`rendered_image_url`	String	Example: `https://example.gitlab.com`
`rendered_link_url`	String	Example: `https://example.gitlab.com`

Batched background migrations

Use this API to list, retrieve, pause, and resume the batched background migrations running on an instance.

These endpoints require administrator access to the instance.

Get the list of batched background migrations

`GET /api/v4/admin/batched_background_migrations`

FE version

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`status` Query	String	Filter by status Allowed values: `paused`, `active`, `finished`, `failed`, `finalizing`, `finalized`
`database` Query	String	The name of the database, the default `main` Allowed values: `main`, `ci`, `sec` Default: `main`
`job_class_name` Query	String	Filter migrations by job class name

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBatchedBackgroundMigration
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Retrieve a batched background migration

`GET /api/v4/admin/batched_background_migrations/{id}`

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background migration id
`database` Query	String	The name of the database Allowed values: `main`, `ci`, `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundMigration
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—

Pause a batched background migration

`PUT /api/v4/admin/batched_background_migrations/{id}/pause`

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background migration id

Request body (`application/json`)

Property	Type	Description
`database`	String	The name of the database Allowed values: `main`, `ci`, `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundMigration
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—
`422`	You can pause only `active` batched background migrations.	—

Resume a batched background migration

`PUT /api/v4/admin/batched_background_migrations/{id}/resume`

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background migration id

Request body (`application/json`)

Property	Type	Description
`database`	String	The name of the database Allowed values: `main`, `ci`, `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundMigration
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—
`422`	You can resume only `paused` batched background migrations.	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBatchedBackgroundMigration

Property	Type	Description
`column_name`	String	Example: `id`
`created_at`	String (date-time)	Example: `2022-11-28T16:26:39+02:00`
`estimated_time_remaining`	String	Example: `1 day`
`id`	String	Example: `1234`
`job_class_name`	String	Example: `CopyColumnUsingBackgroundMigrationJob`
`progress`	Number	Example: `50`
`status`	String	Example: `active`
`table_name`	String	Example: `events`

Batched background operations

Use this API to list, retrieve, restart, and stop the batched background operations running on an instance.

These endpoints require administrator access to the instance.

Get the list of batched background operations

`GET /api/v4/admin/batched_background_operations`

This feature was introduced in GitLab 19.1.

Parameters

Name	Type	Description
`database` Query	String	The name of the database, the default `main` Allowed values: `main`, `ci`, `sec` Default: `main`
`job_class_name` Query	String	Filter operations by job class name

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBatchedBackgroundOperation
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Retrieve a batched background operation

`GET /api/v4/admin/batched_background_operations/{id}`

This feature was introduced in GitLab 19.1.

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background operation id
`database` Query	String	The name of the database Allowed values: `main`, `ci`, `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundOperation
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—

Restart a batched background operation

`PUT /api/v4/admin/batched_background_operations/{id}/restart`

This feature was introduced in GitLab 19.2.

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background operation id

Request body (`application/json`)

Property	Type	Description
`database`	String	The name of the database Allowed values: `main` , `ci` , `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundOperation
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—
`422`	You can restart only `stopped` operations.	—

Stop a batched background operation

`PUT /api/v4/admin/batched_background_operations/{id}/stop`

This feature was introduced in GitLab 19.2.

Parameters

Name	Type	Description
`id` Path, required	Integer	The batched background operation id

Request body (`application/json`)

Property	Type	Description
`database`	String	The name of the database Allowed values: `main` , `ci` , `sec` Default: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBatchedBackgroundOperation
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—
`422`	You can stop only `queued`, `active` or `paused` operations.	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBatchedBackgroundOperation

Property	Type	Description
`column_name`	String	Example: `id`
`created_at`	String (date-time)	Example: `2025-05-15T10:00:00Z`
`finished_at`	String (date-time)	Example: `2025-05-15T11:00:00Z`
`id`	String	Example: `<cluster>:<partition_id>:<id/uuid>`
`job_class_name`	String	Example: `UsersDeleteUnconfirmedSecondaryEmails`
`on_hold_until`	String (date-time)	Example: `2025-05-15T10:15:00Z`
`partition`	Integer	Example: `1`
`started_at`	String (date-time)	Example: `2025-05-15T10:05:00Z`
`status`	String	Example: `active`
`table_name`	String	Example: `users`

Boards

Use this API to manage issue boards and their lists, for both projects and groups.

Every call to this API requires authentication. If the project or group is private and the authenticated user is not a member, a `GET` request results in a `404` status code.

List all group issue boards in a group

`GET /api/v4/groups/{id}/boards`

Lists all group issue boards for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a group issue board

`GET /api/v4/groups/{id}/boards/{board_id}`

Retrieves a specified group issue board.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Update a group issue board

`PUT /api/v4/groups/{id}/boards/{board_id}`

Updates a specified group issue board.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board

Request body (`application/json`)

Property	Type	Description
`hide_backlog_list`	Boolean	Hide the Open list
`hide_closed_list`	Boolean	Hide the Closed list
`name`	String	The board name

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

List all group issue board lists

`GET /api/v4/groups/{id}/boards/{board_id}/lists`

Lists all group issue board lists for a specified board. Does not include `open` and `closed` lists.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Create a group issue board list

`POST /api/v4/groups/{id}/boards/{board_id}/lists`

Creates a group issue board list for a specified board.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board

Request body (`application/json`)

Property	Type	Description
`label_id` Required	Integer	The ID of an existing label

Responses

Code	Description	Schema
`201`	Created	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a group issue board list

`GET /api/v4/groups/{id}/boards/{board_id}/lists/{list_id}`

Retrieves a specified group issue board list.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board
`list_id` Path, required	Integer	The ID of a list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Update a group issue board list

`PUT /api/v4/groups/{id}/boards/{board_id}/lists/{list_id}`

Updates a specified group issue board list. This call is used to change list position.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board
`list_id` Path, required	Integer	The ID of a list

Request body (`application/json`)

Property	Type	Description
`position` Required	Integer	The position of the list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Delete a group issue board list

`DELETE /api/v4/groups/{id}/boards/{board_id}/lists/{list_id}`

Deletes a specified group issue board list. Only for administrators and users with the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`board_id` Path, required	Integer	The ID of a board
`list_id` Path, required	Integer	The ID of a board list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

List all project issue boards

`GET /api/v4/projects/{id}/boards`

Lists all issue boards in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Create an issue board

`POST /api/v4/projects/{id}/boards`

Creates an issue board in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`name` Required	String	The board name

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an issue board

`GET /api/v4/projects/{id}/boards/{board_id}`

Retrieves a specified issue board in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Update an issue board

`PUT /api/v4/projects/{id}/boards/{board_id}`

Updates a specified issue board in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board

Request body (`application/json`)

Property	Type	Description
`hide_backlog_list`	Boolean	Hide the Open list
`hide_closed_list`	Boolean	Hide the Closed list
`name`	String	The board name

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

Delete an issue board

`DELETE /api/v4/projects/{id}/boards/{board_id}`

Deletes a specified issue board in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBoard
`400`	Bad Request	—
`404`	Not Found	—

List all board lists in an issue board

`GET /api/v4/projects/{id}/boards/{board_id}/lists`

Lists all lists in a specified issue board. Does not include `open` and `closed` lists.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Create an issue board list

`POST /api/v4/projects/{id}/boards/{board_id}/lists`

Creates an issue board list.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board

Request body (`application/json`)

Property	Type	Description
`label_id` Required	Integer	The ID of an existing label

Responses

Code	Description	Schema
`201`	Created	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a board list

`GET /api/v4/projects/{id}/boards/{board_id}/lists/{list_id}`

Retrieves a specified list from an issue board.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board
`list_id` Path, required	Integer	ID of the board's list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Update position of a board list

`PUT /api/v4/projects/{id}/boards/{board_id}/lists/{list_id}`

Updates the position of a specified list from an issue board.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board
`list_id` Path, required	Integer	ID of the board's list

Request body (`application/json`)

Property	Type	Description
`position` Required	Integer	Position of the list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Delete a list from an issue board

`DELETE /api/v4/projects/{id}/boards/{board_id}/lists/{list_id}`

Deletes a specified list from an issue board.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`board_id` Path, required	Integer	ID of the board
`list_id` Path, required	Integer	ID of the board's list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesList
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesBoard

Property	Type	Description
`hide_backlog_list`	Boolean	Example: `false`
`hide_closed_list`	Boolean	Example: `false`
`id`	Integer (int64)	Example: `1`
`lists`	Array of APIEntitiesList	—
`name`	String	Example: `Development`
`project`	APIEntitiesBasicProjectDetails	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesLabelBasic

Property	Type	Description
`archived`	Boolean	Example: `false`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `bug`
`text_color`	String	Example: `#FFFFFF`

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesList

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`label`	APIEntitiesLabelBasic	—
`position`	Integer	Example: `1`

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

Branches

Use this API to manage Git branches.

To change the branch protections configured for a project, use the protected branches API.

List all repository branches

`GET /api/v4/projects/{id}/repository/branches`

Lists all repository branches from a specified project, sorted alphabetically by name. Search by name, or use regular expressions to find specific branch patterns. Returns detailed information about the branch, including its protection status, merge status, and commit details.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`search` Query	String	Return branches containing the search string. Use `^term` to find branches that begin with `term`, and `term$` to find branches that end with `term`
`regex` Query	String	Return branches with names matching an RE2 regular expression. Cannot be used together with `search`
`sort` Query	String	Sort results by the specified field and direction Allowed values: `name_asc`, `updated_asc`, `updated_desc`
`page_token` Query	String	Name of the branch to start the pagination from. Used for keyset pagination

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBranch
`400`	Bad Request	—
`404`	404 Project Not Found	—

Create a repository branch

`POST /api/v4/projects/{id}/repository/branches`

Creates a branch in the repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`branch` Required	String	Name of the branch
`ref` Required	String	Branch name or commit SHA to create the branch from

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBranch
`400`	Branch already exists	—
`404`	Not Found	—

Retrieve a repository branch

`GET /api/v4/projects/{id}/repository/branches/{branch}`

Retrieves a specified project repository branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`branch` Path, required	String	Name of the branch

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBranch
`400`	Bad Request	—
`404`	Project Not Found	—

Delete a repository branch

`DELETE /api/v4/projects/{id}/repository/branches/{branch}`

Deletes a specified branch from the repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`branch` Path, required	String	Name of the branch

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Branch Not Found	—

Check if a branch exists

`HEAD /api/v4/projects/{id}/repository/branches/{branch}`

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`branch` Path, required	String	Name of the branch

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Protect a single branch

`PUT /api/v4/projects/{id}/repository/branches/{branch}/protect`

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`branch` Path, required	String	Name of the branch

Request body (`application/json`)

Property	Type	Description
`developers_can_merge`	Boolean	If `true` , allows developers to merge to the branch
`developers_can_push`	Boolean	If `true` , allows developers to push to the branch

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBranch
`400`	Bad Request	—
`404`	404 Branch Not Found	—

Unprotect a single branch

`PUT /api/v4/projects/{id}/repository/branches/{branch}/unprotect`

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`branch` Path, required	String	Name of the branch

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBranch
`400`	Bad Request	—
`404`	404 Branch Not Found	—

Delete all merged branches

`DELETE /api/v4/projects/{id}/repository/merged_branches`

Deletes all branches that are merged into the default branch for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`202`	202 Accepted	—
`400`	Bad Request	—
`404`	404 Project Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBranch

Property	Type	Description
`can_push`	Boolean	Example: `true`
`commit`	APIEntitiesCommit	—
`default`	Boolean	Example: `true`
`developers_can_merge`	Boolean	Example: `true`
`developers_can_push`	Boolean	Example: `true`
`merged`	Boolean	Example: `true`
`name`	String	Example: `master`
`protected`	Boolean	Example: `true`
`web_url`	String	Example: `https://gitlab.example.com/Commit921/the-dude/-/tree/master`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

Broadcast messages

Use this API to interact with banners and notifications displayed in the UI. For more information, see broadcast messages.

GET requests do not require authentication. All other broadcast message API endpoints are accessible only to administrators. Non-GET requests by:

- Unauthenticated users result in `401 Unauthorized`.
- Authenticated non-administrators result in `403 Forbidden`.

List all broadcast messages

`GET /api/v4/broadcast_messages`

Lists all broadcast messages for the instance.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSystemBroadcastMessage
`400`	Bad Request	—

Create a broadcast message

`POST /api/v4/broadcast_messages`

Creates a broadcast message.

Request body (`application/json`)

Property	Type	Description
`broadcast_type`	String	Appearance type of the broadcast message Allowed values: `banner`, `notification` Default: `banner`
`color`	String	Background color hex code. Deprecated. Use `theme` instead
`dismissable`	Boolean	If `true`, the user can dismiss the message
`ends_at`	String (date-time)	Ending time, in UTC. When creating a message, defaults to one hour after the current time
`font`	String	Foreground color hex code. Deprecated. Use `theme` instead
`message` Required	String	Message to display
`starts_at`	String (date-time)	Starting time, in UTC. When creating a message, defaults to the current time
`target_access_levels`	Array of integers	Target access levels (roles) of the broadcast message
`target_path`	String	Target path of the broadcast message
`theme`	String	Color theme for the broadcast message. Applies only to banners Allowed values: `indigo`, `light-indigo`, `blue`, `light-blue`, `green`, `light-green`, `red`, `light-red`, `dark`, `light`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesSystemBroadcastMessage
`400`	Bad Request	—

Retrieve a broadcast message

`GET /api/v4/broadcast_messages/{id}`

Retrieves a specified broadcast message.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the broadcast message

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSystemBroadcastMessage
`400`	Bad Request	—
`404`	Not Found	—

Update a broadcast message

`PUT /api/v4/broadcast_messages/{id}`

Updates a specified broadcast message.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the broadcast message

Request body (`application/json`)

Property	Type	Description
`broadcast_type`	String	Appearance type of the broadcast message Allowed values: `banner`, `notification`
`color`	String	Background color hex code. Deprecated. Use `theme` instead
`dismissable`	Boolean	If `true`, the user can dismiss the message
`ends_at`	String (date-time)	Ending time, in UTC. When creating a message, defaults to one hour after the current time
`font`	String	Foreground color hex code. Deprecated. Use `theme` instead
`message`	String	Message to display
`starts_at`	String (date-time)	Starting time, in UTC. When creating a message, defaults to the current time
`target_access_levels`	Array of integers	Target access levels (roles) of the broadcast message
`target_path`	String	Target path of the broadcast message
`theme`	String	Color theme for the broadcast message. Applies only to banners Allowed values: `indigo`, `light-indigo`, `blue`, `light-blue`, `green`, `light-green`, `red`, `light-red`, `dark`, `light`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSystemBroadcastMessage
`400`	Bad Request	—
`404`	Not Found	—

Delete a broadcast message

`DELETE /api/v4/broadcast_messages/{id}`

Deletes a specified broadcast message.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the broadcast message

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSystemBroadcastMessage
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesSystemBroadcastMessage

Property	Type	Description
`active`	Boolean	—
`broadcast_type`	String	Example: `banner`
`color`	String	Example: `#E75E40`
`dismissable`	Boolean	—
`ends_at`	String (date-time)	Example: `2016-01-06T15:39:55.570Z`
`font`	String	Example: `#FFFFFF`
`id`	Integer (int64)	—
`message`	String	Example: `Example broadcast message`
`starts_at`	String (date-time)	Example: `2016-01-04T15:39:55.570Z`
`target_access_levels`	Array	Example: `[10,30]`
`target_path`	String	Example: `*/welcome`
`theme`	String	Example: `indigo`

CI catalog

Use this API to publish a release of a component project to the CI/CD catalog.

This endpoint is intended for CLI tooling and must be authenticated with a job token.

Publish a new component project release as version to the CI/CD catalog

`POST /api/v4/projects/{id}/catalog/publish`

Publishes a release of a catalog resource as version to the CI/CD catalog.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`metadata` Required	Object	Metadata for the release

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiCatalogResourcesVersion
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiCatalogResourcesVersion

Property	Type	Description
`catalog_url`	String	Example: `https://gitlab.example.com/explore/catalog/my-namespace/my-component-project`

CI jobs

Use this API to interact with CI/CD jobs.

Retrieve a job by job token

`GET /api/v4/job`

Retrieves a job that was generated by a specified job token.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJob
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

List all jobs for a project

`GET /api/v4/projects/{id}/jobs`

Lists all jobs for a specified project. By default, this request returns 20 results at a time because the API results are paginated.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`scope` Query	Array of strings	Filter jobs by status. Accepts a single value or an array of values. If omitted, returns all jobs Example: `["pending", "running"]`
`ref` Query	String	Branch name (ref) to filter jobs by Example: `feature-branch`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a job

`GET /api/v4/projects/{id}/jobs/{job_id}`

Retrieves a job with the specified job ID.

Parameters

Name	Type	Description
`job_id` Path, required	Integer	ID of the job Example: `88`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Cancel a job

`POST /api/v4/projects/{id}/jobs/{job_id}/cancel`

Cancels a specified job in a project.

Parameters

Name	Type	Description
`job_id` Path, required	Integer	ID of the job Example: `88`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`force`	Boolean	If `true` , forces cancellation of a job in `canceling` state Example: `true`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Erase a job

`POST /api/v4/projects/{id}/jobs/{job_id}/erase`

Erases a specified job in a project. This removes job artifacts and the job log.

Parameters

Name	Type	Description
`job_id` Path, required	Integer	ID of the job Example: `88`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`429`	Too Many Requests	—

Run a job

`POST /api/v4/projects/{id}/jobs/{job_id}/play`

Runs a specified job. For a job in manual status, triggers an action to start the job.

Parameters

Name	Type	Description
`job_id` Path, required	Integer	ID of the job Example: `88`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`job_inputs`	Object	Hash of job input values to use when playing the job. Introduced in GitLab 18.10 Example: `{"environment": "production"}`
`job_variables_attributes`	Array of objects	User-defined variables to include when running the job
`job_variables_attributes[].key` Required	String	Name of the variable Example: `foo`
`job_variables_attributes[].value` Required	String	Value of the variable Example: `bar`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJobBasic
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too Many Requests	—

Retry a job

`POST /api/v4/projects/{id}/jobs/{job_id}/retry`

Retries a specified job in a project.

Parameters

Name	Type	Description
`job_id` Path, required	Integer	ID of the job Example: `88`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`inputs`	Object	Input values for the job Example: `{"environment": "production"}`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too Many Requests	—

Get a trace of a specific job of a project

`GET /api/v4/projects/{id}/jobs/{job_id}/trace`

Retrieves a log file for a job.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job Example: `88`
`byte_offset` Query	Integer	Byte offset to start reading the trace from Minimum: `0` Example: `0`
`byte_limit` Query	Integer	Maximum number of bytes to return Maximum: `512000` Minimum: `1` Example: `51200`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobBasic

Property	Type	Description
`allow_failure`	Boolean	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—

Property	Type	Description
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

CI lint

Use this API to validate a GitLab CI/CD configuration.

These endpoints take JSON-encoded YAML content. To preserve the formatting of your CI/CD configuration, it can help to escape and encode the YAML with a third-party tool such as `jq` before you make the request.

Validate existing CI/CD configuration

`GET /api/v4/projects/{id}/ci/lint`

Validates the `.gitlab-ci.yml` configuration for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Query	String	Commit SHA, branch, or tag that the CI/CD configuration content is taken from. Defaults to the head of the project's default branch. Deprecated. Use `content_ref` instead. Mutually exclusive with `content_ref`
`content_ref` Query	String	Commit SHA, branch, or tag that the CI/CD configuration content is taken from. Defaults to the head of the project's default branch. Mutually exclusive with `sha`
`dry_run` Query	Boolean	If `true`, runs a pipeline creation simulation. If `false`, runs only a static check Default: `false`
`include_jobs` Query	Boolean	If `true`, includes the list of jobs that would exist in a static check or pipeline simulation
`ref` Query	String	If `dry_run` is `true`, sets the branch or tag to use to validate the CI/CD YAML configuration. Defaults to the project's default branch. Deprecated. Use `dry_run_ref` instead. Mutually exclusive with `dry_run_ref`
`dry_run_ref` Query	String	If `dry_run` is `true`, sets the branch or tag to use to validate the CI/CD YAML configuration. Defaults to the project's default branch. Mutually exclusive with `ref`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiLintResult
`400`	Bad Request	—
`404`	Not found	—

Validate a CI/CD configuration

`POST /api/v4/projects/{id}/ci/lint`

Validates a provided CI/CD configuration in the context of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`content` Required	String	CI/CD configuration content
`dry_run`	Boolean	If `true` , runs a pipeline creation simulation. If `false` , runs only a static check Default: `false`
`include_jobs`	Boolean	If `true` , includes the list of jobs that would exist in a static check or pipeline simulation
`ref`	String	If `dry_run` is `true` , sets the branch or tag to use to validate the CI/CD YAML configuration. Defaults to the project's default branch

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiLintResult
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiLintResult

Property	Type	Description
`errors`	Array of strings	Example: `["variables config should be a hash of key value pairs"]`
`includes`	Array of APIEntitiesCiLintResultInclude	Example: `[{"blob":"https://gitlab.com/root/example-project/-/blob/..."}]`
`jobs`	Array of objects	Example: `[{"name":"test", "script":["ls"]}]`
`merged_yaml`	String	Example: `---\n:another_test:\n :stage: test\n :script: echo 2\n:test:\n :stage: test\n :script: echo 1\n`
`valid`	Boolean	—
`warnings`	Array of strings	Example: `["jobs:job may allow multiple pipelines ..."]`

APIEntitiesCiLintResultInclude

Property	Type	Description
`blob`	String	Example: `https://gitlab.com/gitlab-org/gitlab/-/blob/e52d6d0246d7375291850e61f0abc101fbda9dc2/.gitlab/ci/build-images.gitlab-ci.yml`
`context_project`	String	Example: `gitlab-org/gitlab`
`context_sha`	String	Example: `e52d6d0246d7375291850e61f0abc101fbda9dc2`
`extra`	Object	Example: `{"job_name":"test","project":"gitlab-org/gitlab","ref":"master"}`
`location`	String	Example: `.gitlab/ci/build-images.gitlab-ci.yml`
`raw`	String	Example: `https://gitlab.com/gitlab-org/gitlab/-/raw/e52d6d0246d7375291850e61f0abc101fbda9dc2/.gitlab/ci/build-images.gitlab-ci.yml`
`type`	String	Example: `local`

CI resource groups

Use this API to interact with resource groups.

List all resource groups

`GET /api/v4/projects/{id}/resource_groups`

Lists all resource groups for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiResourceGroup
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a resource group

`GET /api/v4/projects/{id}/resource_groups/{key}`

Retrieves a specified resource group for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	URL-encoded key of the resource group, for example `resource%5Fa` instead of `resource_a`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiResourceGroup
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a resource group

`PUT /api/v4/projects/{id}/resource_groups/{key}`

Updates the properties for a specified resource group. It returns `200` if the resource group was successfully updated. In case of an error, a status code `400` is returned.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	URL-encoded key of the resource group, for example `resource%5Fa` instead of `resource_a`

Request body (`application/json`)

Property	Type	Description
`process_mode`	String	Process mode of the resource group. Read process modes for more information Allowed values: `unordered`, `oldest_first`, `newest_first`, `newest_ready_first`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiResourceGroup
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve current job for a resource group

`GET /api/v4/projects/{id}/resource_groups/{key}/current_job`

Retrieves the current job for a specified resource group in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	URL-encoded key of the resource group, for example `resource%5Fa` instead of `resource_a`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJobBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all upcoming jobs for a resource group

`GET /api/v4/projects/{id}/resource_groups/{key}/upcoming_jobs`

Lists all upcoming jobs for a specified resource group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	URL-encoded key of the resource group, for example `resource%5Fa` instead of `resource_a`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiJobBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJobBasic

Property	Type	Description
`allow_failure`	Boolean	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—

Property	Type	Description
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiResourceGroup

Property	Type	Description
`created_at`	String (date-time)	Example: `2021-09-01T08:04:59.650Z`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `production`
`process_mode`	String	Example: `unordered`
`updated_at`	String (date-time)	Example: `2021-09-01T08:04:59.650Z`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

CI runners

> Registering a runner with a registration token is deprecated and disabled by default in GitLab 17.0 and later.

Use this API to register a runner with an instance, verify its authentication, reset its authentication token, and unregister it. GitLab Runner uses these endpoints itself.

To create a runner, or to manage runners that already exist, use the Runners endpoints.

Create a runner

`POST /api/v4/runners`

Creates a runner.

Request body (`application/json`)

Property	Type	Description
`access_level`	String	Access level of the runner Allowed values: `not_protected`, `ref_protected`
`active`	Boolean	If `true`, the runner is allowed to receive new jobs. Deprecated. Use `paused` instead. Mutually exclusive with `paused`
`description`	String	Description of the runner
`info`	Object	Runner’s metadata
`info.architecture`	String	Runner’s architecture
`info.name`	String	Runner’s name
`info.platform`	String	Runner’s platform
`info.revision`	String	Runner’s revision
`info.version`	String	Runner’s version
`locked`	Boolean	If `true`, the runner is locked for the current project
`maintainer_note`	String	Free-form maintenance notes for the runner. Deprecated. Use `maintenance_note` instead. Mutually exclusive with `maintenance_note`
`maintenance_note`	String	Free-form maintenance notes for the runner (1024 characters). Mutually exclusive with `maintainer_note`
`maximum_timeout`	Integer	Maximum time, in seconds, that runners can spend running a job
`paused`	Boolean	

Property	Type	Description
		If `true`, the runner ignores new jobs. Mutually exclusive with `active`
`run_untagged`	Boolean	If `true`, the runner handles untagged jobs
`tag_list`	Array of strings	List of runner tags
`token` Required	String	Registration token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiRunnerRegistrationDetails
`400`	Bad Request	—
`403`	Forbidden	—
`410`	Gone	—

Delete a registered runner

`DELETE /api/v4/runners`

Deletes a specified registered runner.

Parameters

Name	Type	Description
`token` Query, required	String	Authentication token of the runner

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`403`	Forbidden	—

Delete a registered runner manager

`DELETE /api/v4/runners/managers`

Parameters

Name	Type	Description
`token` Query, required	String	Authentication token of the runner
`system_id` Query, required	String	System identifier of the runner

Responses

Code	Description	Schema
`204`	Runner manager was deleted	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Reset a runner authentication token with the current token

`POST /api/v4/runners/reset_authentication_token`

Resets a runner authentication token with the token used to authenticate the request.

Request body (`application/json`)

Property	Type	Description
`token` Required	String	Authentication token of the runner

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`403`	Forbidden	—
`422`	Unprocessable Entity	—

Discover Job Router information

`GET /api/v4/runners/router/discovery`

Discovers Job Router information for a runner. You must provide a valid runner authentication token.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJobRouterDiscoveryInformation
`403`	403 Forbidden	—
`501`	501 Not Implemented	—

Verify authentication for a registered runner

`POST /api/v4/runners/verify`

Verifies authentication for a registered runner.

Request body (`application/json`)

Property	Type	Description
`system_id`	String	System identifier of the runner. Must be provided if `token` starts with `glrt-`
`token` Required	String	Authentication token of the runner

Responses

Code	Description	Schema
`200`	Credentials are valid	APIEntitiesCiRunnerRegistrationDetails
`400`	Bad Request	—
`403`	Forbidden	—
`422`	Runner is orphaned	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJobRouterDiscoveryInformation

Property	Type	Description
`server_url`	String	Example: `wss://kas.example.com`

APIEntitiesCiResetTokenResult

Property	Type	Description
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCiRunnerRegistrationDetails

Property	Type	Description
`id`	String	—
`token`	String	—
`token_expires_at`	String	—

CI triggers

Use this API to trigger pipelines.

Trigger a pipeline with a token

`POST /api/v4/projects/{id}/ref/{ref}/trigger/pipeline`

Triggers a pipeline with a token. With a CI/CD job token, the triggered pipeline is a multi-project pipeline. The job that authenticates the request becomes associated with the upstream pipeline, which is visible on the pipeline graph. If you use a trigger token in a job, the job is not associated with the upstream pipeline.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`ref` Path, required	String	Name of the branch or tag to run the pipeline on Example: `develop`

Request body (`application/json`)

Property	Type	Description
`inputs`	Object	Map of inputs, as key-value pairs, to use when creating the pipeline. Generally available in GitLab 18.1. Feature flag `ci_inputs_for_pipelines` removed
`token` Required	String	Trigger token or CI/CD job token Example: `6d056f63e50fe6f8c5f8f4aa10edb7`
`variables`	Object	Map of pipeline variables, as key-value pairs, for example `{ VAR1: "value1", VAR2: "value2" }` Example: `{"VAR1":"value1", "VAR2":"value2"}`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Trigger a pipeline with a token

`POST /api/v4/projects/{id}/trigger/pipeline`

Triggers a pipeline with a token. With a CI/CD job token, the triggered pipeline is a multi-project pipeline. The job that authenticates the request becomes associated with the upstream pipeline, which is visible on the pipeline graph. If you use a trigger token in a job, the job is not associated with the upstream pipeline.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`

Request body (`application/json`)

Property	Type	Description
`inputs`	Object	Map of inputs, as key-value pairs, to use when creating the pipeline. Generally available in GitLab 18.1. Feature flag `ci_inputs_for_pipelines` removed
`token` Required	String	Trigger token or CI/CD job token Example: `6d056f63e50fe6f8c5f8f4aa10edb7`
`variables`	Object	Map of pipeline variables, as key-value pairs, for example `{ VAR1: "value1", VAR2: "value2" }` Example: `{"VAR1":"value1", "VAR2":"value2"}`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all project trigger tokens

`GET /api/v4/projects/{id}/triggers`

Lists all pipeline trigger tokens for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesTrigger
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create a trigger token

`POST /api/v4/projects/{id}/triggers`

Creates a pipeline trigger token for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`

Request body (`application/json`)

Property	Type	Description
`description` Required	String	Description of the trigger token Example: `my trigger token description`
`expires_at`	String (date-time)	Timestamp of when the pipeline trigger token expires Example: `2024-07-01T00:00:00.000Z`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTrigger
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve trigger token details

`GET /api/v4/projects/{id}/triggers/{trigger_id}`

Retrieves details of a pipeline trigger token for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`trigger_id` Path, required	Integer	ID of the trigger token Example: `10`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTrigger
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Update a pipeline trigger token

`PUT /api/v4/projects/{id}/triggers/{trigger_id}`

Updates a pipeline trigger token for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`trigger_id` Path, required	Integer	ID of the trigger token

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the trigger token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTrigger
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a pipeline trigger token

`DELETE /api/v4/projects/{id}/triggers/{trigger_id}`

Deletes a pipeline trigger token for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`trigger_id` Path, required	Integer	ID of the trigger token Example: `10`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`412`	Precondition Failed	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiPipeline

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`
`tag`	Boolean	Example: `false`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesTrigger

Property	Type	Description
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`description`	String	Example: `test`
`expires_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `10`
`last_used`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`owner`	APIEntitiesUserBasic	—
`token`	String	Example: `6d056f63e50fe6f8c5f8f4aa10edb7`
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

DetailedStatusEntity

Property	Type	Description
`action`	Object	—
`action.button_title`	String	Example: `Cancel this job`
`action.confirmation_message`	String	Example: `Are you sure?`
`action.icon`	String	Example: `cancel`
`action.method`	String	Example: `post`
`action.path`	String	Example: `/namespace1/project1/-/jobs/2/cancel`
`action.title`	String	Example: `Cancel`
`details_path`	String	Example: `/test-group/test-project/-/pipelines/287`
`favicon`	String	Example: `/assets/ci_favicons/favicon_status_success.png`
`group`	String	Example: `success`
`has_details`	Boolean	Example: `true`
`icon`	String	Example: `status_success`
`illustration`	Object	Example: `{"content":"This job depends on upstream jobs that need to succeed in order for this job to be triggered","image":"illustrations/empty-state/empty-job-not-triggered-md.svg","size":"","title":"This job has not been triggered yet"}`
`label`	String	Example: `passed`
`text`	String	Example: `passed`
`tooltip`	String	Example: `passed`

CI variables

Use this API to manage CI/CD variables for a project, a group, or an instance.

Managing project variables requires the Maintainer role for the project. Managing group variables requires the Owner role for the group. Managing instance variables requires administrator access.

List all instance variables

`GET /api/v4/admin/ci/variables`

Lists all instance-level variables. Use the `page` and `per_page` pagination parameters to control the pagination of results.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—

Create a new instance-level variable

`POST /api/v4/admin/ci/variables`

FE version

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the variable
`key` Required	String	The key of the variable. Max 255 characters
`masked`	Boolean	Whether the variable is masked
`protected`	Boolean	Whether the variable is protected
`raw`	Boolean	Whether the variable will be expanded
`value` Required	String	The value of a variable
`variable_type`	String	The type of a variable. Available types are: env_var (default) and file Allowed values: `env_var`, `file`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiVariable
`400`	400 Bad Request	—

Retrieve instance variable details

`GET /api/v4/admin/ci/variables/{key}`

Retrieves details of a specified instance-level variable.

Parameters

Name	Type	Description
`key` Path, required	String	Key of the instance variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Instance Variable Not Found	—

Update an instance-level variable

`PUT /api/v4/admin/ci/variables/{key}`

FE version

Parameters

Name	Type	Description
`key` Path, required	String	The key of a variable

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the variable
`masked`	Boolean	Whether the variable is masked
`protected`	Boolean	Whether the variable is protected
`raw`	Boolean	Whether the variable will be expanded
`value`	String	The value of a variable
`variable_type`	String	The type of a variable. Available types are: env_var (default) and file Allowed values: `env_var` , `file`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Instance Variable Not Found	—

Delete an existing instance-level variable

`DELETE /api/v4/admin/ci/variables/{key}`

FE version

Parameters

Name	Type	Description
`key` Path, required	String	The key of a variable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Instance Variable Not Found	—

List all group variables

`GET /api/v4/groups/{id}/variables`

Lists all variables for a specified group. Use the `page` and `per_page` pagination parameters to control the pagination of results.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Not Found	—

Create a group variable

`POST /api/v4/groups/{id}/variables`

Creates a group variable.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the variable. Limited to 255 characters
`environment_scope`	String	The environment scope of a variable
`key` Required	String	Key of the group variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`
`masked`	Boolean	If `true`, the variable is masked
`masked_and_hidden`	Boolean	If `true`, the variable is masked and hidden
`protected`	Boolean	If `true`, the variable is protected
`raw`	Boolean	If `true`, the variable is treated as a raw string. If `false`, the value is expanded
`value` Required	String	Value of the variable
`variable_type`	String	The type of the variable Allowed values: `env_var`, `file` Default: `env_var`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiVariable
`400`	400 Bad Request	—
`404`	Not Found	—

Retrieve details of a group variable

`GET /api/v4/groups/{id}/variables/{key}`

Retrieves details of a specified group variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`key` Path, required	String	Key of the group variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Group Variable Not Found	—

Update a group variable

`PUT /api/v4/groups/{id}/variables/{key}`

Updates a specified group variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`key` Path, required	String	Key of the group variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the variable. Limited to 255 characters
`environment_scope`	String	The environment scope of a variable
`masked`	Boolean	If `true`, the variable is masked
`protected`	Boolean	If `true`, the variable is protected
`raw`	Boolean	If `true`, the variable is treated as a raw string. If `false`, the value is expanded
`value`	String	Value of the variable
`variable_type`	String	The type of the variable. Default: env_var Allowed values: `env_var`, `file`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	400 Bad Request	—
`404`	Group Variable Not Found	—

Delete a group variable

`DELETE /api/v4/groups/{id}/variables/{key}`

Deletes a specified group variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`key` Path, required	String	Key of the group variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Group Variable Not Found	—

List all project variables

`GET /api/v4/projects/{id}/variables`

Lists all variables for a project. Use the `page` and `per_page` pagination parameters to control the pagination of results.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Not Found	—

Create a variable

`POST /api/v4/projects/{id}/variables`

Creates a variable. If a variable with the same `key` already exists, the variable must have a different `environment_scope`. Otherwise, GitLab returns a message similar to: `VARIABLE_NAME has already been taken`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the variable. Limited to 255 characters
`environment_scope`	String	Environment scope of the variable. Defaults to `*`
`key` Required	String	Key of the project variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`
`masked`	Boolean	If `true`, the variable is masked
`masked_and_hidden`	Boolean	If `true`, the variable is masked and hidden. Introduced in Git Lab 17.4
`protected`	Boolean	If `true`, the variable is protected
`raw`	Boolean	If `true`, the variable is treated as a raw string. If `false`, the value is expanded
`value` Required	String	Value of the variable
`variable_type`	String	The type of the variable Allowed values: `env_var`, `file` Default: `env_var`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiVariable
`400`	400 Bad Request	—
`404`	Not Found	—

Retrieve a single variable

`GET /api/v4/projects/{id}/variables/{key}`

Retrieves details of a specified variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	Key of the project variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`
`filter` Query	Object	Filter results when multiple variables share the same key
`filter[environment_scope]` Query	String	Environment scope of the variable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Variable Not Found	—

Update a variable

`PUT /api/v4/projects/{id}/variables/{key}`

Updates a project variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	Key of the project variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the variable. Limited to 255 characters
`environment_scope`	String	Environment scope of the variable
`filter`	Object	Filter results when multiple variables share the same key
`filter.environment_scope`	String	Environment scope of the variable
`masked`	Boolean	If `true`, the variable is masked
`protected`	Boolean	If `true`, the variable is protected
`raw`	Boolean	If `true`, the variable is treated as a raw string. If `false`, the value is expanded
`value`	String	Value of the variable
`variable_type`	String	The type of the variable. Default: env_var Allowed values: `env_var`, `file`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Variable Not Found	—

Delete a variable

`DELETE /api/v4/projects/{id}/variables/{key}`

Deletes a project variable. If there are multiple variables with the same key, use `filter` to select the correct `environment_scope`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key` Path, required	String	Key of the project variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_`
`filter` Query	Object	Filter results when multiple variables share the same key
`filter[environment_scope]` Query	String	Environment scope of the variable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`404`	Variable Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiVariable

Property	Type	Description
`description`	String	Example: `This variable is being used for ...`
`environment_scope`	String	Example: `*`
`hidden`	Boolean	—
`key`	String	Example: `TEST_VARIABLE_1`
`masked`	Boolean	—
`protected`	Boolean	—
`raw`	Boolean	—
`value`	String	Example: `TEST_1`
`variable_type`	String	Example: `env_var`

Cluster agents

Use this API to interact with the GitLab agent for Kubernetes.

List all agents

`GET /api/v4/projects/{id}/cluster_agents`

Lists all agents registered for the project. You must have the Developer, Maintainer, or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClustersAgent
`400`	Bad Request	—
`404`	Not Found	—

Create an agent

`POST /api/v4/projects/{id}/cluster_agents`

Creates an agent for the project. You must have the Maintainer or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`name` Required	String	The name of the agent

Responses

Code	Description	Schema
`201`	Created	APIEntitiesClustersAgent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve details on an agent

`GET /api/v4/projects/{id}/cluster_agents/{agent_id}`

Retrieves details on a specified agent. You must have the Developer, Maintainer, or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClustersAgent
`400`	Bad Request	—
`404`	Not Found	—

Delete an agent

`DELETE /api/v4/projects/{id}/cluster_agents/{agent_id}`

Deletes an existing agent registration. You must have the Maintainer or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

List all agent tokens

`GET /api/v4/projects/{id}/cluster_agents/{agent_id}/tokens`

Lists all active tokens for an agent. You must have the Developer, Maintainer, or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClustersAgentTokenBasic
`400`	Bad Request	—
`404`	Not Found	—

Create an agent token

`POST /api/v4/projects/{id}/cluster_agents/{agent_id}/tokens`

Creates a token for an agent. You must have the Maintainer or Owner role to use this endpoint. An agent can have only two active tokens at one time.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent

Request body (`application/json`)

Property	Type	Description
`description`	String	The description for the token
`name` Required	String	The name for the token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesClustersAgentTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an agent token

`GET /api/v4/projects/{id}/cluster_agents/{agent_id}/tokens/{token_id}`

Retrieves a specified agent token. You must have the Developer, Maintainer, or Owner role to use this endpoint. Returns a `404` if the agent token has been revoked.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent
`token_id` Path, required	Integer	The ID of the agent token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClustersAgentToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke an agent token

`DELETE /api/v4/projects/{id}/cluster_agents/{agent_id}/tokens/{token_id}`

Revokes an agent token. You must have the Maintainer or Owner role to use this endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`agent_id` Path, required	Integer	The ID of an agent
`token_id` Path, required	Integer	The ID of the agent token

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesClustersAgent

Property	Type	Description
`config_project`	APIEntitiesProjectIdentity	—
`created_at`	String (date-time)	—
`created_by_user_id`	Integer (int64)	Example: `1`
`id`	Integer (int64)	Example: `1`
`name`	String	—

APIEntitiesClustersAgentToken

Property	Type	Description
`agent_id`	Integer (int64)	Example: `1`
`created_at`	String (date-time)	—
`created_by_user_id`	Integer (int64)	Example: `1`
`description`	String	—
`id`	Integer (int64)	Example: `1`
`last_used_at`	String (date-time)	—
`name`	String	—
`status`	String	—

APIEntitiesClustersAgentTokenBasic

Property	Type	Description
`agent_id`	Integer (int64)	Example: `1`
`created_at`	String (date-time)	—
`created_by_user_id`	Integer (int64)	Example: `1`
`description`	String	—
`id`	Integer (int64)	Example: `1`
`name`	String	—
`status`	String	—

APIEntitiesClustersAgentTokenWithToken

Property	Type	Description
`agent_id`	Integer (int64)	Example: `1`
`created_at`	String (date-time)	—
`created_by_user_id`	Integer (int64)	Example: `1`
`description`	String	—
`id`	Integer (int64)	Example: `1`
`last_used_at`	String (date-time)	—
`name`	String	—
`status`	String	—
`token`	String	—

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

Clusters

> These certificate-based cluster endpoints are deprecated.

Use this API to manage and discover the certificate-based Kubernetes clusters of an instance, a group, or a project. Connecting a cluster to a group or an instance lets you use the same cluster across multiple projects.

Instance endpoints require administrator access. Group and project endpoints require the Maintainer or Owner role.

List all instance clusters

`GET /api/v4/admin/clusters`

Lists all instance clusters for the instance.

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCluster
`403`	Forbidden	—

Create an instance cluster

`POST /api/v4/admin/clusters/add`

Creates an instance cluster by adding an existing Kubernetes cluster.

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled Default: `true`
`environment_scope`	String	Associated environment to the cluster Default: `*`
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster Default: `true`
`management_project_id`	Integer	ID of the management project for the cluster
`name` Required	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes` Required	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url` Required	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.authorization_type`	String	Cluster authorization type Allowed values: `unknown_authorization`, `rbac`, `abac` Default: `rbac`
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)

Property	Type	Description
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. Allowed only when `managed` is `false`
`platform_kubernetes_attributes.token` Required	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCluster
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a single instance cluster

`GET /api/v4/admin/clusters/{cluster_id}`

Retrieves a specified instance cluster.

Parameters

Name	Type	Description
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCluster
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Update an instance cluster

`PUT /api/v4/admin/clusters/{cluster_id}`

Updates an existing instance cluster.

Parameters

Name	Type	Description
`cluster_id` Path, required	Integer	ID of the cluster

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled
`environment_scope`	String	Associated environment to the cluster
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster
`management_project_id`	Integer	ID of the management project for the cluster
`name`	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes`	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url`	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. Allowed only when `managed` is `false`
`platform_kubernetes_attributes.token`	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCluster
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Delete instance cluster

`DELETE /api/v4/admin/clusters/{cluster_id}`

Deletes an existing instance cluster. Does not remove existing resources in the connected Kubernetes cluster.

Parameters

Name	Type	Description
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCluster
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

List all certificate-based clusters

`GET /api/v4/discover-cert-based-clusters`

Lists all certificate-based clusters associated with a project. This feature was introduced in GitLab 17.9.

Parameters

Name	Type	Description
`group_id` Query, required	Integer	ID of the group to find all certificate-based clusters in the hierarchy

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscoveredClusters
`400`	Bad Request	—
`403`	Forbidden	—

List all group clusters

`GET /api/v4/groups/{id}/clusters`

Lists all group clusters for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCluster
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Create a group cluster

`POST /api/v4/groups/{id}/clusters/user`

Creates a group cluster for a specified group by adding an existing Kubernetes cluster.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled Default: `true`
`environment_scope`	String	Associated environment to the cluster. Premium and Ultimate only Default: `*`
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster Default: `true`
`management_project_id`	Integer	ID of the management project for the cluster
`name` Required	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes` Required	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url` Required	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.authorization_type`	String	Cluster authorization type Allowed values: `unknown_authorization`, `rbac`, `abac` Default: `rbac`
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)

Property	Type	Description
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. Allowed only when `managed` is `false`
`platform_kubernetes_attributes.token` Required	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesClusterGroup
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a group cluster

`GET /api/v4/groups/{id}/clusters/{cluster_id}`

Retrieves a specified group cluster.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterGroup
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Update a group cluster

`PUT /api/v4/groups/{id}/clusters/{cluster_id}`

Updates a specified group cluster.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`cluster_id` Path, required	Integer	ID of the cluster

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled
`environment_scope`	String	Associated environment to the cluster. Premium and Ultimate only
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster
`management_project_id`	Integer	ID of the management project for the cluster
`name`	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes`	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url`	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. Allowed only when `managed` is `false`
`platform_kubernetes_attributes.token`	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterGroup
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Delete a group cluster

`DELETE /api/v4/groups/{id}/clusters/{cluster_id}`

Deletes a specified group cluster. Does not remove existing resources in the connected Kubernetes cluster.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterGroup
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

List all clusters in a project

`GET /api/v4/projects/{id}/clusters`

Lists all clusters in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCluster
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Add a cluster to a project

`POST /api/v4/projects/{id}/clusters/user`

Adds a cluster to a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled Default: `true`
`environment_scope`	String	Associated environment to the cluster. Premium and Ultimate only Default: `*`
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster Default: `true`
`management_project_id`	Integer	ID of the management project for the cluster
`name` Required	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes` Required	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url` Required	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.authorization_type`	String	Cluster authorization type Allowed values: `unknown_authorization`, `rbac`, `abac` Default: `rbac`
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)

Property	Type	Description
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. If `managed` and `namespace_per_environment` are `true`, the environment slug is appended
`platform_kubernetes_attributes.token` Required	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesClusterProject
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a cluster from a project

`GET /api/v4/projects/{id}/clusters/{cluster_id}`

Retrieves a specified cluster in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterProject
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Update a cluster in a project

`PUT /api/v4/projects/{id}/clusters/{cluster_id}`

Updates a cluster in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`cluster_id` Path, required	Integer	ID of the cluster

Request body (`application/json`)

Property	Type	Description
`domain`	String	Base domain of the cluster
`enabled`	Boolean	If `true`, the cluster is active and GitLab's connection to the Kubernetes cluster is enabled
`environment_scope`	String	Associated environment to the cluster. Premium and Ultimate only
`managed`	Boolean	If `true`, GitLab manages namespaces and service accounts for this cluster
`management_project_id`	Integer	ID of the management project for the cluster
`name`	String	Name of the cluster
`namespace_per_environment`	Boolean	If `true`, deploys each environment to a separate Kubernetes namespace Default: `true`
`platform_kubernetes_attributes`	Object	Platform Kubernetes data
`platform_kubernetes_attributes.api_url`	String	URL to access the Kubernetes API
`platform_kubernetes_attributes.ca_cert`	String	TLS certificate (needed if API is using a self-signed TLS certificate)
`platform_kubernetes_attributes.namespace`	String	Kubernetes namespace that environments deploy to. If `managed` and `namespace_per_environment` are `true`, the environment slug is appended
`platform_kubernetes_attributes.token`	String	Token to authenticate against Kubernetes

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterProject
`400`	Validation error	—
`403`	Forbidden	—
`404`	Not found	—

Delete cluster from a project

`DELETE /api/v4/projects/{id}/clusters/{cluster_id}`

Deletes a specified cluster from a project. Does not remove existing resources in the connected Kubernetes cluster.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`cluster_id` Path, required	Integer	ID of the cluster

Responses

Code	Description	Schema
`200`	OK	APIEntitiesClusterProject
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicGroupDetails

Property	Type	Description
`id`	Integer (int64)	—
`name`	String	Example: `Diaspora`
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCluster

Property	Type	Description
`cluster_type`	String	—
`created_at`	String	—
`domain`	String	—
`enabled`	String	—
`environment_scope`	String	—
`id`	String	—
`managed`	String	—
`management_project`	APIEntitiesProjectIdentity	—
`name`	String	—
`namespace_per_environment`	String	—
`platform_kubernetes`	APIEntitiesPlatformKubernetes	—
`platform_type`	String	—
`provider_gcp`	APIEntitiesProviderGcp	—
`provider_type`	String	—
`user`	APIEntitiesUserBasic	—

APIEntitiesClusterGroup

Property	Type	Description
`cluster_type`	String	—
`created_at`	String	—
`domain`	String	—
`enabled`	String	—
`environment_scope`	String	—
`group`	APIEntitiesBasicGroupDetails	—
`id`	String	—
`managed`	String	—
`management_project`	APIEntitiesProjectIdentity	—
`name`	String	—
`namespace_per_environment`	String	—
`platform_kubernetes`	APIEntitiesPlatformKubernetes	—
`platform_type`	String	—
`provider_gcp`	APIEntitiesProviderGcp	—
`provider_type`	String	—
`user`	APIEntitiesUserBasic	—

APIEntitiesClusterProject

Property	Type	Description
`cluster_type`	String	—
`created_at`	String	—
`domain`	String	—
`enabled`	String	—
`environment_scope`	String	—
`id`	String	—
`managed`	String	—
`management_project`	APIEntitiesProjectIdentity	—
`name`	String	—
`namespace_per_environment`	String	—
`platform_kubernetes`	APIEntitiesPlatformKubernetes	—
`platform_type`	String	—
`project`	APIEntitiesBasicProjectDetails	—
`provider_gcp`	APIEntitiesProviderGcp	—
`provider_type`	String	—
`user`	APIEntitiesUserBasic	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiscoveredClusters

Property	Type	Description
`groups`	Object	—
`projects`	Object	—

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesPlatformKubernetes

Property	Type	Description
`api_url`	String	—
`authorization_type`	String	—
`ca_cert`	String	—
`namespace`	String	—

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesProviderGcp

Property	Type	Description
`cluster_id`	String	—
`endpoint`	String	—
`gcp_project_id`	String	—
`machine_type`	String	—
`num_nodes`	String	—
`status_name`	String	—
`zone`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Codeowners

Operations related to codeowners.

List CODEOWNERS entries that match a file path

`GET /api/v4/code-owners-for-file`

Parameters

Name	Type	Description
`project_path` Query, required	String	—
`file_path` Query, required	String	—
`ref` Query	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Validate CODEOWNERS file by project full path

`GET /api/v4/validate-code-owner-file`

Parameters

Name	Type	Description
`project_path` Query, required	String	—
`ref` Query	String	—
`file_path` Query	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Commit statuses

Use this API to list the statuses recorded against a commit, and to add or update the status that an external CI/CD job reports for a commit.

List all commit statuses

`GET /api/v4/projects/{id}/repository/commits/{sha}/statuses`

Lists all commit statuses for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA Example: `18f3e63d05582537db6d183d9d557be09e1f90c8`
`ref` Query	String	Name of the branch or tag. Defaults to the default branch Example: `develop`
`stage` Query	String	Filter statuses by build stage. For example, `test` Example: `test`
`name` Query	String	Filter statuses by job name Example: `bundler:audit`
`pipeline_id` Query	Integer	Filter statuses by pipeline ID Example: `1234`
`all` Query	Boolean	If `true`, includes all statuses instead of only the latest
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `pipeline_id` Default: `id`
`sort` Query	String	Sort statuses in ascending or descending order. Defaults to `asc` Allowed values: `asc`, `desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCommit Status
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create or update a commit pipeline status

`POST /api/v4/projects/{id}/statuses/{sha}`

Creates or updates the status of a commit represented by a job in an `external` stage. If the commit is associated with a merge request, target the commit in the merge request source branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA Example: `18f3e63d05582537db6d183d9d557be09e1f90c8`

Request body (`application/json`)

Property	Type	Description
`context`	String	String label to differentiate this status from the status of other systems Example: `coverage`
`coverage`	Number	Total code coverage Example: `100`
`description`	String	Short description of the status. Must be 255 characters or fewer
`name`	String	String label to differentiate this status from the status of other systems Example: `coverage`
`pipeline_id`	Integer	ID of the pipeline to set the status for. Use when multiple pipelines exist for the same commit SHA
`ref`	String	Branch or tag that the status refers to. Must be 255 characters or fewer Example: `develop`
`state` Required	String	State of the status Allowed values: `pending`, `running`, `success`, `failed`, `canceled`, `skipped` Minimum length: `1`
`target_url`	String	Target URL to associate with this status. Must be 255 characters or fewer Example: `https://gitlab.example.com/janedoe/gitlab-foss/builds/91`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Another update to this commit status is in progress	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCommitStatus

Property	Type	Description
`allow_failure`	Boolean	Example: `false`
`author`	APIEntitiesUserBasic	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`description`	String	—
`finished_at`	String (date-time)	Example: `2016-01-21T08:40:25.832Z`
`id`	Integer (int64)	Example: `93`
`name`	String	Example: `default`
`pipeline_id`	Integer (int64)	Example: `101`
`ref`	String	Example: `develop`
`sha`	String	Example: `18f3e63d05582537db6d183d9d557be09e1f90c8`
`started_at`	String (date-time)	Example: `2016-01-20T08:40:25.832Z`
`status`	String	Example: `success`
`target_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-foss/builds/91`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Commits

Use this API to manage Git commits.

List all repository commits

`GET /api/v4/projects/{id}/repository/commits`

Lists all commits for a specified project repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`ref_name` Query	String	Name of a repository branch, tag, or revision range. Uses the default branch if not given Example: `v1.1.0`
`since` Query	String (date-time)	Only commits after or on this date are returned Example: `2021-09-20T11:50:22.001Z`
`until` Query	String (date-time)	Only commits before or on this date are returned Example: `2021-09-20T11:50:22.001Z`
`path` Query	String	File path relative to the repository Example: `README.md`
`follow` Query	Boolean	If `true`, follows file renames when filtering commits by `path` and returns commits for the file even if it was renamed. If `false`, returns only commits where the file existed at its current path. Used only when `path` specifies a single file
`author` Query	String	Filter commits by commit author Example: `John Smith`
`all` Query	Boolean	If `true`, retrieves every commit from the repository and ignores the `ref_name` parameter
`with_stats` Query	Boolean	If `true`, includes statistics for each commit in the response

Name	Type	Description
`first_parent` Query	Boolean	If `true`, follows only the first parent commit upon seeing a merge commit
`order` Query	String	Order in which to list commits, either reverse chronological or topological Allowed values: `default`, `topo` Default: `default`
`trailers` Query	Boolean	If `true`, parses and includes Git trailers for every commit Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`pagination` Query	String	Specify the pagination method Allowed values: `legacy`, `keyset` Default: `legacy`
`page_token` Query	String	Record from which to start the keyset pagination

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCommit
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a commit

`POST /api/v4/projects/{id}/repository/commits`

This feature was introduced in GitLab 8.13

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The commit content to be created (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitDetail
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a commit

`GET /api/v4/projects/{id}/repository/commits/{sha}`

Retrieves a specified commit identified by the commit hash or name of a branch or tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA or name of a repository branch or tag
`stats` Query	Boolean	If `true`, includes commit statistics in the response Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitDetail
`400`	Bad Request	—
`404`	Not found	—

Cherry-pick a commit

`POST /api/v4/projects/{id}/repository/commits/{sha}/cherry_pick`

Cherry-picks a commit to a specified branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA, or name of a repository branch or tag, to cherry-pick

Request body (`application/json`)

Property	Type	Description
`branch` Required	String	Name of the branch Example: `master`
`dry_run`	Boolean	If `true`, does not commit any changes Default: `false`
`message`	String	Custom commit message to use for the new commit Example: `Initial commit`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommit
`400`	Bad request	—
`404`	Not found	—

List all commit comments

`GET /api/v4/projects/{id}/repository/commits/{sha}/comments`

Lists all the comments of a commit in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`sha` Path, required	String	Commit SHA or name of a repository branch or tag

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCommit Note
`400`	Bad Request	—
`404`	Not found	—

Create a comment on a commit

`POST /api/v4/projects/{id}/repository/commits/{sha}/comments`

Creates a comment on a commit. To comment on a specific line in a file, specify the full commit SHA, `path`, `line`, and set `line_type` to `new`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA or name of a repository branch or tag

Request body (`application/json`)

Property	Type	Description
`line` Required	Integer	Line number where the comment should be placed Example: `11`
`line_type` Required	String	Line type Allowed values: `new`, `old` Default: `new` Minimum length: `1`
`note` Required	String	Text of the comment Example: `Nice code!`
`path`	String	File path relative to the repository Example: `doc/update/5.4-to-6.0.md`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitNote
`400`	Bad request	—
`404`	Not found	—

Retrieve a commit diff

`GET /api/v4/projects/{id}/repository/commits/{sha}/diff`

Retrieves the diff of a commit in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA or name of a repository branch or tag
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`unidiff` Query	Boolean	If `true`, presents diffs in the unified diff format Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDiff
`400`	Bad Request	—
`404`	Not found	—

List all merge requests associated with a commit

`GET /api/v4/projects/{id}/repository/commits/{sha}/merge_requests`

Lists all merge requests associated with a specified commit.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA or name of a repository branch or tag
`state` Query	String	Filter merge requests by state: `opened`, `closed`, `locked`, or `merged`. If omitted, merge requests of all states are returned. Introduced in GitLab 18.2 Example: `merged`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`404`	Not found	—

List all references a commit is pushed to

`GET /api/v4/projects/{id}/repository/commits/{sha}/refs`

Lists all references (from branches or tags) a commit is pushed to. The pagination parameters `page` and `per_page` can be used to restrict the list of references.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA
`type` Query	String	Type of references to return for the commit Allowed values: `branch`, `tag`, `all` Default: `all`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBasicRef
`400`	Bad Request	—
`404`	Not found	—

Revert a commit

`POST /api/v4/projects/{id}/repository/commits/{sha}/revert`

Reverts a commit in a specified branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA to revert

Request body (`application/json`)

Property	Type	Description
`branch` Required	String	Name of the target branch Example: `master`
`dry_run`	Boolean	If `true` , does not commit any changes Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommit
`400`	Bad request	—
`404`	Not found	—

Retrieve a commit sequence

`GET /api/v4/projects/{id}/repository/commits/{sha}/sequence`

Retrieves the commit sequence for a specified commit.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA
`first_parent` Query	Boolean	If `true`, follows only the first parent commit upon seeing a merge commit Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitSequence
`400`	Bad Request	—
`404`	Not found	—

Retrieve a commit signature

`GET /api/v4/projects/{id}/repository/commits/{sha}/signature`

Retrieves the signature from a commit, if it is signed. For unsigned commits, it results in a 404 response.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sha` Path, required	String	Commit SHA or name of a repository branch or tag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitSignature
`400`	Bad Request	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicRef

Property	Type	Description
`name`	String	Example: `v1.1.0`
`type`	String	Example: `tag`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCommitDetail

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by": ["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`last_pipeline`	Object	—
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`project_id`	Integer (int64)	Example: `1`
`short_id`	String	Example: `2695effb`
`stats`	APIEntitiesCommitStats	—
`status`	String	Example: `success`
`title`	String	Example: `Initial commit`

Property	Type	Description
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCommitNote

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2016-01-19T09:44:55.600Z`
`line`	Integer	Example: `11`
`line_type`	String	Example: `new`
`note`	String	Example: `this doc is really nice`
`path`	String	Example: `README.md`

APIEntitiesCommitSequence

Property	Type	Description
`count`	Integer	Example: `1`

APIEntitiesCommitSignature

Property	Type	Description
`commit_source`	String	Example: `gitaly`
`signature`	Object	—
`signature_type`	String	Example: `PGP`

APIEntitiesCommitStats

Property	Type	Description
`additions`	Integer	Example: `1`
`deletions`	Integer	Example: `0`
`total`	Integer	Example: `1`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiff

Property	Type	Description
`a_mode`	String	Example: `100755`
`b_mode`	String	Example: `100644`
`collapsed`	Boolean	—
`deleted_file`	Boolean	—
`diff`	String	Example: `@@ -71,6 +71,8 @@\n...`
`generated_file`	Boolean	—
`new_file`	Boolean	—
`new_path`	String	Example: `doc/update/5.4-to-6.0.md`
`old_path`	String	Example: `doc/update/5.4-to-6.0.md`
`renamed_file`	Boolean	—
`too_large`	Boolean	—

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Container registry

Use this API to manage the GitLab container registry: list the registry repositories in a project or a group, retrieve or delete individual repositories and their tags, and receive registry events.

To authenticate with these endpoints from a CI/CD job, pass the `$CI_JOB_TOKEN` variable as the `JOB-TOKEN` header. The job token only has access to the container registry of the project that created the pipeline.

Receives notifications from the container registry when an operation occurs

`POST /api/v4/container_registry_event/events`

This feature was introduced in GitLab 12.10

Responses

Code	Description	Schema
`200`	Success	—
`400`	Bad Request	—
`401`	Invalid Token	—

List all registry repositories for a group

`GET /api/v4/groups/{id}/registry/repositories`

Lists all registry repositories for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesContainerRegistryRepository
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Group Not Found	—

List all registry repositories for a project

`GET /api/v4/projects/{id}/registry/repositories`

Lists all registry repositories for a specified project. Responses are paginated and return 20 results by default.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`tags` Query	Boolean	If `true`, includes an array of `tags` in the response Default: `false`
`tags_count` Query	Boolean	If `true`, includes `tags_count` in the response Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesContainerRegistryRepository
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Delete registry repository

`DELETE /api/v4/projects/{id}/registry/repositories/{repository_id}`

Deletes a specified repository in the registry. This operation is executed asynchronously and might take some time to execute.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_id` Path, required	Integer	ID of the container registry repository

Responses

Code	Description	Schema
`204`	Success	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all registry repository tags for a project

`GET /api/v4/projects/{id}/registry/repositories/{repository_id}/tags`

Lists all tags for a specified registry repository. Responses are paginated and return 20 results by default.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_id` Path, required	Integer	ID of the container registry repository
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesContainerRegistryTag
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—
`405`	Method Not Allowed	—

Delete multiple registry repository tags

`DELETE /api/v4/projects/{id}/registry/repositories/{repository_id}/tags`

Deletes multiple registry repository tags based on the specified criteria.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_id` Path, required	Integer	ID of the container registry repository
`name_regex_delete` Query	String	RE2 regular expression of the tag names to delete. Specify `.*` to delete all tags. At least one of `name_regex`, `name_regex_delete` must be provided
`name_regex` Query	String	RE2 regular expression of the tag names to delete. Specify `.*` to delete all tags. Deprecated. Use `name_regex_delete` instead. At least one of `name_regex`, `name_regex_delete` must be provided
`name_regex_keep` Query	String	RE2 regular expression of the tag names to keep. Overrides any matches from `name_regex_delete`. Setting to `.*` results in a no-op
`keep_n` Query	Integer	Number of most recent tags with a matching name to keep
`older_than` Query	String	Delete tags older than this age, such as `1h`, `1d`, or `1month`

Responses

Code	Description	Schema
`204`	Success	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve details of a registry repository tag

`GET /api/v4/projects/{id}/registry/repositories/{repository_id}/tags/{tag_name}`

Retrieves details of a specified registry repository tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_id` Path, required	Integer	ID of the container registry repository
`tag_name` Path, required	String	Name of the tag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesContainerRegistryTagDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Delete a registry repository tag

`DELETE /api/v4/projects/{id}/registry/repositories/{repository_id}/tags/{tag_name}`

Deletes a specified container registry repository tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_id` Path, required	Integer	ID of the container registry repository
`tag_name` Path, required	String	Name of the tag

Responses

Code	Description	Schema
`204`	Success	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve details of a container registry repository

`GET /api/v4/registry/repositories/{id}`

Retrieves details of a specified container registry repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID of the repository
`tags` Query	Boolean	If `true`, includes an array of `tags` in the response Default: `false`
`tags_count` Query	Boolean	If `true`, includes `tags_count` in the response Default: `false`
`size` Query	Boolean	If `true`, response includes the deduplicated size of all images in the repository. Only available on GitLab.com for repositories created after `2021-11-04` Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesContainerRegistryRepository
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Repository Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesContainerRegistryRepository

Property	Type	Description
`cleanup_policy_started_at`	String (date-time)	Example: `2020-08-17T03:12:35.489Z`
`created_at`	String (date-time)	Example: `2019-01-10T13:39:08.229Z`
`delete_api_path`	String	Example: `delete/api/path`
`id`	Integer (int64)	Example: `1`
`location`	String	Example: `gitlab.example.com/group/project/releases`
`name`	String	Example: `releases`
`path`	String	Example: `group/project/releases`
`project_id`	Integer (int64)	Example: `9`
`size`	Integer	Example: `12345`
`status`	String	Example: `delete_scheduled`
`tags`	APIEntitiesContainerRegistryTag	—
`tags_count`	Integer	Example: `3`

APIEntitiesContainerRegistryTag

Property	Type	Description
`location`	String	Example: `registry.dev/namespace1/project1/test_image_1:latest`
`name`	String	Example: `latest`
`path`	String	Example: `namespace1/project1/test_image_1:latest`

APIEntitiesContainerRegistryTagDetails

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-01-10T13:39:08.229Z`
`digest`	String	Example: `shadigest`
`location`	String	Example: `registry.dev/namespace1/project1/test_image_1:latest`
`name`	String	Example: `latest`
`path`	String	Example: `namespace1/project1/test_image_1:latest`
`revision`	String	Example: `tagrevision`
`short_revision`	String	Example: `shortrevison`
`total_size`	Integer	Example: `3`

Custom attributes

Use this API to manage the custom attributes set on users, groups, and projects.

These endpoints require administrator access to the instance.

List all custom attributes for a group

`GET /api/v4/groups/{id}/custom_attributes`

Lists all custom attributes for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a custom attribute for a group

`GET /api/v4/groups/{id}/custom_attributes/{key}`

Retrieves a specified custom attribute for a group.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Creates or updates a custom attribute for a group

`PUT /api/v4/groups/{id}/custom_attributes/{key}`

Creates or updates a custom attribute for a specified group. If the attribute already exists, it is updated, otherwise a new attribute is created.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`value` Required	String	The value of the custom attribute

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Delete a custom attribute for a group

`DELETE /api/v4/groups/{id}/custom_attributes/{key}`

Deletes a specified custom attribute for a group.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all custom attributes for a project

`GET /api/v4/projects/{id}/custom_attributes`

Lists all custom attributes for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a custom attribute for a project

`GET /api/v4/projects/{id}/custom_attributes/{key}`

Retrieves a specified custom attribute for a project.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Creates or updates a custom attribute for a project

`PUT /api/v4/projects/{id}/custom_attributes/{key}`

Creates or updates a custom attribute for a specified project. If the attribute already exists, it is updated, otherwise a new attribute is created.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`value` Required	String	The value of the custom attribute

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Delete a custom attribute for a project

`DELETE /api/v4/projects/{id}/custom_attributes/{key}`

Deletes a specified custom attribute for a project.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all custom attributes for a user

`GET /api/v4/users/{id}/custom_attributes`

Lists all custom attributes for a specified user.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a custom attribute for a user

`GET /api/v4/users/{id}/custom_attributes/{key}`

Retrieves a specified custom attribute for a user.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Creates or updates a custom attribute for a user

`PUT /api/v4/users/{id}/custom_attributes/{key}`

Creates or updates a custom attribute for a specified user. If the attribute already exists, it is updated, otherwise a new attribute is created.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the user

Request body (`application/json`)

Property	Type	Description
`value` Required	String	The value of the custom attribute

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCustomAttribute
`400`	Bad Request	—
`404`	Not Found	—

Delete a custom attribute for a user

`DELETE /api/v4/users/{id}/custom_attributes/{key}`

Deletes a specified custom attribute for a user.

Parameters

Name	Type	Description
`key` Path, required	String	The key of the custom attribute
`id` Path, required	String or integer	The ID or URL-encoded path of the user

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

DORA metrics

Use this API to retrieve DORA metrics for groups and projects. Additional metrics are available through the GraphQL API.

These endpoints require the Reporter, Developer, Maintainer, or Owner role.

Get group DORA metrics

`GET /api/v4/groups/{id}/dora/metrics`

Retrieves one DORA metric for a group hierarchy over a selected period.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `12`
`metric` Query, required	String	DORA metric to retrieve Allowed values: `deployment_frequency`, `lead_time_for_changes`, `time_to_restore_service`, `change_failure_rate` Minimum length: `1` Example: `deployment_frequency`
`start_date` Query	String (date)	First day of the period Example: `2026-03-01`
`end_date` Query	String (date)	Last day of the period Example: `2026-03-31`
`interval` Query	String	Size of the aggregation bucket Allowed values: `all`, `monthly`, `daily` Default: `daily` Example: `daily`
`environment_tiers` Query	Array of strings	Environment tiers to include Example: `["production"]`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDoraMetric
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Get project DORA metrics

`GET /api/v4/projects/{id}/dora/metrics`

Retrieves one DORA metric for a project over a selected period.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`metric` Query, required	String	DORA metric to retrieve Allowed values: `deployment_frequency`, `lead_time_for_changes`, `time_to_restore_service`, `change_failure_rate` Minimum length: `1` Example: `deployment_frequency`
`start_date` Query	String (date)	First day of the period Example: `2026-03-01`
`end_date` Query	String (date)	Last day of the period Example: `2026-03-31`
`interval` Query	String	Size of the aggregation bucket Allowed values: `all`, `monthly`, `daily` Default: `daily` Example: `daily`
`environment_tiers` Query	Array of strings	Environment tiers to include Example: `["production"]`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDoraMetric
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesDoraMetric

Property	Type	Description
`date`	String (date)	Example: `2026-03-04`
`value`	Number	Example: `1.5`

Database dictionary

Use this API to list the tables in a GitLab database and retrieve the database dictionary entry for a single table.

Retrieving the entry for a single table requires administrator access to the instance.

Retrieve dictionary details

`GET /api/v4/admin/databases/{database_name}/dictionary/tables/{table_name}`

Parameters

Name	Type	Description
`database_name` Path, required	String	Name of the database Allowed values: `main`, `ci` Minimum length: `1`
`table_name` Path, required	String	Name of the table

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDictionaryTable
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—

List dictionary tables

`GET /api/v4/databases/{database_name}/dictionary/tables`

Returns database dictionary tables filtered by database and optional table size

Parameters

Name	Type	Description
`database_name` Path, required	String	Name of the database Allowed values: `main`, `ci`, `sec` Minimum length: `1`
`table_size` Query	String	Filter by table size classification Allowed values: `small`, `medium`, `large`, `over_limit`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDictionaryTable
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesDictionaryTable

Property	Type	Description
`feature_categories`	Array of strings	Example: `["database"]`
`table_name`	String	Example: `users`
`table_size`	String	Example: `small`

Dependency proxy

Use this API to purge the cache of a group's dependency proxy.

Purge the dependency proxy for a group

`DELETE /api/v4/groups/{id}/dependency_proxy/cache`

Purges the dependency proxy for a specified group and schedules the cached manifests and blobs for deletion. This endpoint requires the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Deploy resources

Use this API to manage deploy keys, deploy tokens, and code deployments to GitLab environments.

List all deploy keys

`GET /api/v4/deploy_keys`

Lists all deploy keys for the instance.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`public` Query	Boolean	If `true`, returns only deploy keys that are public Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeployKey
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—

Create a deploy key

`POST /api/v4/deploy_keys`

Creates a deploy key for the GitLab instance. Requires administrator access.

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date-time)	Date when the deploy key expires. If omitted, the deploy key does not expire
`key` Required	String	Public key value of the deploy key
`title` Required	String	Title of the deploy key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeployKey
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—

List all deploy tokens

`GET /api/v4/deploy_tokens`

Lists all deploy tokens for the instance.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`active` Query	Boolean	If `true`, returns only active deploy tokens

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeploy Token
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—

List all group deploy tokens

`GET /api/v4/groups/{id}/deploy_tokens`

Lists all group deploy tokens.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`active` Query	Boolean	If `true`, returns only active deploy tokens

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeploy Token
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create a group deploy token

`POST /api/v4/groups/{id}/deploy_tokens`

Creates a group deploy token.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date-time)	Date when the deploy token expires. If omitted, the deploy token does not expire
`name` Required	String	Name of the deploy token
`scopes` Required	Array of strings	Scopes of the deploy token. Must include at least one scope
`username`	String	Username for the deploy token. Defaults to `gitlab+deploy-token-{n}`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeployTokenWithToken
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a group deploy token

`GET /api/v4/groups/{id}/deploy_tokens/{token_id}`

Get a single group’s deploy token by ID. This feature was introduced in GitLab 14.9.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the deploy token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeployToken
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete a group deploy token

`DELETE /api/v4/groups/{id}/deploy_tokens/{token_id}`

Deletes a group deploy token.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the deploy token

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all deploy keys for project

`GET /api/v4/projects/{id}/deploy_keys`

Lists all deploy keys for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeployKeysProject
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Add a deploy key for a project

`POST /api/v4/projects/{id}/deploy_keys`

Adds a deploy key for a specified project. If the deploy key already exists in another project, it is joined to the current project only if the original one is accessible by the same user.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`can_push`	Boolean	If `true` , the deploy key can push to the project's repository
`expires_at`	String (date-time)	Date when the deploy key expires. If omitted, the deploy key does not expire
`key` Required	String	Public key value of the deploy key
`title` Required	String	Title of the deploy key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeployKeysProject
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a deploy key

`GET /api/v4/projects/{id}/deploy_keys/{key_id}`

Retrieves a specified deploy key.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key_id` Path, required	Integer	ID of the deploy key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeployKeysProject
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a deploy key

`PUT /api/v4/projects/{id}/deploy_keys/{key_id}`

Updates a deploy key for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key_id` Path, required	Integer	ID of the deploy key

Request body (`application/json`)

Property	Type	Description
`can_push`	Boolean	If `true` , the deploy key can push to the project's repository. At least one of `title` , `can_push` must be provided
`title`	String	Title of the deploy key. At least one of `title` , `can_push` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeployKey
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete deploy key

`DELETE /api/v4/projects/{id}/deploy_keys/{key_id}`

FE version

Removes a deploy key from the project. If the deploy key is used only for this project, it’s deleted from the system.

Parameters

Name	Type	Description
`key_id` Path, required	Integer	The ID of the deploy key
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Enable a deploy key

`POST /api/v4/projects/{id}/deploy_keys/{key_id}/enable`

Enables a deploy key for a project so this can be used. Returns the enabled key, with a status code 201 when successful.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key_id` Path, required	Integer	ID of the deploy key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeployKey
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all project deploy tokens

`GET /api/v4/projects/{id}/deploy_tokens`

Lists all project deploy tokens.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`active` Query	Boolean	If `true`, returns only active deploy tokens

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeployToken
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create a project deploy token

`POST /api/v4/projects/{id}/deploy_tokens`

Creates a project deploy token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date-time)	Date when the deploy token expires. If omitted, the deploy token does not expire
`name` Required	String	Name of the deploy token
`scopes` Required	Array of strings	Scopes of the deploy token. Must include at least one scope
`username`	String	Username for the deploy token. Defaults to `gitlab+deploy-token-{n}`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeployTokenWithToken
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a project deploy token

`GET /api/v4/projects/{id}/deploy_tokens/{token_id}`

Retrieves a project deploy token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`token_id` Path, required	Integer	ID of the deploy token

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeployToken
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete a project deploy token

`DELETE /api/v4/projects/{id}/deploy_tokens/{token_id}`

Deletes a project deploy token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`token_id` Path, required	Integer	ID of the deploy token

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all project deployments

`GET /api/v4/projects/{id}/deployments`

Lists all deployments in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Return deployments ordered by either one of `id`, `iid`, `created_at`, `updated_at` or `ref` fields. Default is `id` Allowed values: `id`, `iid`, `created_at`, `updated_at`, `finished_at` Default: `id`
`sort` Query	String	Return deployments sorted in `asc` or `desc` order. Default is `asc` Allowed values: `asc`, `desc` Default: `asc`
`updated_after` Query	String (date-time)	Return deployments updated after the specified date. Expected in ISO 8601 format (`2019-03-15T08:00:00Z`)
`updated_before` Query	String (date-time)	Return deployments updated before the specified date. Expected in ISO 8601 format (`2019-03-15T08:00:00Z`)
`finished_after` Query	String (date-time)	Return deployments finished after the specified date.

Name	Type	Description
		Expected in ISO 8601 format (`2019-03-15T08:00:00Z`)
`finished_before` Query	String (date-time)	Return deployments finished before the specified date. Expected in ISO 8601 format (`2019-03-15T08:00:00Z`)
`environment` Query	String	The name of the environment to filter deployments by
`status` Query	String	The status to filter deployments by. One of `created` , `running` , `success` , `failed` , `canceled` , or `blocked` Allowed values: `created` , `running` , `success` , `failed` , `canceled` , `skipped` , `blocked`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeployment
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a deployment

`POST /api/v4/projects/{id}/deployments`

Creates a deployment.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user

Request body (`application/json`)

Property	Type	Description
`environment` Required	String	The name of the environment to create the deployment for
`ref` Required	String	The name of the branch or tag that is deployed
`sha` Required	String	The SHA of the commit that is deployed
`status` Required	String	The status of the deployment that is created. One of `running` , `success` , `failed` , or `canceled` Allowed values: `running` , `success` , `failed` , `canceled` Minimum length: `1`
`tag` Required	Boolean	A boolean that indicates if the deployed ref is a tag (`true`) or not (`false`)

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeploymentExtended
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a deployment

`GET /api/v4/projects/{id}/deployments/{deployment_id}`

Retrieves a specified deployment.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user
`deployment_id` Path, required	Integer	The ID of the deployment

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeploymentExtended
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a deployment

`PUT /api/v4/projects/{id}/deployments/{deployment_id}`

Updates a specified deployment.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user
`deployment_id` Path, required	Integer	The ID of the deployment

Request body (`application/json`)

Property	Type	Description
`status` Required	String	The new status of the deployment. One of `running`, `success`, `failed`, or `canceled` Allowed values: `running`, `success`, `failed`, `canceled` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDeploymentExtended
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a deployment

`DELETE /api/v4/projects/{id}/deployments/{deployment_id}`

Deletes a specified deployment that is not currently the last deployment for an environment or in a `running` state.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user
`deployment_id` Path, required	Integer	The ID of the deployment

Responses

Code	Description	Schema
`204`	Deployment destroyed	—
`400`	“Cannot destroy running deployment” or “Deployment currently deployed to environment”	—
`403`	Forbidden	—
`404`	Not Found	—

Approve or reject a deployment

`POST /api/v4/projects/{id}/deployments/{deployment_id}/approval`

Approves or rejects a protected deployment.

Parameters

Name	Type	Description
`deployment_id` Path, required	Integer	The ID of the deployment
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`comment`	String	A comment to go with the approval
`represented_as`	String	The user, group, or role represented by the approval
`status` Required	String	The status of the approval Allowed values: `approved` , `rejected` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDeploymentApproval
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all merge requests associated with a deployment

`GET /api/v4/projects/{id}/deployments/{deployment_id}/merge_requests`

Lists all merge requests shipped with a specified deployment.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`deployment_id` Path, required	Integer	The ID of the deployment
`author_id` Query	Integer	Returns merge requests created by the given user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Returns merge requests created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Returns merge requests created by the given `username`. Mutually exclusive with `assignee_id`
	String	

Name	Type	Description
`reviewer_username` Query		Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `reviewer_id`
`labels` Query	Array of strings	Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one label. Predefined names are case-insensitive
`milestone` Query	String	Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`my_reaction_emoji` Query	String	Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`reviewer_id` Query	Integer or string	Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `reviewer_username`

Name	Type	Description
`state` Query	String	Filter merge requests by state Allowed values: `opened`, `closed`, `locked`, `merged`, `all` Default: `all`
`order_by` Query	String	Returns merge requests ordered by `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at` or `merged_at` fields. Introduced in GitLab 14.8 Allowed values: `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at`, `merged_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`with_merge_status_recheck` Query	Boolean	If `true`, this projection requests (but does not guarantee) an asynchronous recalculation of the `merge_status` field. Enable the `restrict_merge_status_recheck` feature flag to ignore this attribute when requested by users without the Developer,

Name	Type	Description
		Maintainer, or Owner role Default: `false`
`created_after` Query	String (date-time)	Return merge requests created on or after the specified time Example: `2019-03-15T08:00:00Z`
`created_before` Query	String (date-time)	Return merge requests created on or before the specified time Example: `2019-03-15T08:00:00Z`
`updated_after` Query	String (date-time)	Return merge requests updated on or after the specified time Example: `2019-03-15T08:00:00Z`
`updated_before` Query	String (date-time)	Return merge requests updated on or before the specified time Example: `2019-03-15T08:00:00Z`
`merged_after` Query	String (date-time)	Return merge requests merged on or after the specified time Example: `2019-03-15T08:00:00Z`
`merged_before` Query	String (date-time)	Return merge requests merged on or before the specified time Example: `2019-03-15T08:00:00Z`
`view` Query	String	If `simple`, returns the `iid`, URL, title, description, and basic state of merge request Allowed values: `simple`
`scope` Query	String	Return merge requests for the given scope. `reviews_for_me` returns merge requests where

Name	Type	Description
		the current user is assigned as a reviewer Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `reviews_for_me`, `all`
`source_branch` Query	String	Return merge requests with the given source branch
`source_project_id` Query	Integer	Return merge requests with the given source project ID
`target_branch` Query	String	Return merge requests with the given target branch
`search` Query	String	Search merge requests against their `title` and `description`. Combine with the `in` attribute
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description` Example: `title,description`
`wip` Query	String	Filter merge requests by their `wip` status. `yes` returns only draft merge requests, `no` returns non-draft merge requests. Deprecated in GitLab 19.0. Use `draft` instead. Mutually exclusive with `draft` Allowed values: `yes`, `no`
`draft` Query	Boolean	If `true`, returns only draft merge requests. If `false`, returns only non-draft merge

Name	Type	Description
		requests. Mutually exclusive with `wip`
`not` Query	Object	Return merge requests that do not match the parameters supplied. Accepts: `labels`, `milestone`, `author_id`, `author_username`, `assignee_id`, `assignee_username`, `reviewer_id`, `reviewer_username`, `my_reaction_emoji`
`not[author_id]` Query	Integer	`<Negated>` Returns merge requests created by the given user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer or string	`<Negated>` Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[assignee_id]`
`not[reviewer_username]` Query	String	`<Negated>` Returns merge requests which have the user as a reviewer with the given

Name	Type	Description
		`username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `not[reviewer_id]`
`not[labels]` Query	Array of strings	`<Negated>` Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one label. Predefined names are case-insensitive
`not[milestone]` Query	String	`<Negated>` Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`not[my_reaction_emoji]` Query	String	`<Negated>` Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`not[reviewer_id]` Query	Integer	`<Negated>` Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually

Name	Type	Description
		exclusive with `not[reviewer_username]`
`deployed_before` Query	String (date-time)	Return merge requests deployed before the specified time Example: `2019-03-15T08:00:00Z`
`deployed_after` Query	String (date-time)	Return merge requests deployed after the specified time Example: `2019-03-15T08:00:00Z`
`environment` Query	String	Return merge requests deployed to the given environment Example: `production`
`merge_user_id` Query	Integer	Returns merge requests which have been merged by the user with the given user `id`. Mutually exclusive with `merge_user_username`
`merge_user_username` Query	String	Returns merge requests which have been merged by the user with the given `username`. Mutually exclusive with `merge_user_id`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all project deploy keys for a user

`GET /api/v4/users/{user_id}/project_deploy_keys`

Lists all project deploy keys accessible to a specified user.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDeployKey
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDeployKey

Property	Type	Description
`created_at`	String (date-time)	Example: `2015-09-03T07:24:44.627Z`
`expires_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`fingerprint`	String	Example: `4a:9d:64:15:ed:3a:e6:07:6e:89:36:b3:3b:03:05:d9`
`fingerprint_sha256`	String	Example: `Jrs3LD1Ji30xNLtTVf9NDCj7kkBgPBb2pjvTZ3HfIgU`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDNJAkI3Wdf0r13c8a5pEExB2YowPWCSVzfZV22pNBc1CuEbyYLHpUyaD0GwpGvFdx2aP7lMEk35k6Rz3ccBF6jRaVJyhsn5VNnW92PMpBJ/P1UebhXwsFHdQf5rTt082cSxWuk61kGWRQtk4ozt/J2DF/dIUVaLvc+z4HomT41fQ==`
`last_used_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`projects_with_readonly_access`	APIEntitiesProjectIdentity	—
`projects_with_write_access`	APIEntitiesProjectIdentity	—
`title`	String	Example: `Sample key 25`
`usage_type`	String	Example: `auth`

APIEntitiesDeployKeysProject

Property	Type	Description
`can_push`	Boolean	—
`created_at`	String (date-time)	Example: `2015-09-03T07:24:44.627Z`
`expires_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`fingerprint`	String	Example: `4a:9d:64:15:ed:3a:e6:07:6e:89:36:b3:3b:03:05:d9`
`fingerprint_sha256`	String	Example: `Jrs3LD1Ji30xNLtTVf9NDCj7kkBgPBb2pjvTZ3HfIgU`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQDNJAkI3Wdf0r13c8a5pEExB2YowPWCSVzfZV22pNBc1CuEbyYLHpUyaD0GwpGvFdx2aP7lMEk35k6Rz3ccBF6jRaVJyhsn5VNnW92PMpBJ/P1UebhXwsFHdQf5rTt082cSxWuk61kGWRQtk4ozt/J2DF/dIUVaLvc+z4HomT41fQ==`
`last_used_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`projects_with_readonly_access`	APIEntitiesProjectIdentity	—
`projects_with_write_access`	APIEntitiesProjectIdentity	—
`title`	String	Example: `Sample key 25`
`usage_type`	String	Example: `auth`

APIEntitiesDeployment

Property	Type	Description
`created_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`deployable`	APIEntitiesCiJob	—
`environment`	APIEntitiesEnvironmentBasic	—
`id`	Integer (int64)	Example: `41`
`iid`	Integer	Example: `1`
`ref`	String	Example: `main`
`sha`	String	Example: `99d03678b90d914dbb1b109132516d71a4a03ea8`
`status`	String	Example: `created`
`updated_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`user`	APIEntitiesUserBasic	—

APIEntitiesDeploymentApproval

Property	Type	Description
`comment`	String	Example: `Looks good to me`
`created_at`	String (date-time)	Example: `2022-02-24T20:22:30.097Z`
`status`	String	Example: `approved`
`user`	APIEntitiesUserBasic	—

APIEntitiesDeploymentApprovalSummary

Property	Type	Description
`rules`	Array of APIEntitiesDeploymentApprovalSummaryRule	—

APIEntitiesDeploymentApprovalSummaryRule

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`deployment_approvals`	Array of APIEntitiesDeployment Approval	—
`group_id`	Integer (int64)	Example: `3`
`required_approvals`	Integer	Example: `1`
`user_id`	Integer (int64)	Example: `2`

APIEntitiesDeploymentExtended

Property	Type	Description
`approval_summary`	APIEntitiesDeploymentApprovalSummary	—
`approvals`	Array of APIEntitiesDeploymentApproval	—
`created_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`deployable`	APIEntitiesCiJob	—
`environment`	APIEntitiesEnvironmentBasic	—
`id`	Integer (int64)	Example: `41`
`iid`	Integer	Example: `1`
`pending_approval_count`	Integer	Example: `0`
`ref`	String	Example: `main`
`sha`	String	Example: `99d03678b90d914dbb1b109132516d71a4a03ea8`
`status`	String	Example: `created`
`updated_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`user`	APIEntitiesUserBasic	—

APIEntitiesDeployToken

Property	Type	Description
`expired`	Boolean	—
`expires_at`	String (date-time)	Example: `2020-02-14T00:00:00.000Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `MyToken`
`revoked`	Boolean	—
`scopes`	Array	Example: `["read_repository"]`
`username`	String	Example: `gitlab+deploy-token-1`

APIEntitiesDeployTokenWithToken

Property	Type	Description
`expired`	Boolean	—
`expires_at`	String (date-time)	Example: `2020-02-14T00:00:00.000Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `MyToken`
`revoked`	Boolean	—
`scopes`	Array	Example: `["read_repository"]`
`token`	String	Example: `jMRvtPNxrn3crTAGukpZ`
`username`	String	Example: `gitlab+deploy-token-1`

APIEntitiesEnvironmentBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`
`external_url`	String	Example: `https://deploy.gitlab.example.com`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy`
`slug`	String	Example: `deploy`
`updated_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Discussions

Use this API to manage discussions. This includes comments, threads, and system notes about changes to an object (for example, when a milestone changes).

To manage label notes, use the resource label events API.

List all discussions for an issue

`GET /api/v4/projects/{id}/issues/{noteable_id}/discussions`

Lists all discussions for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Create an issue discussion

`POST /api/v4/projects/{id}/issues/{noteable_id}/discussions`

Creates a discussion for a specified issue. Replies can be added to it later.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a discussion for an issue

`GET /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}`

Retrieves a specified discussion for an issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Resolve or unresolve an issue discussion

`PUT /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}`

Resolves or unresolves a specified issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`resolved` Required	Boolean	If `true`, resolves the discussion. If `false`, reopens the discussion

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

List comments in an issue discussion

`GET /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}/notes`

Lists the comments in a specified issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Add a comment to an issue discussion

`POST /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}/notes`

Adds a comment to a specified issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a comment from an issue discussion

`GET /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Retrieves a specified comment from a issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a comment in an issue discussion

`PUT /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Updates or changes the status of a specified comment in a issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note or reply. Exactly one of `body`, `resolved` must be provided
`resolved`	Boolean	If `true`, resolves the note. If `false`, reopens the note. Exactly one of `body`, `resolved` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a comment from an issue discussion

`DELETE /api/v4/projects/{id}/issues/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Deletes a specified comment from a issue discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the issue
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all discussions for a merge request

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions`

Lists all discussions for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Create a merge request discussion

`POST /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions`

Creates a discussion for a specified merge request. Replies can be added to it later.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40z`. Requires administrator or project/group owner rights
`position`	Object	Position when creating a note
`position.base_sha` Required	String	Base commit SHA in the source branch
`position.head_sha` Required	String	SHA referencing HEAD of this merge request
`position.height`	Integer	Height of the image
`position.line_range`	Object	Multi-line start and end
`position.line_range.end`	Object	End line for a multi-line note
`position.line_range.end.line_code`	String	End line code for multi-line note
`position.line_range.end.new_line`	Integer	End new_line line number
`position.line_range.end.old_line`	Integer	End old_line line number
`position.line_range.end.type`	String	End line type for multi-line note
`position.line_range.start`	Object	Start line for a multi-line note
`position.line_range.start.line_code`	String	Start line code for multi-line note
`position.line_range.start.new_line`	Integer	Start new_line line number
`position.line_range.start.old_line`	Integer	Start old_line line number

Property	Type	Description
`position.line_range.start.type`	String	Start line type for multi-line note
`position.new_line`	Integer	Line number after change
`position.new_path`	String	File path after change
`position.old_line`	Integer	Line number before change
`position.old_path`	String	File path before change
`position.position_type` Required	String	Type of the position reference Allowed values: `text`, `image`, `file` Minimum length: `1`
`position.start_sha` Required	String	SHA referencing commit in target branch
`position.width`	Integer	Width of the image
`position.x`	Integer	X coordinate in the image
`position.y`	Integer	Y coordinate in the image

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a discussion for a merge request

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}`

Retrieves a specified discussion for a merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the merge request

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Resolve or unresolve a merge request discussion

`PUT /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}`

Resolves or unresolves a specified merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`resolved` Required	Boolean	If `true`, resolves the discussion. If `false`, reopens the discussion

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

List comments in a merge request discussion

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}/notes`

Lists the comments in a specified merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the merge request

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Add a comment to a merge request discussion

`POST /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}/notes`

Adds a comment to a specified merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a comment from a merge request discussion

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Retrieves a specified comment from a merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a comment in a merge request discussion

`PUT /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Updates or changes the status of a specified comment in a merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note or reply. Exactly one of `body`, `resolved` must be provided
`resolved`	Boolean	If `true`, resolves the note. If `false`, reopens the note. Exactly one of `body`, `resolved` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a comment from a merge request discussion

`DELETE /api/v4/projects/{id}/merge_requests/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Deletes a specified comment from a merge request discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the merge request
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all discussions for a commit

`GET /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions`

Lists all discussions for a specified commit.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Create a commit discussion

`POST /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions`

Creates a discussion for a specified commit. Replies can be added to it later.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z`. Requires administrator or project/group owner rights
`position`	Object	Position when creating a note
`position.base_sha` Required	String	Base commit SHA in the source branch
`position.head_sha` Required	String	SHA referencing HEAD of this merge request
`position.height`	Integer	Height of the image
`position.new_line`	Integer	Line number after change
`position.new_path`	String	File path after change
`position.old_line`	Integer	Line number before change
`position.old_path`	String	File path before change
`position.position_type` Required	String	Type of the position reference Allowed values: `text`, `image`, `file` Minimum length: `1`
`position.start_sha` Required	String	SHA referencing commit in target branch
`position.width`	Integer	Width of the image
`position.x`	Integer	X coordinate in the image
`position.y`	Integer	Y coordinate in the image

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a discussion for a commit

`GET /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}`

Retrieves a specified discussion for a commit.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	String	ID of the commit

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

List comments in a commit discussion

`GET /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}/notes`

Lists the comments in a specified commit discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	String	ID of the commit

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Add a comment to a commit discussion

`POST /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}/notes`

Adds a comment to a specified commit discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z`. Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a comment from a commit discussion

`GET /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Retrieves a specified comment from a commit discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a comment in a commit discussion

`PUT /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Updates or changes the status of a specified comment in a commit discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note or reply. Exactly one of `body` , `resolved` must be provided
`resolved`	Boolean	If `true` , resolves the note. If `false` , reopens the note. Exactly one of `body` , `resolved` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a comment from a commit discussion

`DELETE /api/v4/projects/{id}/repository/commits/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Deletes a specified comment from a commit discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	String	ID of the commit
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all discussions for a snippet

`GET /api/v4/projects/{id}/snippets/{noteable_id}/discussions`

Lists all discussions for a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Create a snippet discussion

`POST /api/v4/projects/{id}/snippets/{noteable_id}/discussions`

Creates a discussion for a specified snippet. Replies can be added to it later.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z`. Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a discussion for a snippet

`GET /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}`

Retrieves a specified discussion for a snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

List comments in a snippet discussion

`GET /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}/notes`

Lists the comments in a specified snippet discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`discussion_id` Path, required	String	ID of the discussion
`noteable_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDiscussion
`400`	Bad Request	—
`404`	Not Found	—

Add a comment to a snippet discussion

`POST /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}/notes`

Adds a comment to a specified snippet discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet
`discussion_id` Path, required	String	ID of the discussion

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note or reply
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40z` . Requires administrator or project/group owner rights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a comment from a snippet discussion

`GET /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Retrieves a specified comment from a snippet discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a comment in a snippet discussion

`PUT /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Updates or changes the status of a specified comment in a snippet discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note or reply. Exactly one of `body`, `resolved` must be provided
`resolved`	Boolean	If `true`, resolves the note. If `false`, reopens the note. Exactly one of `body`, `resolved` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a comment from a snippet discussion

`DELETE /api/v4/projects/{id}/snippets/{noteable_id}/discussions/{discussion_id}/notes/{note_id}`

Deletes a specified comment from a snippet discussion.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the snippet
`discussion_id` Path, required	String	ID of the discussion
`note_id` Path, required	Integer	ID of the discussion note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiscussion

Property	Type	Description
`id`	String	Example: `6a9c1750b37d513a43987b574953fceb50b03ce7`
`individual_note`	Boolean	Example: `false`
`notes`	APIEntitiesNote	—
`resolvable`	Boolean	—
`resolved`	Boolean	—

APIEntitiesNote

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`body`	String	Example: `Note body.`
`commands_changes`	Object	—
`commit_id`	String	Example: `7b09ce7e6f80347baf0316c8c94cdba9a0a7e91d`
`confidential`	Boolean	Example: `false`
`created_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`internal`	Boolean	Example: `false`
`noteable_id`	Integer (int64)	Example: `1`
`noteable_iid`	Integer	Example: `1`
`noteable_type`	String	Example: `Issue`
`position`	Object	—
`project_id`	Integer (int64)	Example: `1`
`resolvable`	Boolean	Example: `false`
`resolved`	Boolean	Example: `false`
`resolved_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`resolved_by`	APIEntitiesUserBasic	—
`suggestions`	APIEntitiesSuggestion	—
`system`	Boolean	Example: `false`
`type`	String	Example: `DiscussionNote`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`

APIEntitiesSuggestion

Property	Type	Description
`appliable`	Boolean	Example: `true`
`applied`	Boolean	Example: `false`
`from_content`	String	Example: `Original content`
`from_line`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`to_content`	String	Example: `New content`
`to_line`	Integer	Example: `1`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Draft notes

Use this API to manage draft notes. These notes are pending, unpublished comments on merge requests. Draft notes can start a discussion, or continue an existing discussion as a reply.

Before publishing, draft notes are visible only to the author.

List all merge request draft notes

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes`

Lists all merge request draft notes.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDraftNote
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a draft note

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes`

Creates a draft note for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Request body (`application/json`)

Property	Type	Description
`commit_id`	String	SHA of the commit to associate the draft note with
`in_reply_to_discussion_id`	String	ID of the discussion the draft note replies to
`note` Required	String	Content of the note
`position`	Object	Position when creating a diff note. If omitted, creates a regular discussion note
`position.base_sha` Required	String	Base commit SHA of the diff
`position.head_sha` Required	String	SHA of the latest commit in the source branch
`position.height`	Integer	Height of the image
`position.line_range`	Object	Line range for a multi-line note
`position.line_range.end`	Object	End line for a multi-line note
`position.line_range.end.line_code`	String	End line code for a multi-line note
`position.line_range.end.new_line`	Integer	Line number after the change at the end of the line range
`position.line_range.end.old_line`	Integer	Line number before the change at the end of the line range
`position.line_range.end.type`	String	End line type for a multi-line note
`position.line_range.start`	Object	Start line for a multi-line note
`position.line_range.start.line_code`	String	Start line code for a multi-line note
`position.line_range.start.new_line`	Integer	Line number after the change at the start of the line range
	Integer	

Property	Type	Description
`position.line_range.start.old_line`		Line number before the change at the start of the line range
`position.line_range.start.type`	String	Start line type for a multi-line note
`position.new_line`	Integer	Line number after the change
`position.new_path`	String	File path after the change
`position.old_line`	Integer	Line number before the change
`position.old_path`	String	File path before the change
`position.position_type` Required	String	Type of the position reference Allowed values: `text`, `image`, `file` Minimum length: `1`
`position.start_sha` Required	String	SHA of the latest commit in the target branch
`position.width`	Integer	Width of the image
`position.x`	Integer	X coordinate in the image
`position.y`	Integer	Y coordinate in the image
`resolve_discussion`	Boolean	If `true`, resolves the associated discussion

Responses

Code	Description	Schema
`201`	Created	APIEntitiesDraftNote
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Publish all pending draft notes

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes/bulk_publish`

Publishes all pending draft notes for the current user on the specified merge request. Optionally sets the reviewer state and posts a summary note.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Request body (`application/json`)

Property	Type	Description
`internal`	Boolean	If `true` , the summary note is internal Default: `false`
`note`	String	Text of the summary note to add to the merge request
`reviewer_state`	String	Sets the review state after publishing. Does not record a formal approval Allowed values: `requested_changes` , `reviewed`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a draft note

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes/{draft_note_id}`

Retrieves a draft note for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`draft_note_id` Path, required	Integer	ID of the draft note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDraftNote
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a draft note

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes/{draft_note_id}`

Updates a draft note for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`draft_note_id` Path, required	Integer	ID of the draft note

Request body (`application/json`)

Property	Type	Description
`note`	String	Content of the note
`position`	Object	Position when creating a diff note. If omitted, creates a regular discussion note
`position.base_sha` Required	String	Base commit SHA of the diff
`position.head_sha` Required	String	SHA of the latest commit in the source branch
`position.height`	Integer	Height of the image
`position.line_range`	Object	Line range for a multi-line note
`position.line_range.end`	Object	End line for a multi-line note
`position.line_range.end.line_code`	String	End line code for a multi-line note
`position.line_range.end.new_line`	Integer	Line number after the change at the end of the line range
`position.line_range.end.old_line`	Integer	Line number before the change at the end of the line range
`position.line_range.end.type`	String	End line type for a multi-line note
`position.line_range.start`	Object	Start line for a multi-line note
`position.line_range.start.line_code`	String	Start line code for a multi-line note
`position.line_range.start.new_line`	Integer	Line number after the change at the start of the line range
`position.line_range.start.old_line`	Integer	Line number before the change at the start of the line range
`position.line_range.start.type`	String	Start line type for a multi-line note
`position.new_line`	Integer	Line number after the change

Property	Type	Description
`position.new_path`	String	File path after the change
`position.old_line`	Integer	Line number before the change
`position.old_path`	String	File path before the change
`position.position_type` Required	String	Type of the position reference Allowed values: `text`, `image`, `file` Minimum length: `1`
`position.start_sha` Required	String	SHA of the latest commit in the target branch
`position.width`	Integer	Width of the image
`position.x`	Integer	X coordinate in the image
`position.y`	Integer	Y coordinate in the image

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDraftNote
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete a draft note

`DELETE /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes/{draft_note_id}`

Deletes a draft note for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`draft_note_id` Path, required	Integer	ID of the draft note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDraftNote
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Publish a draft note

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}/draft_notes/{draft_note_id}/publish`

Publishes a draft note for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`draft_note_id` Path, required	Integer	ID of the draft note

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesDraftNote

Property	Type	Description
`author_id`	Integer (int64)	Example: `4`
`commit_id`	String	Example: `6104942438c14ec7bd21c6cd5bd995272b3faff6`
`discussion_id`	String	Example: `6a9c1750b37d513a43987b574953fceb50b03ce7`
`id`	Integer (int64)	Example: `2`
`line_code`	String	Example: `1c497fbb3a46b78edf0_2_4`
`merge_request_id`	Integer (int64)	Example: `52`
`note`	String	Example: `This is a note`
`position`	Object	Example: `{"base_sha":"aa149113","head_sha":"be3020c7","line_range":{"end":{"line_code":"1c497fbb3a46b78edf04cc2a2fa33f67e3ffbe2a_2_4","new_line":4,"old_line":2,"type":null},"start":{"line_code":"1c497fbb3a46b78edf04cc2a2fa33f67e3ffbe2a_2_4","new_line":4,"old_line":2,"type":null}},"new_line":4,"new_path":"example.md","old_line":2,"old_path":"example.md","position_type":"text","start_sha":"b3a0a8c4"}`
`resolve_discussion`	Boolean	Example: `true`

Environments

Use this API to interact with GitLab environments.

List all environments

`GET /api/v4/projects/{id}/environments`

Lists all environments for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`name` Query	String	Return the environment with this name. Mutually exclusive with `search`
`search` Query	String	Return list of environments matching the search criteria. Must be at least 3 characters. Mutually exclusive with `name`
`states` Query	String	Filter environments by state. If omitted, returns all environments Allowed values: `stopped`, `stopping`, `available`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntities Environment
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create an environment

`POST /api/v4/projects/{id}/environments`

Creates an environment for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`auto_stop_setting`	String	Auto stop setting for the environment Allowed values: `always`, `with_action`
`cluster_agent_id`	Integer	ID of the cluster agent to associate with the environment. When updating an environment, set to `null` to remove it
`description`	String	Description of the environment
`external_url`	String	URL to link to for the environment
`flux_resource_path`	String	Flux resource path to associate with the environment, for example `helm.toolkit.fluxcd.io/v2/namespaces/gitlab-agent/helmreleases/gitlab-agent`. When updating an environment, set to `null` to remove it
`kubernetes_namespace`	String	Kubernetes namespace to associate with the environment. When updating an environment, set to `null` to remove it
`name` Required	String	Name of the environment
`tier`	String	Tier of the environment Allowed values: `production`, `staging`, `testing`, `development`, `other`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesEnvironment
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Schedule multiple stopped review apps for deletion

`DELETE /api/v4/projects/{id}/environments/review_apps`

Schedules multiple stopped review apps for deletion. The deletion is performed after 1 week. By default, only environments 30 days or older are deleted.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`before` Query	String (date-time)	Date before which environments can be deleted. Defaults to 30 days ago
`limit` Query	Integer	Maximum number of environments to delete Default: `100` Maximum: `1000` Minimum: `1`
`dry_run` Query	Boolean	If `true`, performs a dry run where no deletion is actually performed. If `false`, deletes the environment Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEnvironmentBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`409`	Conflict	—

Stop stale environments

`POST /api/v4/projects/{id}/environments/stop_stale`

Stops all environments that were last modified or deployed to before a specified date. Excludes protected environments.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`before` Required	String (date-time)	Stop environments that were modified or deployed to before the specified date. Valid inputs are between 10 years ago and 1 week ago

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve an environment

`GET /api/v4/projects/{id}/environments/{environment_id}`

Retrieves a specified environment for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`environment_id` Path, required	Integer	ID of the environment

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEnvironment
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update an existing environment

`PUT /api/v4/projects/{id}/environments/{environment_id}`

Updates an existing environment for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`environment_id` Path, required	Integer	ID of the environment

Request body (`application/json`)

Property	Type	Description
`auto_stop_setting`	String	Auto stop setting for the environment Allowed values: `always`, `with_action`
`cluster_agent_id`	Integer	ID of the cluster agent to associate with the environment. When updating an environment, set to `null` to remove it
`description`	String	Description of the environment
`external_url`	String	New URL to link to for the environment
`flux_resource_path`	String	Flux resource path to associate with the environment, for example `helm.toolkit.fluxcd.io/v2/namespaces/gitlab-agent/helmreleases/gitlab-agent`. When updating an environment, set to `null` to remove it
`kubernetes_namespace`	String	Kubernetes namespace to associate with the environment. When updating an environment, set to `null` to remove it
`tier`	String	Tier of the environment Allowed values: `production`, `staging`, `testing`, `development`, `other`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEnvironment
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete an environment

`DELETE /api/v4/projects/{id}/environments/{environment_id}`

Deletes an environment from a project. The environment must be stopped first.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`environment_id` Path, required	Integer	ID of the environment

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEnvironment
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Stop an environment

`POST /api/v4/projects/{id}/environments/{environment_id}/stop`

Stops a specified running environment.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`environment_id` Path, required	Integer	ID of the environment

Request body (`application/json`)

Property	Type	Description
`force`	Boolean	If `true` , stops the environment without executing `on_stop` actions Default: `false`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesEnvironment
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesClustersAgent

Property	Type	Description
`config_project`	APIEntitiesProjectIdentity	—
`created_at`	String (date-time)	—
`created_by_user_id`	Integer (int64)	Example: `1`
`id`	Integer (int64)	Example: `1`
`name`	String	—

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDeployment

Property	Type	Description
`created_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`deployable`	APIEntitiesCiJob	—
`environment`	APIEntitiesEnvironmentBasic	—
`id`	Integer (int64)	Example: `41`
`iid`	Integer	Example: `1`
`ref`	String	Example: `main`
`sha`	String	Example: `99d03678b90d914dbb1b109132516d71a4a03ea8`
`status`	String	Example: `created`
`updated_at`	String (date-time)	Example: `2016-08-11T11:32:35.444Z`
`user`	APIEntitiesUserBasic	—

APIEntitiesEnvironment

Property	Type	Description
`auto_stop_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`
`auto_stop_setting`	String	Example: `always`
`cluster_agent`	APIEntitiesClustersAgent	—
`created_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`
`description`	String	Example: `description`
`external_url`	String	Example: `https://deploy.gitlab.example.com`
`flux_resource_path`	String	—
`id`	Integer (int64)	Example: `1`
`kubernetes_namespace`	String	—
`last_deployment`	APIEntitiesDeployment	—
`name`	String	Example: `deploy`
`project`	APIEntitiesBasicProjectDetails	—
`slug`	String	Example: `deploy`
`state`	String	Example: `available`
`tier`	String	Example: `development`
`updated_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`

APIEntitiesEnvironmentBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`
`external_url`	String	Example: `https://deploy.gitlab.example.com`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy`
`slug`	String	Example: `deploy`
`updated_at`	String (date-time)	Example: `2019-05-25T18:55:13.252Z`

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Error tracking

Use this API to manage error tracking settings and client keys for a project.

These endpoints require the Maintainer or Owner role.

List all project client keys

`GET /api/v4/projects/{id}/error_tracking/client_keys`

Lists all integrated error tracking client keys for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesErrorTrackingClientKey
`400`	Bad Request	—
`404`	Not Found	—

Create a client key

`POST /api/v4/projects/{id}/error_tracking/client_keys`

Creates a client key for integrated error tracking in a specified project. The public key attribute is generated automatically.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesErrorTrackingClientKey
`400`	Bad Request	—
`404`	Not Found	—

Delete a client key

`DELETE /api/v4/projects/{id}/error_tracking/client_keys/{key_id}`

Deletes an integrated error tracking client key from a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`key_id` Path, required	Integer	ID of the client key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesErrorTrackingClientKey
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve Error Tracking settings for a project

`GET /api/v4/projects/{id}/error_tracking/settings`

Retrieves the Error Tracking settings for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesErrorTrackingProjectSetting
`400`	Bad Request	—
`404`	Not Found	—

Create Error Tracking settings for a project

`PUT /api/v4/projects/{id}/error_tracking/settings`

Creates Error Tracking settings for a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user

Request body (`application/json`)

Property	Type	Description
`active` Required	Boolean	Pass true to enable the configured Error Tracking settings or false to disable it
`integrated` Required	Boolean	Pass true to enable the integrated Error Tracking backend

Responses

Code	Description	Schema
`200`	OK	APIEntitiesErrorTrackingProjectSetting
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Update Error Tracking settings for a project

`PATCH /api/v4/projects/{id}/error_tracking/settings`

Updates Error Tracking settings for a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project owned by the authenticated user

Request body (`application/json`)

Property	Type	Description
`active` Required	Boolean	Pass true to enable the already configured Error Tracking settings or false to disable it
`integrated`	Boolean	Pass true to enable the integrated Error Tracking backend. Available in GitLab 14.2 and later

Responses

Code	Description	Schema
`200`	OK	APIEntitiesErrorTrackingProjectSetting
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesErrorTrackingClientKey

Property	Type	Description
`active`	Boolean	—
`id`	Integer (int64)	Example: `1`
`public_key`	String	Example: `glet_aa77551d849c083f76d0bc545ed053a3`
`sentry_dsn`	String	Example: `https://glet_aa77551d849c083f76d0bc545ed053a3@example.com/errortracking/api/v1/projects/5`

APIEntitiesErrorTrackingProjectSetting

Property	Type	Description
`active`	Boolean	—
`api_url`	String	Example: `https://sentry.io/api/0/projects/myawesomeproject/project`
`integrated`	Boolean	—
`project_name`	String	Example: `sample sentry project`
`sentry_external_url`	String	Example: `https://sentry.io/myawesomeproject/project`

Events

Use this API to review event activity for a user, a project, or an instance. Events cover a wide range of actions, such as joining a project, commenting on an issue, pushing changes to a merge request, or closing an epic.

Activity is subject to retention limits. For more information, see the user activity time period limit and the project activity time period limit.

These endpoints have some limitations:

- Some epic features, such as child items, linked items, start dates, due dates, and health statuses, are not returned.
- Some merge request notes use the `DiscussionNote` type instead, which is not supported.
- Bulk push events, created when a push exceeds the push event activities limit, are returned with limited details.

List all events

`GET /api/v4/events`

Lists all events for the authenticated user. Does not return events associated with epics or merge requests. Returns bulk push events with limited commit details.

Parameters

Name	Type	Description
`scope` Query	String	Include all events across the user’s projects Example: `all`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`action` Query	String	Event action to filter on
`target_type` Query	String	Event target type to filter on Allowed values: `issue`, `milestone`, `merge_request`, `note`, `project`, `snippet`, `user`, `wiki`, `design`
`before` Query	String (date)	Include only events created before this date
`after` Query	String (date)	Include only events created after this date
`sort` Query	String	Return events sorted in ascending and descending order Allowed values: `asc`, `desc` Default: `desc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesEvent
`400`	Bad Request	—
`401`	Unauthorized	—

List all visible events for a project

`GET /api/v4/projects/{id}/events`

Lists all visible events for a specified project. When a push exceeds the Push event activities limit, a single bulk push event is returned instead of individual commit events. Bulk push events have limited commit details: `commit_count` is `0`, `ref_count` shows the number of refs pushed, and individual commit attributes are `null`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`action` Query	String	Event action to filter on
`target_type` Query	String	Event target type to filter on Allowed values: `issue`, `milestone`, `merge_request`, `note`, `project`, `snippet`, `user`, `wiki`, `design`
`before` Query	String (date)	Include only events created before this date
`after` Query	String (date)	Include only events created after this date
`sort` Query	String	Return events sorted in ascending and descending order Allowed values: `asc`, `desc` Default: `desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve contribution events for a user

`GET /api/v4/users/{id}/events`

Retrieves the contribution events for a specified user. Does not return events associated with epics or merge requests. Returns bulk push events with limited commit details.

Parameters

Name	Type	Description
`id` Path, required	String	ID or username of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`action` Query	String	Event action to filter on
`target_type` Query	String	Event target type to filter on Allowed values: `issue`, `milestone`, `merge_request`, `note`, `project`, `snippet`, `user`, `wiki`, `design`
`before` Query	String (date)	Include only events created before this date
`after` Query	String (date)	Include only events created after this date
`sort` Query	String	Return events sorted in ascending and descending order Allowed values: `asc`, `desc` Default: `desc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesEvent
`400`	Bad Request	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesEvent

Property	Type	Description
`action_name`	String	Example: `closed`
`author`	APIEntitiesUserBasic	—
`author_id`	Integer (int64)	Example: `25`
`author_username`	String	Example: `root`
`created_at`	String	Example: `2017-02-09T10:43:19.667Z`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `none`
`note`	APIEntitiesNote	—
`project_id`	Integer (int64)	Example: `2`
`push_data`	APIEntitiesPushEventPayload	—
`target_id`	Integer (int64)	Example: `160`
`target_iid`	Integer	Example: `157`
`target_title`	String	Example: `Public project search field`
`target_type`	String	Example: `Issue`
`wiki_page`	APIEntitiesWikiPageBasic	—

APIEntitiesNote

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`body`	String	Example: `Note body.`
`commands_changes`	Object	—
`commit_id`	String	Example: `7b09ce7e6f80347baf0316c8c94cdba9a0a7e91d`
`confidential`	Boolean	Example: `false`
`created_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`internal`	Boolean	Example: `false`
`noteable_id`	Integer (int64)	Example: `1`
`noteable_iid`	Integer	Example: `1`
`noteable_type`	String	Example: `Issue`
`position`	Object	—
`project_id`	Integer (int64)	Example: `1`
`resolvable`	Boolean	Example: `false`
`resolved`	Boolean	Example: `false`
`resolved_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`resolved_by`	APIEntitiesUserBasic	—
`suggestions`	APIEntitiesSuggestion	—
`system`	Boolean	Example: `false`
`type`	String	Example: `DiscussionNote`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`

APIEntitiesPushEventPayload

Property	Type	Description
`action`	String	Example: `pushed`
`commit_count`	Integer	Example: `1`
`commit_from`	String	Example: `50d4420237a9de7be1304607147aec22e4a14af7`
`commit_title`	String	Example: `Add simple search to projects in public area`
`commit_to`	String	Example: `c5feabde2d8cd023215af4d2ceeb7a64839fc428`
`ref`	String	Example: `master`
`ref_count`	Integer	Example: `1`
`ref_type`	String	Example: `branch`

APIEntitiesSuggestion

Property	Type	Description
`appliable`	Boolean	Example: `true`
`applied`	Boolean	Example: `false`
`from_content`	String	Example: `Original content`
`from_line`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`to_content`	String	Example: `New content`
`to_line`	Integer	Example: `1`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesWikiPageBasic

Property	Type	Description
`format`	String	Example: `markdown`
`slug`	String	Example: `deploy`
`title`	String	Example: `deploy`
`wiki_page_meta_id`	Integer (int64)	Example: `123`

External status checks

Use this API to manage external status checks.

Get list of project external status checks

`GET /api/v4/projects/{id}/external_status_checks`

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesExternalStatusCheck
`400`	Bad Request	—
`404`	Not Found	—

Create external status check

`POST /api/v4/projects/{id}/external_status_checks`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branch_scope` Required	String	External status check branch scope Allowed values: `all_branches`, `all_protected_branches`, `protected_branch_specific` Minimum length: `1`
`external_url` Required	String	External status check resource url Example: `https://www.example.com`
`name` Required	String	External status check display name Example: `lorem ipsum`
`protected_branch_ids`	Array of integers	IDs of protected branches
`shared_secret`	String	HMAC shared secret Example: `hmac`
`timeout_minutes`	Integer	External status check timeout in minutes Example: `5`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesExternalStatusCheck
`400`	Bad Request	—
`404`	Not Found	—

Update external status check by id

`PUT /api/v4/projects/{id}/external_status_checks/{check_id}`

Parameters

Name	Type	Description
`check_id` Path, required	Integer	External status check ID Example: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branch_scope`	String	External status check branch scope. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must be provided Allowed values: `all_branches`, `all_protected_branches`, `protected_branch_specific`
`external_url`	String	External status check resource url. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must be provided Example: `https://www.example.com`
`name`	String	Display name of external status check. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must be provided Example: `lorem ipsum`
`protected_branch_ids`	Array of integers	IDs of protected branches to scope the rule by. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must be provided
`shared_secret`	String	HMAC shared secret. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must

Property	Type	Description
		be provided Example: `hmac`
`timeout_minutes`	Integer	External status check timeout in minutes. At least one of `name`, `shared_secret`, `external_url`, `timeout_minutes`, `branch_scope`, `protected_branch_ids` must be provided Example: `5`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesExternalStatusCheck
`400`	Bad Request	—
`404`	Not Found	—

Delete external status check

`DELETE /api/v4/projects/{id}/external_status_checks/{check_id}`

Parameters

Name	Type	Description
`check_id` Path, required	Integer	ID of an external status check
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Set status of an external status check

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/status_check_responses`

Parameters

Name	Type	Description
`id` Path, required	String	ID of a project Example: `1`
`merge_request_iid` Path, required	Integer	IID of a merge request Example: `1`

Request body (`application/json`)

Property	Type	Description
`external_status_check_id` Required	Integer	ID of an external status check Example: `1`
`sha` Required	String	SHA at HEAD of the source branch Example: `5957a570eee0ac4580ec027fb874ad7514d1e576`
`status` Required	String	Set to `passed` to pass the check or `failed` to fail it Allowed values: `passed`, `failed` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequestsExternalStatusCheckResponse
`400`	Bad Request	—
`404`	Not Found	—

List status checks for a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/status_checks`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestsExternalStatusCheck
`404`	Not Found	—

Retry failed external status check

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/status_checks/{external_status_check_id}/retry`

Parameters

Name	Type	Description
`id` Path, required	String	Project ID Example: `1`
`merge_request_iid` Path, required	Integer	Merge Request IID Example: `1`
`external_status_check_id` Path, required	Integer	Failed external status check ID

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiPipeline

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`
`tag`	Boolean	Example: `false`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiffRefs

Property	Type	Description
`base_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`head_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`start_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`

APIEntitiesExternalStatusCheck

Property	Type	Description
`branch_scope`	String	Example: `all_branches`
`external_url`	String	Example: `https://www.example.com`
`hmac?`	Boolean	Example: `true`
`id`	Integer	Example: `100`
`name`	String	Example: `lorem ipsum`
`project_id`	Integer	Example: `101`
`protected_branches`	Array of APIEntitiesProtected Branch	—
`timeout_minutes`	Integer	Example: `5`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesMergeRequest

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`changes_count`	String	Example: `1`
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`diff_refs`	APIEntitiesDiffRefs	—
`discussion_locked`	Boolean	—
`diverged_commits_count`	Integer	Example: `0`
`downvotes`	Integer	—
`draft`	Boolean	—
`first_contribution`	Boolean	—
`first_deployed_to_production_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`force_remove_source_branch`	Boolean	—

Property	Type	Description
`has_conflicts`	Boolean	—
`head_pipeline`	APIEntitiesCiPipeline	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`
`labels`	Array of strings	—
`latest_build_finished_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`latest_build_started_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_error`	String	Example: `Merge error`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`pipeline`	APIEntitiesCiPipelineBasic	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`

Property	Type	Description
`rebase_in_progress`	Boolean	—
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`
`subscribed`	Boolean	—
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user`	Object	—

Property	Type	Description
`user.can_merge`	Boolean	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestsExternalStatusCheck

Property	Type	Description
`error_reason`	String	Example: `Timeout error`
`external_url`	String	Example: `https://www.test.com`
`id`	Integer	Example: `1`
`name`	String	Example: `Security`
`status`	String	Example: `pending`

APIEntitiesMergeRequestsExternalStatusCheckResponse

Property	Type	Description
`external_status_check`	APIEntitiesExternalStatusCheck	—
`id`	Integer	Example: `1`
`merge_request`	APIEntitiesMergeRequest	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesProtectedBranch

Property	Type	Description
`allow_force_push`	Boolean	—
`code_owner_approval_required`	Boolean	—
`id`	Integer (int64)	Example: `1`
`merge_access_levels`	Array of APIEntitiesProtectedRefAccess	—
`name`	String	Example: `main`
`push_access_levels`	Array of APIEntitiesProtectedRefAccess	—

APIEntitiesProtectedRefAccess

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`deploy_key_id`	Integer (int64)	Example: `1`
`group_id`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`user_id`	Integer	Example: `1`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

DetailedStatusEntity

Property	Type	Description
`action`	Object	—
`action.button_title`	String	Example: `Cancel this job`
`action.confirmation_message`	String	Example: `Are you sure?`
`action.icon`	String	Example: `cancel`
`action.method`	String	Example: `post`
`action.path`	String	Example: `/namespace1/project1/-/jobs/2/cancel`
`action.title`	String	Example: `Cancel`
`details_path`	String	Example: `/test-group/test-project/-/pipelines/287`
`favicon`	String	Example: `/assets/ci_favicons/favicon_status_success.png`
`group`	String	Example: `success`
`has_details`	Boolean	Example: `true`
`icon`	String	Example: `status_success`
`illustration`	Object	Example: `{"content":"This job depends on upstream jobs that need to succeed in order for this job to be triggered","image":"illustrations/empty-state/empty-job-not-triggered-md.svg","size":"","title":"This job has not been triggered yet"}`
`label`	String	Example: `passed`
`text`	String	Example: `passed`
`tooltip`	String	Example: `passed`

Feature flags

Use this API to manage a project's feature flags and the user lists that scope them to particular users.

These endpoints require the Developer, Maintainer, or Owner role.

List all feature flags for a project

`GET /api/v4/projects/{id}/feature_flags`

Lists all feature flags of the requested project. Use the `page` and `per_page` pagination parameters to control the pagination of results.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`scope` Query	String	Filter feature flags by condition Allowed values: `enabled`, `disabled`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesFeature Flag
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a feature flag

`POST /api/v4/projects/{id}/feature_flags`

Creates a feature flag for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`active`	Boolean	If `true` , the feature flag is active Default: `true`
`description`	String	Description of the feature flag
`name` Required	String	Name of the feature flag
`strategies`	Array of objects	Array of feature flag strategies
`strategies[].name` Required	String	Name of the strategy. Can be `default` , `gradualRolloutUserId` , `userWithId` , `gitlabUserList` , or `flexibleRollout`
`strategies[].parameters`	Object	Strategy parameters as a JSON-formatted string, for example `{"userIds":"user1"}`
`strategies[].scopes`	Array of objects	Array of scopes for the strategy
`strategies[].scopes[].environment_scope` Required	String	Environment scope of the scope
`strategies[].user_list_id`	Integer	ID of the feature flag user list. Applies only if the strategy is `gitlabUserList`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesFeatureFlag
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a feature flag

`GET /api/v4/projects/{id}/feature_flags/{feature_flag_name}`

Retrieves a specified feature flag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`feature_flag_name` Path, required	String	Name of the feature flag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlag
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a feature flag

`PUT /api/v4/projects/{id}/feature_flags/{feature_flag_name}`

Updates a specified feature flag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`feature_flag_name` Path, required	String	Name of the feature flag

Request body (`application/json`)

Property	Type	Description
`active`	Boolean	If `true` , the feature flag is active
`description`	String	Description of the feature flag
`name`	String	New name of the feature flag
`strategies`	Array of objects	Array of feature flag strategies
`strategies[]._destroy`	Boolean	If `true` , deletes the strategy
`strategies[].id`	Integer	ID of the feature flag strategy
`strategies[].name`	String	Name of the strategy
`strategies[].parameters`	Object	Strategy parameters as a JSON-formatted string, for example `{"userIds":"user1"}`
`strategies[].scopes`	Array of objects	Array of scopes for the strategy
`strategies[].scopes[]._destroy`	Boolean	If `true` , deletes the scope
`strategies[].scopes[].environment_scope`	String	Environment scope of the scope
`strategies[].scopes[].id`	Integer	ID of the scope
`strategies[].user_list_id`	Integer	ID of the feature flag user list. Applies only if the strategy is `gitlabUserList`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlag
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a feature flag

`DELETE /api/v4/projects/{id}/feature_flags/{feature_flag_name}`

Deletes a specified feature flag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`feature_flag_name` Path, required	String	Name of the feature flag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlag
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve feature flag settings

`GET /api/v4/projects/{id}/feature_flags_settings`

Retrieves the feature flag settings of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlagSettings
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update feature flag settings

`PUT /api/v4/projects/{id}/feature_flags_settings`

Updates the feature flag settings of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`minimum_role` Required	String	Minimum role required to create, update, toggle, and delete feature flags Allowed values: `no_one_allowed` , `developer` , `maintainer` , `owner` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlagSettings
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all feature flag user lists for a project

`GET /api/v4/projects/{id}/feature_flags_user_lists`

Lists all feature flag user lists for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`search` Query	String	Return user lists matching the search criteria
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesFeatureFlagUserList
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a feature flag user list

`POST /api/v4/projects/{id}/feature_flags_user_lists`

Creates a feature flag user list in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`name` Required	String	The name of the list
`user_xids` Required	String	A comma separated list of external user ids

Responses

Code	Description	Schema
`201`	Created	APIEntitiesFeatureFlagUserList
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a feature flag user list

`GET /api/v4/projects/{id}/feature_flags_user_lists/{iid}`

Retrieves a specified feature flag user list.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`iid` Path, required	String or integer	The internal ID of the project's feature flag user list

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlagUserList
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a feature flag user list

`PUT /api/v4/projects/{id}/feature_flags_user_lists/{iid}`

Updates a specified feature flag user list.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`iid` Path, required	String or integer	The internal ID of the project's feature flag user list

Request body (`application/json`)

Property	Type	Description
`name`	String	The name of the list
`user_xids`	String	A comma separated list of external user ids

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFeatureFlagUserList
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete feature flag user list

`DELETE /api/v4/projects/{id}/feature_flags_user_lists/{iid}`

Deletes a specified feature flag user list.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`iid` Path, required	String or integer	The internal ID of the project's feature flag user list

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`409`	Conflict	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesFeatureFlag

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	Example: `2019-11-04T08:13:51.423Z`
`description`	String	Example: `merge train feature flag`
`name`	String	Example: `merge_train`
`scopes`	Array	—
`strategies`	APIEntitiesFeatureFlagStrategy	—
`updated_at`	String (date-time)	Example: `2019-11-04T08:13:51.423Z`
`version`	String	Example: `new_version_flag`

APIEntitiesFeatureFlagBasicUserList

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `1`
`name`	String	Example: `user_list`
`user_xids`	String	Example: `user1,user2`

APIEntitiesFeatureFlagScope

Property	Type	Description
`environment_scope`	String	Example: `production`
`id`	Integer (int64)	Example: `1`

APIEntitiesFeatureFlagSettings

Property	Type	Description
`minimum_role`	String	Example: `developer`

APIEntitiesFeatureFlagStrategy

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `userWithId`
`parameters`	String	Example: `{"userIds": "user1"}`
`scopes`	APIEntitiesFeatureFlagScope	—
`user_list`	APIEntitiesFeatureFlagBasicUserList	—

APIEntitiesFeatureFlagUserList

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-02-04T08:13:10.507Z`
`edit_path`	String	—
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `1`
`name`	String	Example: `user_list`
`path`	String	—
`project_id`	Integer (int64)	Example: `2`
`updated_at`	String (date-time)	Example: `2020-02-04T08:13:10.507Z`
`user_xids`	String	Example: `user1,user2`

Features

Use this API to manage the Flipper-based feature flags used in the development of GitLab itself.

These endpoints support only boolean and percentage-of-time gate values, and require administrator authorization.

List all feature flags

`GET /api/v4/features`

Lists all feature flags for the instance.

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesFeature
`401`	Unauthorized	—
`403`	Forbidden	—

List all feature flag definitions

`GET /api/v4/features/definitions`

Lists all feature flag definitions.

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesFeature Definition
`401`	Unauthorized	—
`403`	Forbidden	—

Create or update a feature flag

`POST /api/v4/features/{name}`

Creates or updates a feature flag value. If a feature with the given name doesn’t exist yet, the operation creates one. The value can be a boolean or an integer to indicate percentage of time.

Parameters

Name	Type	Description
`name` Path, required	String	Name of the feature flag

Request body (`application/json`)

Property	Type	Description
`endpoint`	String	Caller ID identifying a code path, for example `GET /api/v4/projects/:id` or `Projects Controller#show`. Use a comma to separate multiple endpoint paths. Mutually exclusive with `key`
`feature_group`	String	Name of the feature group. Mutually exclusive with `key`
`force`	Boolean	If `true`, skips feature flag validation checks, such as a YAML definition
`group`	String	GitLab group path, for example `gitlab-org`, or a comma-separated list of group paths. Mutually exclusive with `key`
`key`	String	Rollout strategy for a percentage `value`. Omit it to apply the percentage to time. Mutually exclusive with `feature_group`, `user`, `group`, `namespace`, `project`, `organization`, `repository`, `runner`, `endpoint` Allowed values: `percentage_of_actors`, `percentage_of_time`
`namespace`	String	GitLab group or user namespace path, for example `john-doe`, or a comma-separated list of namespace paths. Introduced in GitLab 15.0. Mutually exclusive with `key`
`organization`	String	Organization ID or path, for example `1` or `default`, or a comma-separated list of

Property	Type	Description
		organization IDs or paths. Mutually exclusive with `key`
`project`	String	Project path, for example `gitlab-org/gitlab-foss`, or a comma-separated list of project paths. Mutually exclusive with `key`
`repository`	String	Repository path, for example `gitlab-org/gitlab-test.git`, `gitlab-org/gitlab-test.wiki.git`, or `snippets/21.git`. Use commas to separate multiple repository paths. Mutually exclusive with `key`
`runner`	String	Runner ID or a comma-separated list of runner IDs. Mutually exclusive with `key`
`user`	String	GitLab username or a comma-separated list of usernames. Mutually exclusive with `key`
`value` Required	String or integer	Value to set for the feature flag. Use `true` or `false` to enable or disable it, or an integer for the percentage of time it should be enabled

Responses

Code	Description	Schema
`201`	Created	APIEntitiesFeature
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a feature

`DELETE /api/v4/features/{name}`

Deletes a feature gate. Returns the same response if the feature gate does not exist.

Parameters

Name	Type	Description
`name` Path, required	String	Name of the feature flag

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesFeature

Property	Type	Description
`definition`	APIEntitiesFeatureDefinition	—
`gates`	APIEntitiesFeatureGate	—
`name`	String	Example: `experimental_feature`
`state`	String	Example: `off`

APIEntitiesFeatureDefinition

Property	Type	Description
`default_enabled`	Boolean	—
`feature_issue_url`	String	—
`group`	String	—
`intended_to_rollout_by`	String	—
`introduced_by_url`	String	—
`log_state_changes`	Boolean	—
`milestone`	String	—
`name`	String	—
`rollout_issue_url`	String	—
`type`	String	—

APIEntitiesFeatureGate

Property	Type	Description
`key`	String	Example: `percentage_of_actors`
`value`	Integer	Example: `34`

Files

Use this API to manage repository files. You can also configure rate limits for this API.

Retrieve a file from a repository

`GET /api/v4/projects/{id}/repository/files/{file_path}`

Retrieves information about a specified file in a repository. This includes information like the name, size, and the file contents. File content is Base64 encoded.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`
`ref` Query, required	String	Commit SHA or name of a repository branch or tag Example: `main`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Create a file in a repository

`POST /api/v4/projects/{id}/repository/files/{file_path}`

Creates a file in a specified repository. Use the Commits API to create multiple files with a single request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The file content to be created (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Update a file in a repository

`PUT /api/v4/projects/{id}/repository/files/{file_path}`

Updates a specified file in a repository. Use the Commits API to update multiple files with a single request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The file content to be updated (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a file in a repository

`DELETE /api/v4/projects/{id}/repository/files/{file_path}`

Deletes a specified file in a repository. Use the Commits API to delete multiple files with a single request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`
`branch` Query, required	String	Name of the branch to commit the deletion into. Combine with `start_branch` to create a new branch Example: `main`
`commit_message` Query, required	String	Commit message Example: `Initial commit`
`start_branch` Query	String	Name of the base branch to create the branch from Example: `main`
`author_email` Query	String	Email address of the commit author Example: `johndoe@example.com`
`author_name` Query	String	Name of the commit author Example: `John Doe`
`last_commit_id` Query	String	Last known file commit ID Example: `2695effb5807a22ff3d138d593fd856244e155e7`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve file metadata

`HEAD /api/v4/projects/{id}/repository/files/{file_path}`

Retrieves metadata for a specified file in a repository.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`

Request body (`application/json`)

Property	Type	Description
`ref` Required	String	Commit SHA or name of a repository branch or tag Example: `main`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve file blame history from a repository

`GET /api/v4/projects/{id}/repository/files/{file_path}/blame`

Retrieves blame history for a specified file in a repository. Each blame range contains lines and their corresponding commit information.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`
`ref` Query, required	String	Commit SHA or name of a repository branch or tag Example: `main`
`range` Query	Object	Range of lines to return blame information for
`range[start]` Query, required	Integer	First line number of the range to blame Minimum: `1`
`range[end]` Query, required	Integer	Last line number of the range to blame Minimum: `1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBlameRange
`400`	Bad Request	—
`404`	Not Found	—

Retrieve file blame metadata

`HEAD /api/v4/projects/{id}/repository/files/{file_path}/blame`

Retrieves blame metadata for lines in a specified file.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`

Request body (`application/json`)

Property	Type	Description
`ref` Required	String	Commit SHA or name of a repository branch or tag Example: `main`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a raw file from a repository

`GET /api/v4/projects/{id}/repository/files/{file_path}/raw`

Retrieves the raw file contents for a specified file in a repository.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`file_path` Path, required	String	URL-encoded full path to the file, for example `lib%2Fclass%2Erb` Example: `lib%2Fclass%2Erb`
`ref` Query	String	Commit SHA or name of a repository branch or tag Example: `main`
`lfs` Query	Boolean	If `true`, returns the Git LFS file contents instead of the pointer. Ignored if the file isn't tracked by Git LFS Default: `false`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBlameRange

Property	Type	Description
`commit`	APIEntitiesBlameRangeCommit	—
`lines`	Array of strings	Example: `["lorem ipsum"]`

APIEntitiesBlameRangeCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`

Freeze periods

Use this API to interact with deployment freeze periods.

List all freeze periods

`GET /api/v4/projects/{id}/freeze_periods`

Lists all freeze periods for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesFreeze Period
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a freeze period

`POST /api/v4/projects/{id}/freeze_periods`

Creates a freeze period for a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`cron_timezone`	String	Time zone for the cron fields
`freeze_end` Required	String	End of the freeze period in cron format
`freeze_start` Required	String	Start of the freeze period in cron format

Responses

Code	Description	Schema
`201`	Created	APIEntitiesFreezePeriod
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve a freeze period

`GET /api/v4/projects/{id}/freeze_periods/{freeze_period_id}`

Retrieves a freeze period for a specified `freeze_period_id`. You must have the Reporter, Developer, Maintainer, or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`freeze_period_id` Path, required	Integer	ID of the freeze period

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFreezePeriod
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a freeze period

`PUT /api/v4/projects/{id}/freeze_periods/{freeze_period_id}`

Updates a freeze period for a specified `freeze_period_id`. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`freeze_period_id` Path, required	Integer	ID of the freeze period

Request body (`application/json`)

Property	Type	Description
`cron_timezone`	String	Time zone for the cron fields
`freeze_end`	String	End of the freeze period in cron format
`freeze_start`	String	Start of the freeze period in cron format

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFreezePeriod
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Delete a freeze period

`DELETE /api/v4/projects/{id}/freeze_periods/{freeze_period_id}`

Deletes a freeze period for a specified `freeze_period_id`. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`freeze_period_id` Path, required	Integer	ID of the freeze period

Responses

Code	Description	Schema
`200`	OK	APIEntitiesFreezePeriod
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesFreezePeriod

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-15T17:03:35.702Z`
`cron_timezone`	String	Example: `UTC`
`freeze_end`	String	Example: `0 8 * * 1`
`freeze_start`	String	Example: `0 23 * * 5`
`id`	Integer (int64)	Example: `1`
`updated_at`	String (date-time)	Example: `2020-05-15T17:03:35.702Z`

GLQL

Use this API to execute GitLab Query Language (GLQL) queries programmatically. GLQL provides a simplified query language to search and filter GitLab resources such as issues, merge requests, and epics across projects and groups.

The group or project must allow access to its data. For private groups and projects, you must authenticate with a personal access token that has the appropriate permissions.

Execute a GLQL query

`POST /api/v4/glql`

Executes a GLQL query to search and filter GitLab resources.

Request body (`application/json`)

Property	Type	Description
`after`	String	Cursor for pagination. Use the `data.pageInfo.endCursor` value from a previous query to fetch the next page of results
`glql_yaml` Required	String	GLQL query that may include YAML configuration. Maximum size is 10,000 bytes. See query formats for details

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGlqlResult
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`429`	Too Many Requests	—
`500`	Internal server error	—

Retrieve the GLQL schema

`GET /api/v4/glql/schema`

Retrieves the GLQL schema: data sources with their filter, display and sort fields, the operator, value kind and reference type vocabularies, the available functions, and the display types a query can be rendered as.

Responses

Code	Description	Schema
`200`	OK	—
`500`	Internal server error	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesGlqlData

Property	Type	Description
`count`	Integer	Number of found items Example: `42`
`nodes`	Array of arrays	The list of found items Example: `[]`
`pageInfo`	APIEntitiesGlqlPageInfo	Pagination information

APIEntitiesGlqlField

Property	Type	Description
`field`	String	Base field name. For aliased parameterised fields this is the underlying field (for example, `durationQuantile`), while `key` is the alias (for example, `p50`). For standard fields, same as `key` Example: `title`
`key`	String	Unique field key Example: `title`
`label`	String	Human-readable field label Example: `Title`
`name`	String	Underlying name of field, often the same as `key`, but it may be different if one type of field has multiple possible keys. Example `created` and `createdAt` Example: `title`
`parameters`	Object	Resolved parameter metadata for parameterised fields, absent when the field has no parameters Example: `{"granularity": "weekly"}`
`type`	String	Field classification. Either `dimension` or `metric` for analytics mode fields, absent for standard fields Example: `dimension`

APIEntitiesGlqlPageInfo

Property	Type	Description
`endCursor`	String	Cursor for the last item Example: `eyJpZCI6IjE3In0`
`hasNextPage`	Boolean	Whether there are more items Example: `true`
`hasPreviousPage`	Boolean	Whether there are previous items Example: `false`
`startCursor`	String	Cursor for the first item Example: `eyJpZCI6IjE3In0`

APIEntitiesGlqlResult

Property	Type	Description
`data`	APIEntitiesGlqlData	Query result data containing count, nodes, and pagination info
`error`	String	Error message if query failed
`fields`	Array of APIEntitiesGlqlField	Field definitions for the query results
`success`	Boolean	Example: `true`

Geo

Use this API to manage Geo sites. All of these endpoints require administrator access.

> The Geo Nodes endpoints are deprecated and are planned for removal in v5 of the API. Use the Geo Sites API instead. This change is a breaking change.

Returns a Geo proxy response

`GET /api/v4/geo/proxy`

Determines if the current Geo site should proxy requests.

Responses

Code	Description	Schema
`200`	OK	—
`403`	Forbidden	—

GitLab Pages

Use this API to administer and use GitLab Pages, including managing custom domains.

GitLab Pages must be enabled to use these endpoints.

List all Pages domains

`GET /api/v4/pages/domains`

Lists all Pages domains on the instance. You must have administrator access to the instance.

Parameters

Name	Type	Description
`domain` Query	String	Domain of the GitLab Pages site to filter on
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPagesDomainBasic
`400`	Bad Request	—

Retrieve Pages settings for a project

`GET /api/v4/projects/{id}/pages`

Retrieves Pages settings for a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Update Pages settings for a project

`PATCH /api/v4/projects/{id}/pages`

Updates Pages settings for a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`pages_https_only`	Boolean	If `true`, forces HTTPS
`pages_primary_domain`	String	Primary domain, from the existing assigned domains, to redirect all Pages requests to. Introduced in GitLab 17.8
`pages_unique_domain_enabled`	Boolean	If `true`, uses a unique domain

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Unpublish Pages

`DELETE /api/v4/projects/{id}/pages`

Unpublishes Pages from a specified project. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

List all Pages domains in a project

`GET /api/v4/projects/{id}/pages/domains`

Lists all Pages domains in a specified project. You must have permissions to view Pages domains.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPages Domain
`400`	Bad Request	—
`404`	Not Found	—

Create Pages domain

`POST /api/v4/projects/{id}/pages/domains`

Creates a Pages domain in a specified project. You must have permissions to create Pages domains.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`auto_ssl_enabled`	Boolean	If `true`, enables automatic generation of SSL certificates issued by Let's Encrypt for custom domains Default: `false`
`certificate`	String (binary) or string	Certificate in PEM format with intermediates following in most specific to least specific order
`domain` Required	String	Custom domain
`key`	String (binary) or string	Certificate key in PEM format
`user_provided_certificate`	String	All or none of `user_provided_certificate`, `user_provided_key` must be provided
`user_provided_key`	String	All or none of `user_provided_certificate`, `user_provided_key` must be provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPagesDomain
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a Pages domain

`GET /api/v4/projects/{id}/pages/domains/{domain}`

Retrieves a Pages domain from a specified project. You must have permissions to view Pages domains.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`domain` Path, required	String	Custom domain

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPagesDomain
`400`	Bad Request	—
`404`	Not Found	—

Update Pages domain

`PUT /api/v4/projects/{id}/pages/domains/{domain}`

Updates a specified Pages domain in a project. You must have permissions to change an existing Pages domain.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`domain` Path, required	String	Custom domain

Request body (`application/json`)

Property	Type	Description
`auto_ssl_enabled`	Boolean	If `true`, enables automatic generation of SSL certificates issued by Let's Encrypt for custom domains
`certificate`	String (binary) or string	Certificate in PEM format with intermediates following in most specific to least specific order
`key`	String (binary) or string	Certificate key in PEM format

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete Pages domain

`DELETE /api/v4/projects/{id}/pages/domains/{domain}`

Deletes a specified Pages domain in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`domain` Path, required	String	Custom domain

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Verify Pages domain

`PUT /api/v4/projects/{id}/pages/domains/{domain}/verify`

Verifies a specified Pages domain in a project. You must have permissions to update Pages domains.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`domain` Path, required	String	Custom domain to verify

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPagesDomain
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesPagesDomain

Property	Type	Description
`auto_ssl_enabled`	String	—
`certificate`	APIEntitiesPagesDomainCertificate	—
`domain`	String	—
`enabled_until`	String	—
`url`	String	—
`verification_code`	String	—
`verified`	String	—

APIEntitiesPagesDomainBasic

Property	Type	Description
`auto_ssl_enabled`	String	—
`certificate_expiration`	APIEntitiesPagesDomainCertificateExpiration	—
`domain`	String	—
`enabled_until`	String	—
`project_id`	String	—
`url`	String	—
`verification_code`	String	—
`verified`	String	—

APIEntitiesPagesDomainCertificate

Property	Type	Description
`certificate`	String	—
`certificate_text`	String	—
`expired`	String	—
`subject`	String	—

APIEntitiesPagesDomainCertificateExpiration

Property	Type	Description
`expiration`	String	—
`expired`	String	—

Group import and export

Use this API to migrate a group structure by exporting and importing a file. When you use it with the project import and export API, you can preserve relationships that span the group, such as connections between project issues and group epics. Run the group export and import first, then import the project exports into the group structure.

A group export includes group milestones, boards, labels, badges, members, events, wikis (Premium and Ultimate only), and subgroups. Each subgroup includes all of the same data.

To preserve the member list and permissions of an imported group, make sure those users exist on the destination instance before you import.

Because of issue 405168, imported groups have a `private` visibility level unless you import them into a parent group. By default, groups imported into a parent group inherit the visibility of the parent.

The destination instance uses the group relations export endpoints during group migration by direct transfer. You don't usually need to call them yourself. In this context, a relation is an exportable item such as an epic, including any related items such as a label. These endpoints require your instance to meet certain prerequisites, and can't be used with the file-based export and import endpoints.

Create a group import

`POST /api/v4/groups/import`

Creates a group import. The maximum import file size can be set by the Administrator on GitLab Self-Managed (defaults to `0` (unlimited)).

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The group export file to be imported
`name` Required	String	Group name
`organization_id`	Integer	The ID of the organization that the group will be part of
`parent_id`	Integer	The ID of the parent group that the group will be imported into. Defaults to the current user's namespace
`path` Required	String	Group path

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`429`	Too many requests	—
`503`	Service unavailable	—

Workhorse authorize the group import upload

`POST /api/v4/groups/import/authorize`

This feature was introduced in GitLab 12.8

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Create a group export

`POST /api/v4/groups/{id}/export`

Creates a group export for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—
`503`	Service unavailable	—

Retrieve a group export download

`GET /api/v4/groups/{id}/export/download`

Retrieves the exported archive for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Schedule a relations export for a group

`POST /api/v4/groups/{id}/export_relations`

Schedules a relations export for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Request body (`application/json`)

Property	Type	Description
`batched`	Boolean	Whether to export in batches

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Download a relations export for a group

`GET /api/v4/groups/{id}/export_relations/download`

Downloads a group relations export file.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`relation` Query, required	String	Group relation name
`batched` Query	Boolean	Whether to download in batches. All or none of `batched`, `batch_number` must be provided
`batch_number` Query	Integer	Batch number to download. All or none of `batched`, `batch_number` must be provided

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Retrieve the status of an relations export for a group

`GET /api/v4/groups/{id}/export_relations/status`

Retrieves the status of a relations export for a group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`relation` Query	String	Group relation name

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBulkImportsExportStatus
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBulkImportsExportBatchStatus

Property	Type	Description
`batch_number`	Integer	Example: `1`
`error`	String	Example: `Error message`
`objects_count`	Integer	Example: `100`
`status`	String	Allowed values: `started`, `finished`, `failed` Example: `started`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

APIEntitiesBulkImportsExportStatus

Property	Type	Description
`batched`	Boolean	Example: `true`
`batches`	APIEntitiesBulkImportsExportBatchStatus	—
`batches_count`	Integer	Example: `2`
`error`	String	Example: `Error message`
`relation`	String	Example: `issues`
`status`	String	Allowed values: `pending`, `started`, `finished`, `failed` Example: `started`
`total_objects_count`	Integer	Example: `100`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

Groups

Use this API to view and manage groups and the resources they contain, including:

- Group settings, members, and security settings.
- Group issues and issue statistics.
- Markdown uploads referenced in epics or wiki pages.
- Bulk reassignment of placeholder users after an import.

Endpoint responses might vary based on the permissions of the authenticated user in the group. If a user isn't a member of a private group, requests to that group return a `404 Not Found` status code.

For project members, use the project members API.

List all groups

`GET /api/v4/groups`

Lists all visible groups for the authenticated user. Unauthenticated requests return only public groups.

Parameters

Name	Type	Description
`statistics` Query	Boolean	If `true`, includes group statistics. Administrators only Default: `false`
`archived` Query	Boolean	If `true`, returns only archived groups or projects
`skip_groups` Query	Array of integers	Skip the specified group IDs
`all_available` Query	Boolean	If `true`, returns all accessible groups. If `false`, returns only groups where the user is a member. When omitted, administrators receive all groups and other authenticated users receive only their own. Unauthenticated requests always return all public groups. The `owned` and `min_access_level` attributes take precedence
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized groups matching the search criteria
`owned` Query	Boolean	If `true`, returns only groups explicitly owned by the current user Default: `false`
`order_by` Query	String	Sort results by the specified field Allowed values: `name`, `path`, `id`, `similarity` Default: `name`

Name	Type	Description
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`min_access_level` Query	Integer	Return groups where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`top_level_only` Query	Boolean	If `true`, returns only top-level groups, excluding subgroups
`marked_for_deletion_on` Query	String (date)	Return groups that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only groups that are not archived and not marked for deletion
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—

Create a group

`POST /api/v4/groups`

Creates a project group. Available only for users who can create groups.

Request body (`multipart/form-data`)

Property	Type	Description
`auto_devops_enabled`	Boolean	Default to Auto DevOps pipeline for all projects within this group
`avatar`	String (binary)	Avatar image for the group
`crm_enabled`	Boolean	Enable Customer Relations Management for this group
`default_branch`	String	The default branch of group's projects Example: `main`
`default_branch_protection`	Integer	Determine if developers can push to default branch Allowed values: `0`, `3`, `1`, `2`, `4`
`default_branch_protection_defaults`	Object	Determine if developers can push to default branch
`default_branch_protection_defaults.allow_force_push`	Boolean	Allow force push for all users with push access
`default_branch_protection_defaults.allowed_to_merge`	Array of objects	An array of access levels allowed to merge
`default_branch_protection_defaults.allowed_to_merge[].access_level` Required	Integer	A valid access level Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.allowed_to_push`	Array of objects	An array of access levels allowed to push
`default_branch_protection_defaults.allowed_to_push[].access_level` Required	Integer	A valid access level Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.code_owner_approval_required`	Boolean	Require approval from code owners

Property	Type	Description
`default_branch_protection_defaults.developer_can_initial_push`	Boolean	Allow developers to initial push
`description`	String	The description of the group
`emails_disabled`	Boolean	*(Deprecated)* Disable email notifications. Use: emails_enabled
`emails_enabled`	Boolean	Enable email notifications
`enabled_git_access_protocol`	String	Allow only the selected protocols to be used for Git access Allowed values: `ssh`, `http`, `all`
`lfs_enabled`	Boolean	Enable/disable LFS for the projects in this group
`lock_resource_access_token_notify_inherited`	Boolean	Prevent subgroups from overriding the access token expiry notification setting
`mentions_disabled`	Boolean	Disable a group from getting mentioned
`name` Required	String	The name of the group
`organization_id`	Integer	The organization id for the group
`parent_id`	Integer	The parent group id for creating nested group
`path` Required	String	The path of the group
`project_creation_level`	String	Determine if developers can create projects in the group Allowed values: `noone`, `owner`, `maintainer`, `developer`, `administrator`

Property	Type	Description
`request_access_enabled`	Boolean	Allow users to request member access
`require_two_factor_authentication`	Boolean	Require all users in this group to setup Two-factor authentication
`resource_access_token_notify_inherited`	Boolean	Send access token expiry notifications to all inherited members of the group
`share_with_group_lock`	Boolean	Prevent sharing a project with another group within this group
`show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`subgroup_creation_level`	String	Allowed to create subgroups Allowed values: `owner`, `maintainer`
`two_factor_grace_period`	Integer	Time before Two-factor authentication is enforced
`visibility`	String	The visibility of the group Allowed values: `private`, `internal`, `public`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroup
`400`	Bad Request	—

Retrieve a group

`GET /api/v4/groups/{id}`

Retrieves a specified group by ID or path.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only
`with_projects` Query	Boolean	If `true`, includes details of the projects that belong to the group. Deprecated and scheduled for removal in API v5. To list the projects in a group, use the List all projects in a group operation instead Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupDetail
`400`	Bad Request	—
`404`	Not Found	—

Update group attributes

`PUT /api/v4/groups/{id}`

Updates the attributes for a specified group. You must be an administrator or have the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`multipart/form-data`)

Property	Type	Description
`auto_devops_enabled`	Boolean	Default to Auto DevOps pipeline for all projects within this group
`avatar`	String (binary)	Avatar image for the group
`crm_enabled`	Boolean	Enable Customer Relations Management for this group
`default_branch`	String	The default branch of group's projects Example: `main`
`default_branch_protection`	Integer	Determine if developers can push to default branch Allowed values: `0`, `3`, `1`, `2`, `4`
`default_branch_protection_defaults`	Object	Determine if developers can push to default branch
`default_branch_protection_defaults.allow_force_push`	Boolean	Allow force push for all users with push access
`default_branch_protection_defaults.allowed_to_merge`	Array of objects	An array of access levels allowed to merge
`default_branch_protection_defaults.allowed_to_merge[].access_level` Required	Integer	A valid access level Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.allowed_to_push`	Array of objects	An array of access levels allowed to push
`default_branch_protection_defaults.allowed_to_push[].access_level` Required	Integer	A valid access level Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.code_owner_approval_required`	Boolean	Require approval from code owners

Property	Type	Description
`default_branch_protection_defaults.developer_can_initial_push`	Boolean	Allow developers to initial push
`description`	String	The description of the group
`emails_disabled`	Boolean	*(Deprecated)* Disable email notifications. Use: emails_enabled
`emails_enabled`	Boolean	Enable email notifications
`enabled_git_access_protocol`	String	Allow only the selected protocols to be used for Git access Allowed values: `ssh`, `http`, `all`
`lfs_enabled`	Boolean	Enable/disable LFS for the projects in this group
`lock_math_rendering_limits_enabled`	Boolean	Indicates if math rendering limits are locked for all descendent groups
`lock_require_sha_for_merge`	Boolean	Enforce the require_sha_for_merge setting for all descendant groups
`lock_resource_access_token_notify_inherited`	Boolean	Prevent subgroups from overriding the access token expiry notification setting
`math_rendering_limits_enabled`	Boolean	Indicates if math rendering limits are used for this group
`max_artifacts_size`	Integer	Set the maximum file size for each job’s artifacts
`mentions_disabled`	Boolean	Disable a group from getting mentioned
`name`	String	The name of the group
`path`	String	The path of the group

Property	Type	Description
`prevent_sharing_groups_outside_hierarchy`	Boolean	Prevent sharing groups within this namespace with any groups outside the namespace. Only available on top-level groups
`project_creation_level`	String	Determine if developers can create projects in the group Allowed values: `noone`, `owner`, `maintainer`, `developer`, `administrator`
`request_access_enabled`	Boolean	Allow users to request member access
`require_sha_for_merge`	Boolean	Indicates whether merge requests in this group require a valid commit SHA for calls to the merge a merge request endpoint
`require_two_factor_authentication`	Boolean	Require all users in this group to setup Two-factor authentication
`resource_access_token_notify_inherited`	Boolean	Send access token expiry notifications to all inherited members of the group
`share_with_group_lock`	Boolean	Prevent sharing a project with another group within this group
`shared_runners_setting`	String	Enable/disable shared runners for the group and its subgroups and projects Allowed values: `disabled_and_unoverridable`, `disabled_and_overridable`, `enabled`
`show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails

Property	Type	Description
`step_up_auth_required_oauth_provider`	String	OAuth provider required for step-up authentication. Pass empty string to disable
`subgroup_creation_level`	String	Allowed to create subgroups Allowed values: `owner`, `maintainer`
`two_factor_grace_period`	Integer	Time before Two-factor authentication is enforced
`visibility`	String	The visibility of the group Allowed values: `private`, `internal`, `public`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

Schedule a group for deletion

`DELETE /api/v4/groups/{id}`

Schedules a group for deletion. Groups are deleted at the end of the retention period (30 days by default). Use the `permanently_remove` param to override the retention period.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`404`	Not Found	—

Archive a group

`POST /api/v4/groups/{id}/archive`

Archives a specified group. You must be an administrator or have the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroup
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all descendant groups

`GET /api/v4/groups/{id}/descendant_groups`

Lists all descendant groups for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`statistics` Query	Boolean	If `true`, includes group statistics. Administrators only Default: `false`
`archived` Query	Boolean	If `true`, returns only archived groups or projects
`skip_groups` Query	Array of integers	Skip the specified group IDs
`all_available` Query	Boolean	If `true`, returns all accessible groups. If `false`, returns only groups where the user is a member. When omitted, administrators receive all groups and other authenticated users receive only their own. Unauthenticated requests always return all public groups. The `owned` and `min_access_level` attributes take precedence
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized groups matching the search criteria
`owned` Query	Boolean	If `true`, returns only groups explicitly owned by the current user Default: `false`
`order_by` Query	String	Sort results by the specified field Allowed values: `name`, `path`,

Name	Type	Description
		`id`, `similarity` Default: `name`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`min_access_level` Query	Integer	Return groups where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`top_level_only` Query	Boolean	If `true`, returns only top-level groups, excluding subgroups
`marked_for_deletion_on` Query	String (date)	Return groups that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only groups that are not archived and not marked for deletion
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key

Name	Type	Description
		and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

List all shared groups

`GET /api/v4/groups/{id}/groups/shared`

Lists all groups shared with a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`skip_groups` Query	Array of integers	Skip the specified group IDs
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized groups matching the search criteria
`min_access_level` Query	Integer	Return groups where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`order_by` Query	String	Sort results by the specified field Allowed values: `name`, `path`, `id`, `similarity` Default: `name`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Name	Type	Description
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

List all invited groups

`GET /api/v4/groups/{id}/invited_groups`

Lists all groups invited to a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`relation` Query	Array of strings	Filter groups by relation
`search` Query	String	Return the list of authorized groups matching the search criteria
`min_access_level` Query	Integer	Return groups where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

List all issues for a group

`GET /api/v4/groups/{id}/issues`

Lists all issues for a specified group. If the group is private, you must provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`state` Query	String	Filter issues by state Allowed values: `opened`, `closed`, `all` Default: `all`
`closed_by_id` Query	Integer	Return issues closed by the user with the given ID
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `due_date`, `label_priority`, `milestone_due`, `popularity`, `priority`, `relative_position`, `title`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`due_date` Query	String	Return issues that have no due date, are overdue, or whose due date is this week, this month, or between two weeks ago and next month Allowed values: `0`, `any`, `today`, `tomorrow`, `overdue`, `week`, `month`, `next_month_and_previous_two_weeks`, ``

Name	Type	Description
`issue_type` Query	String	The type of the issue. Accepts: issue, incident, test_case, requirement, task, ticket Allowed values: `issue`, `incident`, `test_case`, `requirement`, `task`, `ticket`
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for

Name	Type	Description
		`Upcoming` and `Started` differs from the logic used in the GraphQL API. Mutually exclusive with `milestone` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid

Name	Type	Description
		parameter error is returned. When multiple usernames are given, only issues assigned to all of them are returned. Mutually exclusive with `assignee_id`
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`

Name	Type	Description
`not[author_username]` Query	String	Return issues not authored by the user with the given username. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer	Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `all`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Name	Type	Description
`non_archived` Query	Boolean	If `true`, returns only issues from non-archived projects. If `false`, returns issues from both archived and non-archived projects Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Retrieve issues statistics for a group

`GET /api/v4/groups/{id}/issues_statistics`

Retrieves statistics for issues in a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for `Upcoming` and `Started` differs from the logic used in the GraphQL API. Mutually exclusive with `milestone`

Name	Type	Description
		Allowed values: `Any`, `None`, `Upcoming`, `Started`
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid parameter error is returned. When multiple usernames are given, only issues assigned to all of them are returned.

Name	Type	Description
		Mutually exclusive with `assignee_id`
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	Return issues not authored by the user with the given

Name	Type	Description
		username. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer	Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `all`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve pending reassignments

`GET /api/v4/groups/{id}/placeholder_reassignments`

Retrieves a CSV file with a list of pending reassignments. This feature was introduced in GitLab 17.10.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	String (`text/csv`)
`400`	Bad Request	—
`404`	Not Found	—

Reassign placeholders

`POST /api/v4/groups/{id}/placeholder_reassignments`

Reassigns placeholder users with an uploaded CSV file.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The CSV file containing the reassignments

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Workhorse authorization for the reassignment CSV file

`POST /api/v4/groups/{id}/placeholder_reassignments/authorize`

Authorizes Workhorse to handle CSV file uploads for placeholder reassignments. This feature was introduced in GitLab 17.10

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

List all projects in a group

`GET /api/v4/groups/{id}/projects`

Lists all projects in a specified group accessible to the authenticated user. Unauthenticated requests return only public projects with a limited subset of attributes.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`active` Query	Boolean	If `true`, returns active projects. If `false`, returns projects that are archived or marked for deletion. Introduced in GitLab 18.8
`archived` Query	Boolean	If `true`, returns only archived groups or projects
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized projects matching the search criteria
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`owned` Query	Boolean	If `true`, returns only groups explicitly owned by the current

Name	Type	Description
		user Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`with_shared` Query	Boolean	If `true`, includes projects shared with the group Default: `true`
`include_subgroups` Query	Boolean	If `true`, includes projects in subgroups of this group Default: `false`
`include_ancestor_groups` Query	Boolean	If `true`, includes projects in the ancestors of this group Default: `false`
`min_access_level` Query	Integer	Return projects where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProject
`400`	Bad Request	—
`404`	Not Found	—

List all shared projects

`GET /api/v4/groups/{id}/projects/shared`

Lists all projects shared with a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`archived` Query	Boolean	If `true`, returns only archived groups or projects
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized projects matching the search criteria
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `star_count` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`

Name	Type	Description
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`min_access_level` Query	Integer	Return projects where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProject
`400`	Bad Request	—
`404`	Not Found	—

Transfer a project to a group

`POST /api/v4/groups/{id}/projects/{project_id}`

Transfers a specified project to another group. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`project_id` Path, required	String	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupDetail
`400`	Bad Request	—
`404`	Not Found	—

Restore a group

`POST /api/v4/groups/{id}/restore`

Restores a specified group that was previously scheduled for deletion. Can not restore groups outside of the retention period (30 days by default).

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

List all runners in a group

`GET /api/v4/groups/{id}/runners`

Lists all runners available in a specified group and any ancestor groups, including any allowed instance runners.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`403`	Forbidden	—
`404`	Not Found	—

Reset runner registration token

`POST /api/v4/groups/{id}/runners/reset_registration_token`

FE version

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Group Not Found	—

Add a group to a group

`POST /api/v4/groups/{id}/share`

Adds a group to a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the share expires
`group_access` Required	Integer	Access level to grant to the invited group. For a list of values, see roles Allowed values: `10` , `15` , `20` , `25` , `30` , `40` , `50`
`group_id` Required	Integer	ID of the group to invite
`member_role_id`	Integer	ID of a custom role to assign to the invited group. If set, `group_access` must match the base role used to create the custom role

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupDetail
`400`	Bad Request	—
`404`	Not Found	—

Remove a group from a group

`DELETE /api/v4/groups/{id}/share/{group_id}`

Removes a group from a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`group_id` Path, required	Integer	ID of the shared group

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Remove a shared project from a group

`DELETE /api/v4/groups/{id}/shared_projects/{project_id}`

Removes a shared project targeting this group. The group Owner can remove a shared project without access to the source project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`project_id` Path, required	Integer	ID of the shared project

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Forbidden	—
`404`	Not found	—

List all subgroups

`GET /api/v4/groups/{id}/subgroups`

Lists all subgroups for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`statistics` Query	Boolean	If `true`, includes group statistics. Administrators only Default: `false`
`archived` Query	Boolean	If `true`, returns only archived groups or projects
`skip_groups` Query	Array of integers	Skip the specified group IDs
`all_available` Query	Boolean	If `true`, returns all accessible groups. If `false`, returns only groups where the user is a member. When omitted, administrators receive all groups and other authenticated users receive only their own. Unauthenticated requests always return all public groups. The `owned` and `min_access_level` attributes take precedence
`visibility` Query	String	Return groups with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return the list of authorized groups matching the search criteria
`owned` Query	Boolean	If `true`, returns only groups explicitly owned by the current user Default: `false`
`order_by` Query	String	Sort results by the specified field Allowed values: `name`, `path`,

Name	Type	Description
		`id`, `similarity` Default: `name`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`min_access_level` Query	Integer	Return groups where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`top_level_only` Query	Boolean	If `true`, returns only top-level groups, excluding subgroups
`marked_for_deletion_on` Query	String (date)	Return groups that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only groups that are not archived and not marked for deletion
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key

Name	Type	Description
		and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

Transfer a group

`POST /api/v4/groups/{id}/transfer`

Transfers a group to another parent group or transforms a subgroup into a top-level group. You must be an administrator or have the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`group_id`	Integer	ID of the new parent group. If omitted, the group becomes a top-level group

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

List all transfer locations for a group

`GET /api/v4/groups/{id}/transfer_locations`

Lists all groups that a specified source group can be transferred to.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`search` Query	String	Group name to search for
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGroup
`400`	Bad Request	—
`404`	Not Found	—

Transfer a group to an organization

`POST /api/v4/groups/{id}/transfer_to_organization`

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`organization_id` Required	Integer	ID of the organization to transfer the group to

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupDetail
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Group or Organization not found	—
`422`	Unprocessable entity	—

Unarchive a group

`POST /api/v4/groups/{id}/unarchive`

Unarchives a specified group. You must be an administrator or have the Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroup
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all uploads for a group

`GET /api/v4/groups/{id}/uploads`

Lists all uploads for a specified group sorted by `created_at` in descending order. You must have the Maintainer or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMarkdown UploadAdmin
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Upload a file to a group

`POST /api/v4/groups/{id}/uploads`

Uploads a file to the specified group. Returns a markdown-formatted link to the file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The file to upload

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupUpload
`400`	Bad request	—
`404`	Not found	—

Workhorse authorize the file upload

`POST /api/v4/groups/{id}/uploads/authorize`

This feature was introduced in GitLab 19.0

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not found	—

Download an uploaded file by secret and filename

`GET /api/v4/groups/{id}/uploads/{secret}/{filename}`

Downloads an uploaded file with a specified secret and filename. You must have the Guest, Planner, Reporter, Developer, Maintainer, or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`secret` Path, required	String	Secret of the upload
`filename` Path, required	String	Filename of the upload

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete an uploaded file by secret and filename

`DELETE /api/v4/groups/{id}/uploads/{secret}/{filename}`

Deletes an uploaded file with a specified secret and filename. You must have the Maintainer or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`secret` Path, required	String	Secret of the upload
`filename` Path, required	String	Filename of the upload

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Download an uploaded file by ID

`GET /api/v4/groups/{id}/uploads/{upload_id}`

Downloads an uploaded file with a specified ID. You must have the Maintainer or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`upload_id` Path, required	Integer	ID of the upload

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete an uploaded file by ID

`DELETE /api/v4/groups/{id}/uploads/{upload_id}`

Deletes an uploaded file with a specified ID. You must have the Maintainer or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`upload_id` Path, required	Integer	ID of the upload

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Get the users list of a group

`GET /api/v4/groups/{id}/users`

Parameters

Name	Type	Description
`search` Query	String	Return list of users matching the search criteria Example: `user`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

List all groups available to invite to a project

`GET /api/v4/projects/{id}/share_locations`

Lists all groups that can be invited to a project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project
`search` Query	String	Return groups matching the search criteria

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPublicGroupDetails
`400`	Bad Request	—
`404`	Not Found	—

Reset the runner registration token for the instance

`POST /api/v4/runners/reset_registration_token`

Resets the runner registration token for the GitLab instance.

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`403`	Forbidden	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCiResetTokenResult

Property	Type	Description
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesContainerExpirationPolicy

Property	Type	Description
`cadence`	String	—
`enabled`	String	—
`keep_n`	String	—
`name_regex`	String	—
`name_regex_keep`	String	—
`next_run_at`	String	—
`older_than`	String	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesGroup

Property	Type	Description
`archived`	Boolean	—
`auto_devops_enabled`	String	—
`avatar_url`	String	—
`created_at`	String	—
`crm_enabled`	Boolean	—
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	—
`default_branch_protection`	Integer	—
`default_branch_protection_defaults`	String	—
`description`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`full_name`	String	—
`full_path`	String	—
`id`	Integer (int64)	—
`lfs_enabled`	Boolean	—
`lock_math_rendering_limits_enabled`	Boolean	—
`lock_resource_access_token_notify_inherited`	Boolean	—
`marked_for_deletion_on`	String	—
`math_rendering_limits_enabled`	Boolean	—
`max_artifacts_size`	Integer	—
`mentions_disabled`	String	—

Property	Type	Description
`name`	String	Example: `Diaspora`
`organization_id`	Integer (int64)	—
`parent_id`	String	—
`path`	String	—
`project_creation_level`	String	—
`request_access_enabled`	Boolean	—
`require_two_factor_authentication`	Boolean	—
`resource_access_token_notify_inherited`	Boolean	—
`root_storage_statistics`	APIEntitiesNamespaceRootStorageStatistics	—
`share_with_group_lock`	Boolean	—
`shared_runners_setting`	String	—
`show_diff_preview_in_email`	Boolean	—
`statistics`	Object	—
`statistics.job_artifacts_size`	String	—
`statistics.lfs_objects_size`	String	—
`statistics.packages_size`	String	—
`statistics.pipeline_artifacts_size`	String	—
`statistics.repository_size`	String	—
`statistics.snippets_size`	String	—
`statistics.storage_size`	String	—
`statistics.uploads_size`	String	—

Property	Type	Description
`statistics.wiki_size`	String	—
`subgroup_creation_level`	String	—
`two_factor_grace_period`	Integer	—
`visibility`	String	—
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesGroupDetail

Property	Type	Description
`archived`	Boolean	—
`auto_devops_enabled`	String	—
`avatar_url`	String	—
`created_at`	String	—
`crm_enabled`	Boolean	—
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	—
`default_branch_protection`	Integer	—
`default_branch_protection_defaults`	String	—
`description`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`enabled_git_access_protocol`	String	Example: `ssh`
`full_name`	String	—
`full_path`	String	—
`id`	Integer (int64)	—
`lfs_enabled`	Boolean	—
`lock_math_rendering_limits_enabled`	Boolean	—
`lock_require_sha_for_merge`	Boolean	—
`lock_resource_access_token_notify_inherited`	Boolean	—
`marked_for_deletion_on`	String	—
`math_rendering_limits_enabled`	Boolean	—

Property	Type	Description
`max_artifacts_size`	Integer	—
`mentions_disabled`	String	—
`name`	String	Example: `Diaspora`
`organization_id`	Integer (int64)	—
`parent_id`	String	—
`path`	String	—
`prevent_sharing_groups_outside_hierarchy`	Boolean	—
`project_creation_level`	String	—
`projects`	APIEntitiesProject	—
`request_access_enabled`	Boolean	—
`require_sha_for_merge`	Boolean	—
`require_two_factor_authentication`	Boolean	—
`resource_access_token_notify_inherited`	Boolean	—
`root_storage_statistics`	APIEntitiesNamespaceRootStorageStatistics	—
`runners_token`	String	Example: `b8bc4a7a29eb76ea83cf79e4908c2b`
`share_with_group_lock`	Boolean	—
`shared_projects`	APIEntitiesProject	—
`shared_runners_setting`	String	—
`shared_with_groups`	Array of objects	—
`show_diff_preview_in_email`	Boolean	—
`statistics`	Object	—

Property	Type	Description
`statistics.job_artifacts_size`	String	—
`statistics.lfs_objects_size`	String	—
`statistics.packages_size`	String	—
`statistics.pipeline_artifacts_size`	String	—
`statistics.repository_size`	String	—
`statistics.snippets_size`	String	—
`statistics.storage_size`	String	—
`statistics.uploads_size`	String	—
`statistics.wiki_size`	String	—
`step_up_auth_required_oauth_provider`	String	OAuth provider required for step-up authentication
`subgroup_creation_level`	String	—
`two_factor_grace_period`	Integer	—
`visibility`	String	—
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesGroupUpload

Property	Type	Description
`alt`	String	The name of the file
`full_path`	String	The full path to the file
`id`	Integer (int64)	The ID of the file
`markdown`	String	A markdown-formatted link to the file
`url`	String	The URL to access the file

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesIssue

Property	Type	Description
`_links`	Object	—
`_links.award_emoji`	String	Example: `http://example.com/api/v4/projects/1/issues/2/award_emoji`
`_links.closed_as_duplicate_of`	String	Example: `http://example.com/api/v4/projects/1/issues/75`
`_links.notes`	String	Example: `http://example.com/api/v4/projects/1/issues/2/notes`
`_links.project`	String	Example: `http://example.com/api/v4/projects/1`
`_links.self`	String	Example: `http://example.com/api/v4/projects/1/issues/2`
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`

Property	Type	Description
`has_tasks`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`moved_to_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `4`
`references`	APIEntitiesIssuableReferences	—
`service_desk_reply_to`	String	Example: `user@example.com`
`severity`	String	The severity of the issue Allowed values: `UNKNOWN`, `LOW`, `MEDIUM`, `HIGH`, `CRITICAL`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`subscribed`	Boolean	Example: `false`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`task_status`	String	Example: `2 of 4 tasks completed`
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`

Property	Type	Description
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesMarkdownUploadAdmin

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`filename`	String	Example: `image.png`
`id`	Integer (int64)	Example: `1`
`size`	Integer	Example: `1024`
`uploaded_by`	APIEntitiesUserSafe	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesNamespaceRootStorageStatistics

Property	Type	Description
`build_artifacts_size`	Integer	CI artifacts size in bytes
`container_registry_size`	Integer	container registry size in bytes
`container_registry_size_is_estimated`	Boolean	Indicates whether the deduplicated container registry size for the namespace is an estimated value or not
`dependency_proxy_size`	Integer	Dependency Proxy sizes in bytes
`lfs_objects_size`	Integer	LFS objects size in bytes
`packages_size`	Integer	Packages size in bytes
`pipeline_artifacts_size`	Integer	CI pipeline artifacts size in bytes
`repository_size`	Integer	Git repository size in bytes
`snippets_size`	Integer	Snippets size in bytes
`storage_size`	Integer	Total storage in bytes
`uploads_size`	Integer	Uploads size in bytes
`wiki_size`	Integer	Wiki size in bytes

APIEntitiesProject

Property	Type	Description
`_links`	Object	—
`_links.cluster_agents`	String	Example: `https://gitlab.example.com/api/v4/projects/4/cluster_agents`
`_links.events`	String	Example: `https://gitlab.example.com/api/v4/projects/4/events`
`_links.issues`	String	Example: `https://gitlab.example.com/api/v4/projects/4/issues`
`_links.labels`	String	Example: `https://gitlab.example.com/api/v4/projects/4/labels`
`_links.members`	String	Example: `https://gitlab.example.com/api/v4/projects/4/members`
`_links.merge_requests`	String	Example: `https://gitlab.example.com/api/v4/projects/4/merge_requests`
`_links.repo_branches`	String	Example: `https://gitlab.example.com/api/v4/projects/4/repository/branches`
`_links.self`	String	Example: `https://gitlab.example.com/api/v4/projects/4`
`allow_merge_on_skipped_pipeline`	Boolean	—
`allow_pipeline_trigger_approve_deployment`	Boolean	—
`analytics_access_level`	String	Example: `enabled`
`archived`	Boolean	—
	String	Example: `enabled`

Property	Type	Description
`auto_cancel_pending_pipelines`		
`auto_devops_deploy_strategy`	String	Example: `continuous`
`auto_devops_enabled`	Boolean	—
`autoclose_referenced_issues`	Boolean	—
`automatic_rebase_enabled`	Boolean	—
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`build_git_strategy`	String	Example: `fetch`
`build_timeout`	Integer	Example: `3600`
`builds_access_level`	String	Example: `enabled`
`can_create_merge_request_in`	Boolean	—
`ci_allow_fork_pipelines_to_run_in_parent_project`	Boolean	—
`ci_config_path`	String	Example: ``
`ci_default_git_depth`	Integer	Example: `20`
`ci_delete_pipelines_in_seconds`	Integer	Example: `86400`
`ci_display_pipeline_variables`	Boolean	—
`ci_forward_deployment_enabled`	Boolean	—
`ci_forward_deployment_rollback_allowed`	Boolean	—
`ci_id_token_sub_claim_components`	Array of strings	—

Property	Type	Description
`ci_job_token_scope_enabled`	Boolean	—
`ci_pipeline_variables_minimum_override_role`	String	—
`ci_push_repository_for_job_token_allowed`	Boolean	—
`ci_separated_caches`	Boolean	—
`ci_skip_branch_pipelines_for_mrs`	Boolean	—
`container_expiration_policy`	APIEntitiesContainerExpirationPolicy	—
`container_registry_access_level`	String	Example: `enabled`
`container_registry_enabled`	Boolean	—
`container_registry_image_prefix`	String	Example: `registry.gitlab.example.com/gitlab/gitlab-client`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`creator_id`	Integer (int64)	Example: `1`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`description_html`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`empty_repo`	Boolean	—
`enforce_auth_checks_on_uploads`	Boolean	—

Property	Type	Description
`environments_access_level`	String	Example: `enabled`
`fe_project_setting_attributes`	Object	—
`feature_flags_access_level`	String	Example: `enabled`
`forked_from_project`	APIEntitiesBasicProjectDetails	—
`forking_access_level`	String	Example: `enabled`
`forks_count`	Integer	Example: `1`
`group_runners_enabled`	Boolean	—
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`import_error`	String	Example: `Import error`
`import_status`	String	Example: `none`
`import_type`	String	Example: `git`
`import_url`	String	Example: `https://gitlab.com/gitlab/gitlab.git`
`infrastructure_access_level`	String	Example: `enabled`
`issue_branch_template`	String	Example: `%(title)`
`issues_access_level`	String	Example: `enabled`
`issues_enabled`	Boolean	—
`jaga_task_required_to_merge`	Boolean	—
`jobs_enabled`	Boolean	—
`keep_latest_artifact`	Boolean	—
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`

Property	Type	Description
`lfs_enabled`	Boolean	—
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`marked_for_deletion_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`marked_for_deletion_on`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`max_artifacts_size`	Integer	—
`max_pipelines_per_merge_train`	Integer	—
`merge_commit_template`	String	Example: `%(title)`
`merge_method`	String	Example: `merge`
`merge_pipelines_enabled`	Boolean	—
`merge_request_title_regex`	String	—
`merge_request_title_regex_description`	String	—
`merge_requests_access_level`	String	Example: `enabled`
`merge_requests_enabled`	Boolean	—
`merge_train_enforced`	Boolean	—
`merge_trains_enabled`	Boolean	—
`merge_trains_skip_train_allowed`	Boolean	—
`model_experiments_access_level`	String	Example: `enabled`
`model_registry_access_level`	String	Example: `enabled`

Property	Type	Description
`monitor_access_level`	String	Example: `enabled`
`mr_default_target_self`	Boolean	—
`mr_default_title_template`	String	Example: `%(source_branch)`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	—
`only_allow_merge_if_pipeline_succeeds`	Boolean	—
`open_issues_count`	Integer	Example: `1`
`owner`	APIEntitiesUserBasic	—
`package_registry_access_level`	String	Example: `enabled`
`packages_enabled`	Boolean	—
`pages_access_level`	String	Example: `enabled`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`printing_merge_request_link_enabled`	Boolean	—
`protect_merge_request_pipelines`	Boolean	—
`public_jobs`	Boolean	—
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`releases_access_level`	String	Example: `enabled`
`remove_source_branch_after_merge`	Boolean	—
`repository_access_level`	String	Example: `enabled`
`repository_object_format`	String	Example: `sha1`
`repository_storage`	String	Example: `default`
`request_access_enabled`	Boolean	—
`resolve_outdated_diff_discussions`	Boolean	—
`resource_group_default_process_mode`	String	Example: `unordered`
`restrict_user_defined_variables`	Boolean	—
`runner_token_expiration_interval`	Integer	Example: `3600`
`runners_token`	String	Example: `b8547b1dc37721d05889db52fa2f02`
`security_and_compliance_access_level`	String	Example: `enabled`
`service_desk_address`	String	Example: `address@example.com`
`service_desk_enabled`	Boolean	—
`shared_runners_enabled`	Boolean	—
`shared_with_groups`	Array of objects	—
`show_diff_preview_in_email`	Boolean	—
`snippets_access_level`	String	Example: `enabled`
`snippets_enabled`	Boolean	—

Property	Type	Description
`squash_commit_template`	String	Example: `%(source_branch)`
`squash_option`	String	Example: `default_off`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`statistics`	APIEntitiesProjectStatistics	—
`suggestion_commit_message`	String	Example: `Suggestion message`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`updated_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`visibility`	String	Example: `public`
`warn_about_potentially_unwanted_characters`	Boolean	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`
`wiki_access_level`	String	Example: `enabled`
`wiki_enabled`	Boolean	—

APIEntitiesProjectStatistics

Property	Type	Description
`commit_count`	Integer	Example: `37`
`container_registry_size`	Integer	Example: `0`
`job_artifacts_size`	Integer	Example: `0`
`lfs_objects_size`	Integer	Example: `0`
`packages_size`	Integer	Example: `0`
`pipeline_artifacts_size`	Integer	Example: `0`
`repository_size`	Integer	Example: `1.03809e+06`
`snippets_size`	Integer	Example: `0`
`storage_size`	Integer	Example: `1.03809e+06`
`uploads_size`	Integer	Example: `0`
`wiki_size`	Integer	Example: `0`

APIEntitiesPublicGroupDetails

Property	Type	Description
`avatar_url`	String	Example: `http://gitlab.example.com/uploads/group/avatar/1/avatar.png`
`full_name`	String	Example: `Foobar Group`
`full_path`	String	Example: `foo-bar`
`id`	Integer (int64)	—
`name`	String	Example: `Diaspora`
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserSafe

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`username`	String	Example: `admin`

Hooks

Use this API to manage the hooks that GitLab uses to notify external services about events:

- Project webhooks are limited to a single project.
- Group webhooks apply to all projects and subgroups in a group.
- System hooks apply to the entire instance.

You can also test a hook, inspect its event log, resend an event, and manage its custom headers and URL variables.

Managing project webhooks requires administrator access or the Maintainer or Owner role for the project. Managing group webhooks requires administrator access or the Owner role for the group. Managing system hooks requires administrator access.

List all system hooks

`GET /api/v4/hooks`

Lists all system hooks for the instance.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesHook
`400`	Bad Request	—

Create a system hook

`POST /api/v4/hooks`

Creates a system hook.

Request body (`application/json`)

Property	Type	Description
`branch_filter_strategy`	String	Filter push events by branch. Defaults to `wildcard` Allowed values: `wildcard`, `regex`, `all_branches`
`custom_headers`	Array of objects	Custom headers for the hook
`custom_headers[].key` Required	String	Name of the header Example: `X-Custom-Header`
`custom_headers[].value` Required	String	Value of the header Example: `value`
`custom_webhook_template`	String	Custom template for the system hook request payload
`description`	String	Description of the hook. Introduced in GitLab 17.1
`enable_ssl_verification`	Boolean	If `true`, verifies the SSL certificate when the hook is triggered
`merge_requests_events`	Boolean	If `true`, triggers the system hook on merge request events
`name`	String	Name of the hook. Introduced in GitLab 17.1
`push_events`	Boolean	If `true`, triggers the system hook on push events
`push_events_branch_filter`	String	Filter push events by branch name
`repository_update_events`	Boolean	If `true`, triggers the system hook on repository update events
`signing_token`	String	HMAC signing token used to compute the `webhook-signature` header. Must be in `whsec_<base64>` format encoding a 32-byte key, and is

Property	Type	Description
		not returned in the response. Introduced in GitLab 19.0
`tag_push_events`	Boolean	If `true`, triggers the system hook on tag push events
`token`	String	Secret token used to validate received payloads. Not returned in the response, and changing the hook URL resets it
`url` Required	String	Webhook URL Example: `http://example.com/hook`
`url_variables`	Array of objects	URL variables for interpolation
`url_variables[].key` Required	String	Name of the variable Example: `token`
`url_variables[].value` Required	String	Value of the variable Example: `123`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesHook
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve a system hook

`GET /api/v4/hooks/{hook_id}`

Retrieves a specified system hook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the system hook

Responses

Code	Description	Schema
`200`	OK	APIEntitiesHook
`400`	Bad Request	—
`404`	Not found	—

Create a test run

`POST /api/v4/hooks/{hook_id}`

Creates a test run for a webhook. Executes the webhook with mock data.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the system hook

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Update a system hook

`PUT /api/v4/hooks/{hook_id}`

Updates a specified system hook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the system hook

Request body (`application/json`)

Property	Type	Description
`branch_filter_strategy`	String	Filter push events by branch. Defaults to `wildcard` Allowed values: `wildcard`, `regex`, `all_branches`
`custom_headers`	Array of objects	Custom headers for the hook
`custom_headers[].key` Required	String	Name of the header Example: `X-Custom-Header`
`custom_headers[].value` Required	String	Value of the header Example: `value`
`custom_webhook_template`	String	Custom template for the system hook request payload
`description`	String	Description of the hook. Introduced in GitLab 17.1
`enable_ssl_verification`	Boolean	If `true`, verifies the SSL certificate when the hook is triggered
`merge_requests_events`	Boolean	If `true`, triggers the system hook on merge request events
`name`	String	Name of the hook. Introduced in GitLab 17.1
`push_events`	Boolean	If `true`, triggers the system hook on push events
`push_events_branch_filter`	String	Filter push events by branch name
`repository_update_events`	Boolean	If `true`, triggers the system hook on repository update events
`signing_token`	String	HMAC signing token used to compute the `webhook-signature` header. Must be in `whsec_<base64>` format encoding a 32-byte key, and is

Property	Type	Description
		not returned in the response. Introduced in GitLab 19.0
`tag_push_events`	Boolean	If `true`, triggers the system hook on tag push events
`token`	String	Secret token used to validate received payloads. Not returned in the response, and changing the hook URL resets it
`url`	String	Webhook URL
`url_variables`	Array of objects	URL variables for interpolation
`url_variables[].key` Required	String	Name of the variable Example: `token`
`url_variables[].value` Required	String	Value of the variable Example: `123`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesHook
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a system hook

`DELETE /api/v4/hooks/{hook_id}`

Deletes a specified system hook. Administrators only.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the system hook

Responses

Code	Description	Schema
`200`	OK	APIEntitiesHook
`400`	Bad Request	—
`404`	Not found	—

Update a custom header

`PUT /api/v4/hooks/{hook_id}/custom_headers/{key}`

Updates a custom header for a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Name of the custom header

Request body (`application/json`)

Property	Type	Description
`value` Required	String	Value of the custom header

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a custom header

`DELETE /api/v4/hooks/{hook_id}/custom_headers/{key}`

Deletes a custom header from a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Name of the custom header

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Update a URL variable

`PUT /api/v4/hooks/{hook_id}/url_variables/{key}`

Updates a URL variable for a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Key of the URL variable

Request body (`application/json`)

Property	Type	Description
`value` Required	String	Value of the URL variable

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a URL variable

`DELETE /api/v4/hooks/{hook_id}/url_variables/{key}`

Deletes a URL variable from a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Key of the URL variable

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all webhooks for a project

`GET /api/v4/projects/{id}/hooks`

Lists all webhooks for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProject Hook
`400`	Bad Request	—
`404`	Not Found	—

Add a webhook to a project

`POST /api/v4/projects/{id}/hooks`

Adds a webhook to a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`branch_filter_strategy`	String	Filter push events by branch. Defaults to `wildcard` Allowed values: `wildcard`, `regex`, `all_branches`
`confidential_issues_events`	Boolean	If `true`, triggers the project webhook on confidential issue events
`confidential_note_events`	Boolean	If `true`, triggers the project webhook on confidential note events
`custom_headers`	Array of objects	Custom headers for the hook
`custom_headers[].key` Required	String	Name of the header Example: `X-Custom-Header`
`custom_headers[].value` Required	String	Value of the header Example: `value`
`custom_webhook_template`	String	Custom template for the webhook request payload
`deployment_events`	Boolean	If `true`, triggers the project webhook on deployment events
`description`	String	Description of the project webhook. Introduced in GitLab 17.1
`duo_flow_callback_enabled`	Boolean	If `true`, allows Duo Agent Platform flows to send lifecycle and progress events to this project webhook. A flow must reference the webhook's ID as `callback_hook_id` when it starts. Introduced in GitLab 19.4 with a feature flag named `duo_flow_callback_hooks`. Disabled by default
`emoji_events`	Boolean	

Property	Type	Description
		If `true`, triggers the project webhook on emoji events
`enable_ssl_verification`	Boolean	If `true`, verifies the SSL certificate when the webhook is triggered
`feature_flag_events`	Boolean	If `true`, triggers the project webhook on feature flag events
`issues_events`	Boolean	If `true`, triggers the project webhook on issue events
`job_events`	Boolean	If `true`, triggers the project webhook on job events
`merge_requests_events`	Boolean	If `true`, triggers the project webhook on merge request events
`milestone_events`	Boolean	If `true`, triggers the project webhook on milestone events
`name`	String	Name of the project webhook. Introduced in GitLab 17.1
`note_events`	Boolean	If `true`, triggers the project webhook on note events
`pipeline_events`	Boolean	If `true`, triggers the project webhook on pipeline events
`push_events`	Boolean	If `true`, triggers the project webhook on push events
`push_events_branch_filter`	String	Filter push events by branch name
`releases_events`	Boolean	If `true`, triggers the project webhook on release events
`resource_access_token_events`	Boolean	If `true`, triggers the project webhook on project access token expiry events
	Boolean	

Property	Type	Description
`resource_deploy_token_events`		If `true`, triggers the project webhook on project deploy token expiry events
`signing_token`	String	HMAC signing token used to compute the `webhook-signature` header. Must be in `whsec_<base64>` format encoding a 32-byte key, and is not returned in the response. Introduced in GitLab 19.0
`tag_push_events`	Boolean	If `true`, triggers the project webhook on tag push events
`token`	String	Secret token used to validate received payloads. Not returned in the response, and changing the webhook URL resets it
`url` Required	String	Webhook URL Example: `http://example.com/hook`
`url_variables`	Array of objects	URL variables for interpolation
`url_variables[].key` Required	String	Name of the variable Example: `token`
`url_variables[].value` Required	String	Value of the variable Example: `123`
`vulnerability_events`	Boolean	If `true`, triggers the project webhook on vulnerability events
`wiki_page_events`	Boolean	If `true`, triggers the project webhook on wiki page events

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectHook
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve a project webhook

`GET /api/v4/projects/{id}/hooks/{hook_id}`

Retrieves a specified webhook for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`hook_id` Path, required	Integer	ID of the project webhook

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectHook
`400`	Bad Request	—
`404`	Not found	—

Update a project webhook

`PUT /api/v4/projects/{id}/hooks/{hook_id}`

Updates a specified webhook for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`hook_id` Path, required	Integer	ID of the project webhook

Request body (`application/json`)

Property	Type	Description
`branch_filter_strategy`	String	Filter push events by branch. Defaults to `wildcard` Allowed values: `wildcard`, `regex`, `all_branches`
`confidential_issues_events`	Boolean	If `true`, triggers the project webhook on confidential issue events
`confidential_note_events`	Boolean	If `true`, triggers the project webhook on confidential note events
`custom_headers`	Array of objects	Custom headers for the hook
`custom_headers[].key` Required	String	Name of the header Example: `X-Custom-Header`
`custom_headers[].value` Required	String	Value of the header Example: `value`
`custom_webhook_template`	String	Custom template for the webhook request payload
`deployment_events`	Boolean	If `true`, triggers the project webhook on deployment events
`description`	String	Description of the project webhook. Introduced in GitLab 17.1
`duo_flow_callback_enabled`	Boolean	If `true`, allows Duo Agent Platform flows to send lifecycle and progress events to this project webhook. A flow must reference the webhook's ID as `callback_hook_id` when it starts. Introduced in GitLab 19.4 with a feature flag named `duo_flow_callback_hooks`. Disabled by default
`emoji_events`	Boolean	

Property	Type	Description
		If `true`, triggers the project webhook on emoji events
`enable_ssl_verification`	Boolean	If `true`, verifies the SSL certificate when the webhook is triggered
`feature_flag_events`	Boolean	If `true`, triggers the project webhook on feature flag events
`issues_events`	Boolean	If `true`, triggers the project webhook on issue events
`job_events`	Boolean	If `true`, triggers the project webhook on job events
`merge_requests_events`	Boolean	If `true`, triggers the project webhook on merge request events
`milestone_events`	Boolean	If `true`, triggers the project webhook on milestone events
`name`	String	Name of the project webhook. Introduced in GitLab 17.1
`note_events`	Boolean	If `true`, triggers the project webhook on note events
`pipeline_events`	Boolean	If `true`, triggers the project webhook on pipeline events
`push_events`	Boolean	If `true`, triggers the project webhook on push events
`push_events_branch_filter`	String	Filter push events by branch name
`releases_events`	Boolean	If `true`, triggers the project webhook on release events
`resource_access_token_events`	Boolean	If `true`, triggers the project webhook on project access token expiry events
	Boolean	

Property	Type	Description
`resource_deploy_token_events`		If `true`, triggers the project webhook on project deploy token expiry events
`signing_token`	String	HMAC signing token used to compute the `webhook-signature` header. Must be in `whsec_<base64>` format encoding a 32-byte key, and is not returned in the response. Introduced in GitLab 19.0
`tag_push_events`	Boolean	If `true`, triggers the project webhook on tag push events
`token`	String	Secret token used to validate received payloads. Not returned in the response, and changing the webhook URL resets it
`url`	String	Webhook URL
`url_variables`	Array of objects	URL variables for interpolation
`url_variables[].key` Required	String	Name of the variable Example: `token`
`url_variables[].value` Required	String	Value of the variable Example: `123`
`vulnerability_events`	Boolean	If `true`, triggers the project webhook on vulnerability events
`wiki_page_events`	Boolean	If `true`, triggers the project webhook on wiki page events

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectHook
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a project webhook

`DELETE /api/v4/projects/{id}/hooks/{hook_id}`

Deletes a specified webhook for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`hook_id` Path, required	Integer	ID of the project webhook

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectHook
`400`	Bad Request	—
`404`	Not found	—

Update a custom header

`PUT /api/v4/projects/{id}/hooks/{hook_id}/custom_headers/{key}`

Updates a custom header for a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Name of the custom header
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`value` Required	String	Value of the custom header

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a custom header

`DELETE /api/v4/projects/{id}/hooks/{hook_id}/custom_headers/{key}`

Deletes a custom header from a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Name of the custom header
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all events

`GET /api/v4/projects/{id}/hooks/{hook_id}/events`

Lists all events for a specified webhook.

Parameters

Name	Type	Description
`status` Query	Array of strings	Response status code of the events, for example `200` or `500`. You can search by status category: `successful` (200-299), `client_failure` (400-499), and `server_failure` (500-599)
`per_page` Query	Integer	Number of items to list per page Default: `20` Maximum: `20` Minimum: `1` Example: `20`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`hook_id` Path, required	Integer	ID of the hook

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`403`	Forbidden	—
`404`	Not found	—

Resend a webhook event

`POST /api/v4/projects/{id}/hooks/{hook_id}/events/{hook_log_id}/resend`

Resends a webhook event. This endpoint has a rate limit of five requests per minute for each authenticated user for a given project or group. On GitLab Self-Managed and GitLab Dedicated, an administrator can change this limit in the application settings.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`hook_log_id` Path, required	Integer	ID of the hook log entry
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not found	—
`422`	Unprocessable entity	—
`429`	Too many requests	—

Trigger a test webhook

`POST /api/v4/projects/{id}/hooks/{hook_id}/test/{trigger}`

Triggers a test webhook. This endpoint has a rate limit of five requests per minute for each authenticated user for a given project or group. On GitLab Self-Managed and GitLab Dedicated, an administrator can change this limit in the application settings.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`trigger` Path, required	String	Type of webhook event to test Allowed values: `confidential_issues_events`, `confidential_note_events`, `deployment_events`, `emoji_events`, `feature_flag_events`, `issues_events`, `job_events`, `merge_requests_events`, `milestone_events`, `note_events`, `pipeline_events`, `push_events`, `releases_events`, `resource_access_token_events`, `resource_deploy_token_events`, `tag_push_events`, `wiki_page_events` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad request	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—
`429`	Too many requests	—

Update a URL variable

`PUT /api/v4/projects/{id}/hooks/{hook_id}/url_variables/{key}`

Updates a URL variable for a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Key of the URL variable
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`value` Required	String	Value of the URL variable

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a URL variable

`DELETE /api/v4/projects/{id}/hooks/{hook_id}/url_variables/{key}`

Deletes a URL variable from a specified webhook.

Parameters

Name	Type	Description
`hook_id` Path, required	Integer	ID of the hook
`key` Path, required	String	Key of the URL variable
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesHook

Property	Type	Description
`alert_status`	String	Example: `executable`
`branch_filter_strategy`	String	Example: `wildcard`
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`custom_headers`	Array of objects	Example: `[{"X-Custom-Header":"value"}]`
`custom_webhook_template`	String	Example: `{"event":"{{object_kind}}"}`
`description`	String	Example: `Hook description`
`disabled_until`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`enable_ssl_verification`	Boolean	—
`id`	Integer (int64)	Example: `1`
`merge_requests_events`	Boolean	—
`name`	String	Example: `Hook name`
`organization_id`	Integer	Example: `1`
`push_events`	Boolean	—
`push_events_branch_filter`	String	Example: `my-branch-*`
`repository_update_events`	Boolean	—
`signing_token_present`	Boolean	Whether an HMAC signing token is configured Example: `false`
`tag_push_events`	Boolean	—
`token_present`	Boolean	Whether a secret token is configured Example: `false`
`url`	String	Example: `https://webhook.site`

Property	Type	Description
`url_variables`	Array of objects	Example: `[{"token": "secr3t"}]`

APIEntitiesProjectHook

Property	Type	Description
`alert_status`	String	Example: `executable`
`branch_filter_strategy`	String	Example: `wildcard`
`confidential_issues_events`	Boolean	—
`confidential_note_events`	Boolean	—
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`custom_headers`	Array of objects	Example: `[{"X-Custom-Header":"value"}]`
`custom_webhook_template`	String	Example: `{"event":"{{object_kind}}"}`
`deployment_events`	Boolean	—
`description`	String	Example: `Hook description`
`disabled_until`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`duo_flow_callback_enabled`	Boolean	—
`emoji_events`	Boolean	—
`enable_ssl_verification`	Boolean	—
`feature_flag_events`	Boolean	—
`id`	Integer (int64)	Example: `1`
`issues_events`	Boolean	—
`job_events`	Boolean	—
`merge_requests_events`	Boolean	—
`milestone_events`	Boolean	—
`name`	String	Example: `Hook name`
`note_events`	Boolean	—
`organization_id`	Integer	Example: `1`

Property	Type	Description
`pipeline_events`	Boolean	—
`project_id`	Integer (int64)	Example: `1`
`push_events`	Boolean	—
`push_events_branch_filter`	String	Example: `my-branch-*`
`releases_events`	Boolean	—
`repository_update_events`	Boolean	—
`resource_access_token_events`	Boolean	—
`resource_deploy_token_events`	Boolean	—
`signing_token_present`	Boolean	Whether an HMAC signing token is configured Example: `false`
`tag_push_events`	Boolean	—
`token_present`	Boolean	Whether a secret token is configured Example: `false`
`url`	String	Example: `https://webhook.site`
`url_variables`	Array of objects	Example: `[{"token": "secr3t"}]`
`vulnerability_events`	Boolean	—
`wiki_page_events`	Boolean	—

Imports

Use this API to migrate groups and projects by using direct transfer, and to import repositories from external sources.

Before migrating by direct transfer, see the prerequisites.

> User contribution mapping is not supported when you import projects to a personal namespace. All contributions are assigned to the personal namespace owner and cannot be reassigned.

List all group or project migrations

`GET /api/v4/bulk_imports`

Lists all group or project migrations.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`sort` Query	String	Sort results in ascending or descending order by creation date Allowed values: `asc`, `desc` Default: `desc`
`status` Query	String	Filter migrations by import status Allowed values: `created`, `started`, `finished`, `timeout`, `failed`, `canceled`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBulkImport
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

Start a group or project migration

`POST /api/v4/bulk_imports`

Starts a group or project migration. To migrate a project, specify `entities[project_entity]`.

Request body (`application/x-www-form-urlencoded`)

Property	Type	Description
`configuration` Required	Object	The source GitLab instance configuration
`configuration.access_token` Required	String	Access token to the source Git Lab instance
`configuration.url` Required	String	Source GitLab instance URL
`entities` Required	Array of objects	List of entities to import
`entities[].destination_name`	String	Deprecated: Use :destination_slug instead. Destination slug for the entity Example: `'destination_slug' not 'destination/slug'`
`entities[].destination_namespace` Required	String	Destination namespace for the entity Example: `'destination_namespace' or 'destination/namespace'`
`entities[].destination_slug`	String	Destination slug for the entity Example: `'destination_slug' not 'destination/slug'`
`entities[].migrate_memberships`	Boolean	The option to migrate memberships or not Default: `true`
`entities[].migrate_projects`	Boolean	Indicates group migration should include nested projects Default: `true`
`entities[].source_full_path` Required	String	Relative path of the source entity to import Example: `'source/full/path' not`

Property	Type	Description
		`'https://example.com/source/full/path'`
`entities[].source_type` Required	String	Source entity type Allowed values: `group_entity`, `project_entity` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBulkImport
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—
`503`	Service unavailable	—

List all group or project migration entities

`GET /api/v4/bulk_imports/entities`

Lists all group or project migration entities.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`sort` Query	String	Sort results in ascending or descending order by creation date Allowed values: `asc`, `desc` Default: `desc`
`status` Query	String	Filter entities by import status Allowed values: `created`, `started`, `finished`, `timeout`, `failed`, `canceled`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBulkImports Entity
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

Retrieve a group or project migration

`GET /api/v4/bulk_imports/{import_id}`

Retrieves details of a group or project migration.

Parameters

Name	Type	Description
`import_id` Path, required	Integer	ID of the GitLab Migration

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBulkImport
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

Cancel a migration

`POST /api/v4/bulk_imports/{import_id}/cancel`

Cancels a direct transfer migration. This feature was introduced in GitLab 17.1.

Parameters

Name	Type	Description
`import_id` Path, required	Integer	ID of the GitLab Migration

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBulkImport
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

List all group or project migration entities

`GET /api/v4/bulk_imports/{import_id}/entities`

Lists all group or project migration entities for a specified migration.

Parameters

Name	Type	Description
`import_id` Path, required	Integer	ID of the GitLab Migration
`status` Query	String	Filter entities by import status Allowed values: `created`, `started`, `finished`, `timeout`, `failed`, `canceled`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBulkImportsEntity
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

Retrieve a group or project migration entity

`GET /api/v4/bulk_imports/{import_id}/entities/{entity_id}`

Retrieves details of a group or project migration entity.

Parameters

Name	Type	Description
`import_id` Path, required	Integer	ID of the GitLab Migration
`entity_id` Path, required	Integer	ID of the GitLab Migration entity

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBulkImportsEntity
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

List all failed import records for a migration entity

`GET /api/v4/bulk_imports/{import_id}/entities/{entity_id}/failures`

Lists all failed import records for a group or project migration entity. This feature was introduced in GitLab 16.6.

Parameters

Name	Type	Description
`import_id` Path, required	Integer	ID of the GitLab Migration
`entity_id` Path, required	Integer	ID of the GitLab Migration entity

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBulkImportsEntityFailure
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`503`	Service unavailable	—

Import GitHub gists into GitLab snippets

`POST /api/v4/import/github/gists`

Imports personal GitHub gists into GitLab snippets. You can import gists with up to 10 files. GitHub gists with more than 10 files are skipped. You should manually migrate these GitHub gists. If any gists cannot be imported, an email is sent with a list of gists that were not imported.

Request body (`application/json`)

Property	Type	Description
`personal_access_token` Required	String	GitHub personal access token

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`422`	Unprocessable Entity	—
`429`	Too Many Requests	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBulkImport

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`has_failures`	Boolean	Example: `false`
`id`	Integer (int64)	Example: `1`
`source_type`	String	Example: `gitlab`
`source_url`	String	Example: `https://source.gitlab.com/`
`status`	String	Allowed values: `created`, `started`, `finished`, `timeout`, `failed` Example: `finished`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

APIEntitiesBulkImportsEntity

Property	Type	Description
`bulk_import_id`	Integer (int64)	Example: `1`
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`destination_full_path`	String	Example: `some_group/source_project`
`destination_name`	String	Example: `destination_slug`
`destination_namespace`	String	Example: `destination_path`
`destination_slug`	String	Example: `destination_slug`
`entity_type`	String	Allowed values: `group`, `project`
`failures`	Array of APIEntitiesBulkImportsEntityFailure	—
`has_failures`	Boolean	Example: `false`
`id`	Integer (int64)	Example: `1`
`migrate_memberships`	Boolean	Example: `true`
`migrate_projects`	Boolean	Example: `true`
`namespace_id`	Integer (int64)	Example: `1`
`parent_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `1`
`source_full_path`	String	Example: `source_group`
`stats`	Object	—
`status`	String	Allowed values: `created`, `started`, `finished`, `timeout`, `failed` Example: `created`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

APIEntitiesBulkImportsEntityFailure

Property	Type	Description
`correlation_id_value`	String	Example: `dfcf583058ed4508e4c7c617bd7f0edd`
`exception_class`	String	Example: `Exception`
`exception_message`	String	Example: `error message`
`relation`	String	Example: `label`
`source_title`	String	Example: `title`
`source_url`	String	Example: `https://source.gitlab.com/group/-/epics/1`

Instance

Use this API to manage an instance's appearance and application settings, and to retrieve statistics about the instance.

Changes to application settings are subject to caching and might not take effect immediately. By default, GitLab caches application settings for 60 seconds. For more information, see application cache interval.

All of these endpoints require administrator access to the instance.

Get the current appearance

`GET /api/v4/application/appearance`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAppearance
`401`	Unauthorized	—
`403`	Forbidden	—

Modify appearance

`PUT /api/v4/application/appearance`

Request body (`multipart/form-data`)

Property	Type	Description
`description`	String	Markdown text shown on the sign in / sign up page
`email_header_and_footer_enabled`	Boolean	Add header and footer to all outgoing emails if enabled
`favicon`	String (binary)	Instance favicon in .ico/.png format
`footer_message`	String	Message within the system footer bar
`header_logo`	String (binary)	Instance image used for the main navigation bar
`header_message`	String	Message within the system header bar
`logo`	String (binary)	Instance image used on the sign in / sign up page
`member_guidelines`	String	Markdown text shown on the members page of a group or project
`message_background_color`	String	Background color for the system header / footer bar
`message_font_color`	String	Font color for the system header / footer bar
`new_project_guidelines`	String	Markdown text shown on the new project page
`profile_image_guidelines`	String	Markdown text shown on the profile page below Public Avatar
`pwa_description`	String	An explanation of what the Progressive Web App does
`pwa_icon`	String (binary)	Icon used for Progressive Web App
`pwa_name`	String	

Property	Type	Description
		Name of the Progressive Web App
`pwa_short_name`	String	Optional, short name for Progressive Web App
`site_name`	String	Last part of the webpage title. Defaults to empty
`title`	String	Instance title on the sign in / sign up page

Responses

Code	Description	Schema
`200`	OK	APIEntitiesAppearance
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—

Retrieve application settings

`GET /api/v4/application/settings`

Retrieves the current application settings for this GitLab instance.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplicationSetting
`401`	Unauthorized	—
`403`	Forbidden	—

Update application settings

`PUT /api/v4/application/settings`

Updates the current application settings for this GitLab instance. At least one setting must be updated.

Request body (`application/json`)

Property	Type	Description
`abuse_notification_email`	String	If set, abuse reports are sent to this address. Abuse reports are always available in the **Admin** area
`admin_mode`	Boolean	If `true`, requires administrators to re-authenticate before performing potentially dangerous administrative operations
`admin_notification_email`	String	If set, abuse reports are sent to this address. Abuse reports are always available in the **Admin** area. Deprecated. Use `abuse_notification_email` instead
`after_sign_out_path`	String	Users are redirected to this page after they sign out
`after_sign_up_text`	String	Text shown after sign up
`ai_action_api_rate_limit`	Integer	Maximum requests a user can make per 8 hours to aiAction endpoint
`akismet_api_key`	String	Generate API key at http://www.akismet.com
`akismet_enabled`	Boolean	If `true`, helps prevent bots from creating issues. Requires `akismet_api_key`
`allow_account_deletion`	String	If `true`, allows users to delete their accounts. Premium and Ultimate only
`allow_application_default_credentials_for_offline_transfer`	String	If `true`, allows Google Cloud Application Default Credentials for offline transfer. Even when enabled, only administrators can use these credentials, and the bucket name must start with `gitlab-offline-`

Property	Type	Description
		`transfer-`. Has no effect on GitLab.com. For more information, see Allow application default credentials for offline transfer. Introduced in GitLab 19.3
`allow_bypass_placeholder_confirmation`	String	If `true`, skips confirmation when administrators reassign placeholder users. Introduced in GitLab 18.0
`allow_contribution_mapping_to_admins`	String	—
`allow_deploy_tokens_and_keys_with_external_authn`	String	—
`allow_local_requests_from_hooks_and_services`	Boolean	If `true`, allows requests to the local network from webhooks and integrations. Deprecated. Use `allow_local_requests_from_web_hooks_and_services` instead
`allow_local_requests_from_system_hooks`	String	If `true`, allows requests to the local network from system hooks
`allow_local_requests_from_web_hooks_and_services`	String	If `true`, allows requests to the local network from webhooks and integrations
`allow_possible_spam`	String	—
`allow_project_creation_for_guest_and_below`	String	If `true`, users assigned up to the Guest role can create groups and personal projects. Defaults to `true`
`allow_runner_registration_token`	Boolean	If `true`, allows using a registration token to create a runner. Defaults to `true`

Property	Type	Description
`allow_s3_compatible_storage_for_offline_transfer`	String	If `true`, allows S3-compatible object storage for offline transfer. Introduced in GitLab 18.9
`anonymous_searches_allowed`	String	—
`archive_builds_in_human_readable`	String	Set the duration for which the jobs are considered as old and expired. After that time passes, the jobs are archived and no longer able to be retried. Make it empty to never expire jobs. It has to be no less than 1 day, for example: `15 days`, `1 month`, `2 years`
`asciidoc_max_includes`	String	Maximum number of AsciiDoc include directives processed in any one document. Default: `32`. Maximum: `64`
`asset_proxy_allowlist`	Array of strings	Assets that match these domains are not proxied. Wildcards allowed. Your GitLab installation URL is automatically allowlisted. Git Lab restart is required to apply changes
`asset_proxy_enabled`	Boolean	Enable proxying of assets
`asset_proxy_secret_key`	String	Shared secret with the asset proxy server
`asset_proxy_url`	String	URL of the asset proxy server
`asset_proxy_whitelist`	Array of strings	Assets that match these domains are not proxied. Wildcards allowed. Your GitLab installation URL is automatically allowlisted. Git Lab restart is required to apply

Property	Type	Description
		changes. Deprecated. Use `asset_proxy_allowlist` instead
`authn_data_retention_cleanup_enabled`	Boolean	If `true`, runs cleanup workers that permanently delete authentication login history older than one year, and previously revoked OAuth access tokens and grants older than one month. Introduced in GitLab 18.7
`authorized_keys_enabled`	String	If `true`, GitLab uses the `authorized_keys` file to authenticate SSH keys, which supports Git over SSH without additional configuration. Disable only if you have configured your OpenSSH server to use `AuthorizedKeysCommand`. Defaults to `true`
`auto_accept_awarded_achievements`	String	If `true`, newly awarded achievements are accepted automatically and appear on user profiles immediately. Does not affect achievements awarded before this setting is enabled. Recipients can still hide any achievement. Default value: `false`. Introduced in Git Lab 19.4
`auto_devops_domain`	String	Specify a domain to use by default for every project's Auto Review Apps and Auto Deploy stages
`auto_devops_enabled`	String	If `true`, enables Auto DevOps for projects by default. Auto DevOps automatically builds,

Property	Type	Description
		tests, and deploys applications based on a predefined CI/CD configuration
`autocomplete_users_limit`	Integer	Maximum number of authenticated requests per minute per user to the users autocomplete endpoint
`autocomplete_users_unauthenticated_limit`	Integer	Maximum number of unauthenticated requests per minute per IP address to the users autocomplete endpoint
`background_operations_max_jobs`	String	—
`block_jwt_for_reclaimed_paths`	String	—
`bulk_import_concurrent_pipeline_batch_limit`	Integer	Maximum number of simultaneous direct transfer batch exports to process
`bulk_import_enabled`	Boolean	If `true`, enables migrating Git Lab groups and projects by direct transfer
`bulk_import_max_download_file`	Integer	Maximum download file size, in MB, when importing from source GitLab instances by direct transfer
`bulk_import_max_download_file_size`	String	Maximum download file size when importing from source Git Lab instances by direct transfer
`can_create_group`	String	If `true`, users can create top-level groups. Defaults to `true`
`can_create_organization`	String	—
`ci_delete_pipelines_in_seconds_limit_human_readable`	String	

Property	Type	Description
		Maximum value that is allowed for configuring pipeline retention. Defaults to `1 year`
`ci_job_live_trace_enabled`	Boolean	If `true`, turns on incremental logging for job logs. Archived job logs are incrementally uploaded to object storage, which must be configured
`ci_job_trace_update_interval`	Integer	Job log update interval when the job log is not open, in seconds
`ci_job_trace_update_interval_when_being_watched`	Integer	Job log update interval when the job log is open, in seconds
`ci_lint_limit_per_user`	Integer	Maximum number of CI Lint requests per minute per user. Set to `0` for unlimited requests. Disabled by default
`ci_max_caches_per_job`	Integer	Maximum number of caches that can be defined in a single CI/CD job
`ci_max_includes`	Integer	Maximum number of includes per pipeline. Default is `150`
`ci_max_total_yaml_size_bytes`	String	Maximum amount of memory, in bytes, that can be allocated for the pipeline configuration, with all included YAML configuration files
`ci_partitions_in_seconds_limit`	String	Time window, in seconds, before new CI partitions are created and the system switches to the next set of partitions. Must be between 1 month and 6 months. Default is 1 month (`2592000`). Write-only. Not returned in GET responses. Deprecated. Use

Property	Type	Description
		`ci_partitions_in_seconds_limit_human_readable` instead. Scheduled for removal in API v5
`ci_partitions_in_seconds_limit_human_readable`	String	Time window before new CI partitions are created and the system switches to the next set of partitions. Must be between `1 month` and `6 months`. Defaults to `1 month`
`code_dropdown_custom_clients`	Array of objects	Custom “Open with” clients shown in the project Code dropdown list
`code_dropdown_custom_clients[].http_url_template`	String	URL template opened for the HTTPS clone URL. Must contain {url} exactly once
`code_dropdown_custom_clients[].name` Required	String	Name shown to users in the Code dropdown list
`code_dropdown_custom_clients[].ssh_url_template`	String	URL template opened for the SSH clone URL. Must contain {url} exactly once
`code_suggestions_api_rate_limit`	Integer	Maximum number of requests a user can make to the code suggestions endpoint per minute
`commit_email_hostname`	String	Custom hostname (for private commit emails)
`concurrent_bitbucket_import_jobs_limit`	Integer	Maximum number of simultaneous import jobs for the Bitbucket Cloud importer. Default is `100`
`concurrent_bitbucket_server_import_jobs_limit`	Integer	Maximum number of simultaneous import jobs for

Property	Type	Description
		the Bitbucket Server importer. Default is `100`
`concurrent_github_import_jobs_limit`	Integer	Maximum number of simultaneous import jobs for the GitHub importer. Default is `1000`
`concurrent_pull_request_import_jobs_limit`	Integer	Maximum number of simultaneous pull request import jobs for the GitHub, Bitbucket Cloud, and Bitbucket Server importers. Default is `200`
`concurrent_relation_batch_export_limit`	Integer	Maximum number of simultaneous batch export jobs to process
`concurrent_relation_export_limit`	String	—
`container_expiration_policies_enable_historic_entries`	String	If `true`, enables cleanup policies for all projects
`container_registry_cleanup_tags_service_max_list_size`	String	Maximum number of tags that can be deleted in a single execution of cleanup policies
`container_registry_delete_tags_service_timeout`	String	Maximum time, in seconds, that the cleanup process can take to delete a batch of tags for cleanup policies
`container_registry_expiration_policies_caching`	String	If `true`, enables caching during the execution of cleanup policies
`container_registry_expiration_policies_worker_capacity`	String	Number of workers for cleanup policies
`container_registry_token_expire_delay`	Integer	

Property	Type	Description
		Duration, in minutes, that container registry authorization tokens remain valid
`create_organization_api_limit`	String	—
`custom_http_clone_url_root`	String	Set a custom Git clone URL for HTTP(S)
`deactivate_dormant_users`	String	—
`deactivate_dormant_users_period`	String	—
`deactivation_email_additional_text`	String	—
`decompress_archive_file_timeout`	Integer	Timeout for decompressing archived files, in seconds. Set to `0` to disable the timeout
`default_artifacts_expire_in`	String	Set the default expiration time for each job's artifacts
`default_branch_name`	String	Set the initial branch name for all projects in an instance
`default_branch_protection`	Integer	Level of protection for the default branch. Deprecated in GitLab 17.0. Use `default_branch_protection_defaults` instead Allowed values: `0`, `3`, `1`, `2`, `4`
`default_branch_protection_defaults`	Object	Default branch protection settings. For available options, see Options for `default_branch_protection_defaults`. Introduced in GitLab 17.0
`default_branch_protection_defaults.allow_force_push`	Boolean	If `true`, allows force push for all users with push access

Property	Type	Description
`default_branch_protection_defaults.allowed_to_merge`	Array of objects	Array of access levels allowed to merge into the default branch
`default_branch_protection_defaults.allowed_to_merge[].access_level` Required	Integer	Access level allowed to merge into the default branch Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.allowed_to_push`	Array of objects	Array of access levels allowed to push to the default branch
`default_branch_protection_defaults.allowed_to_push[].access_level` Required	Integer	Access level allowed to push to the default branch Allowed values: `30`, `40`, `60`, `0`
`default_branch_protection_defaults.code_owner_approval_required`	Boolean	Require approval from code owners
`default_branch_protection_defaults.developer_can_initial_push`	Boolean	If `true`, allows developers to make the initial push
`default_ci_config_path`	String	Default CI/CD configuration file and path for new projects. If empty, new projects use `.gitlab-ci.yml`
`default_dark_syntax_highlighting_theme`	String	Default dark mode syntax highlighting theme for users who are new or not signed in. See IDs of available themes
`default_group_visibility`	String	What visibility level new groups receive. Default is `private`. Cannot be set to any levels in `restricted_visibility_levels` Allowed values: `private`, `internal`, `public`

Property	Type	Description
`default_preferred_language`	String	Default preferred language for users who are not logged in
`default_project_creation`	Integer	Default minimum role required to create projects. Can take: `0` *(No one)*, `1` *(Maintainers)*, `2` *(Developers)*, `3` *(Administrators)*, or `4` *(Owners)* Allowed values: `0`, `3`, `4`, `1`, `2`
`default_project_visibility`	String	What visibility level new projects receive. Default is `private`. Cannot be set to any levels in `restricted_visibility_levels` Allowed values: `private`, `internal`, `public`
`default_projects_limit`	Integer	Maximum number of personal projects per user. Default is `100000`
`default_search_scope`	String	—
`default_snippet_visibility`	String	What visibility level new snippets receive. Default is `private` Allowed values: `private`, `internal`, `public`
`default_syntax_highlighting_theme`	String	Default syntax highlighting theme for users who are new or not signed in. See IDs of available themes
`delay_user_account_self_deletion`	String	—
`delete_inactive_projects`	String	Enable dormant project deletion. Default is `false`

Property	Type	Description
`deletion_adjourned_period`	String	Number of days to wait before deleting a project or group that is marked for deletion. Value must be between `1` and `90`. Defaults to `30`
`deny_all_requests_except_allowed`	String	—
`dependency_management_settings`	Object	Dependency management settings. Set `security_update_scheduler_max_concurrency` (integer) to cap the number of security update scheduler jobs that run concurrently across the Sidekiq fleet. Default: `30`. Capped at `200`. Set to `0` to remove the concurrency limit. Ultimate only. Introduced in GitLab 19.1
`dependency_management_settings.security_update_scheduler_max_concurrency`	Integer	Maximum number of dependency management security update scheduler jobs to run concurrently across the Sidekiq fleet
`description_and_note_max_size`	Integer	Maximum work item, merge request, and vulnerability description and comment content size in bytes
`diagramsnet_enabled`	Boolean	If `true`, enables Diagrams.net integration. Requires `diagramsnet_url`. Defaults to `true`
`diagramsnet_url`	String	The Diagrams.net server URL
`diff_max_commits`	String	Maximum number of diff commits per merge request
`diff_max_files`	String	

Property	Type	Description
		Maximum number of files in a diff
`diff_max_lines`	String	Maximum number of lines in a diff
`diff_max_patch_bytes`	String	Maximum diff patch size, in bytes
`diff_max_versions`	String	Maximum number of diff versions per merge request
`disable_admin_oauth_scopes`	Boolean	Stop administrators from connecting to non-trusted OAuth applications
`disable_feed_token`	Boolean	Disable display of RSS/Atom and Calendar `feed_tokens`
`disable_password_authentication_for_users_with_sso_identities`	String	Disable password authentication in the web interface for users with an SSO identity. This does not affect Git operations over HTTP(S). Default is `false`
`disabled_oauth_sign_in_sources`	Array of strings	OAuth sign-in sources to disable
`dns_rebinding_protection_enabled`	String	If `true`, enforces DNS-rebinding attack protection
`domain_allowlist`	Array of strings	Only users with email addresses that match these domains can sign up. Wildcards allowed. For example, `domain.com` or `*.domain.com`. If empty, all domains are allowed
`domain_allowlist_raw`	String	—
`domain_denylist`	Array of strings	Users with e-mail addresses that match these domain(s) will NOT be able to sign-up.

Property	Type	Description
		Wildcards allowed. Enter multiple entries on separate lines. Ex: domain.com, *. domain.com
`domain_denylist_enabled`	Boolean	If `true`, enables the domain denylist for sign-ups. Requires `domain_denylist`
`domain_denylist_raw`	String	—
`downstream_pipeline_trigger_limit_per_project_user_sha`	Integer	Maximum number of downstream pipelines that can be triggered per minute for a given project, user, and commit. Default: `0` (no restriction)
`dsa_key_restriction`	Integer	Restrictions on the complexity of uploaded DSA keys. A value of -1 disables all DSA keys Allowed values: `0`, `1024`, `2048`, `3072`, `-1`
`dynamic_client_registration_enabled`	String	—
`ecdsa_key_restriction`	Integer	Restrictions on the complexity of uploaded ECDSA keys. A value of -1 disables all ECDSA keys Allowed values: `0`, `256`, `384`, `521`, `-1`
`ecdsa_sk_key_restriction`	Integer	Restrictions on the complexity of uploaded ECDSA_SK keys. A value of -1 disables all ECDSA_SK keys Allowed values: `0`, `256`, `-1`
`ed25519_key_restriction`	Integer	Restrictions on the complexity of uploaded ED25519 keys. A value of -1 disables all

Property	Type	Description
		ED25519 keys Allowed values: `0`, `256`, `-1`
`ed25519_sk_key_restriction`	Integer	Restrictions on the complexity of uploaded ED25519_SK keys. A value of -1 disables all ED25519_SK keys Allowed values: `0`, `256`, `-1`
`eks_access_key_id`	String	Access key ID for the EKS integration IAM user
`eks_account_id`	String	Amazon account ID for EKS integration
`eks_integration_enabled`	Boolean	Enable integration with Amazon EKS
`eks_secret_access_key`	String	Secret access key for the EKS integration IAM user
`email_author_in_body`	Boolean	If `true`, includes the name of the issue, merge request, or comment author in the email body. Useful when email servers do not support overriding the sender name
`email_confirmation_setting`	String	Specifies whether users must confirm their email address before signing in. `off` does not require confirmation, `soft` allows sign-in during a confirmation period, and `hard` requires confirmation before sign-in Allowed values: `off`, `soft`, `hard`
`email_otp_enabled`	Boolean	If `true`, enables email-based one-time passwords (OTP) as a multi-factor authentication method. Requires `require_`

Property	Type	Description
		`email_verification_on_account_locked` to be `true`
`email_restrictions`	String	Regular expression that is checked against the email used during registration
`email_restrictions_enabled`	String	If `true`, prevents new users from signing up with an email address that matches the `email_restrictions` regular expression
`enable_artifact_external_redirect_warning_page`	Boolean	If `true`, shows the external redirect page that warns about user-generated content in Git Lab Pages
`enable_language_server_restrictions`	Boolean	Enables enforcing language server restrictions
`enabled_git_access_protocol`	String	Protocol allowed for Git access. Set to `all` to allow both SSH and HTTP(S) Allowed values: `ssh`, `http`, `all`
`enforce_ci_inbound_job_token_scope_enabled`	String	—
`enforce_email_subaddress_restrictions`	String	—
`enforce_granular_tokens`	String	—
`enforce_terms`	String	If `true`, enforces the application terms of service for all users. Requires `terms`
`error_tracking_api_url`	String	—
`error_tracking_enabled`	String	—
`external_auth_client_cert`	String	Certificate to use to authenticate with the external

Property	Type	Description
		authorization service. Requires `external_auth_client_key`
`external_auth_client_key`	String	Private key for the certificate when authentication is required for the external authorization service. Encrypted when stored
`external_auth_client_key_pass`	String	Passphrase to use for the private key when authenticating with the external authorization service. Encrypted when stored
`external_authorization_service_default_label`	String	Default classification label to use when requesting authorization and no classification label has been specified on the project
`external_authorization_service_enabled`	String	If `true`, uses an external authorization service for accessing projects. Requires `external_authorization_service_default_label`, `external_authorization_service_timeout`, and `external_authorization_service_url`
`external_authorization_service_timeout`	String	The timeout after which an authorization request is aborted, in seconds. When a request times out, access is denied to the user. (min: 0.001, max: 10, step: 0.001)
`external_authorization_service_url`	String	URL to which authorization requests are directed
`external_pipeline_validation_service_timeout`	String	Time to wait for a response from the pipeline validation service, in seconds. Assumes `OK` if it times out

Property	Type	Description
`external_pipeline_validation_service_token`	String	Token to include as the `X-Gitlab-Token` header in requests to the URL in `external_pipeline_validation_service_url`
`external_pipeline_validation_service_url`	String	URL to use for pipeline validation requests
`failed_login_attempts_unlock_period_in_minutes`	String	Time period in minutes after which the user is unlocked when maximum number of failed sign-in attempts reached
`fe_application_settings_attributes`	Object	Provides FEApplicationSetting parameters
`fe_application_settings_attributes.gitaly_pull_mirroring_timeout`	Integer	—
`fe_application_settings_attributes.maintenance_mode`	Boolean	—
`fe_application_settings_attributes.pause_opensearch_indexing`	Boolean	—
`fe_application_settings_attributes.pipeline_execution_policy_limit`	Integer	Maximum pipeline execution policies per policy document and per pipeline (1 to 20)
`fe_application_settings_attributes.repository_branch_indexing_mode`	String	Repository branch indexing mode for OpenSearch code indexing Allowed values: `default_branch_only`, `allow_project_regex`
`fe_application_settings_attributes.use_advanced_search`	Boolean	—
`first_day_of_week`	String	

Property	Type	Description
		Start day of the week for calendar views and date pickers. Valid values are `0` (default) for Sunday, `1` for Monday, and `6` for Saturday
`floc_enabled`	Boolean	If `true`, enables FloC (Federated Learning of Cohorts)
`force_pages_access_control`	String	—
`git_push_pipeline_limit`	Integer	Maximum number of branch or tag pipelines that can be triggered by a single Git push. Set to `0` to disable the limit
`gitaly_timeout_default`	Integer	Default Gitaly timeout, in seconds. Set to `0` to disable timeouts
`gitaly_timeout_fast`	Integer	Gitaly fast operation timeout, in seconds. Set to 0 to disable timeouts
`gitaly_timeout_medium`	Integer	Medium Gitaly timeout, in seconds. Set to 0 to disable timeouts
`gitlab_dedicated_instance`	String	If `true`, the instance was provisioned for GitLab Dedicated
`gitlab_environment_toolkit_instance`	String	If `true`, the instance was provisioned with the GitLab Environment Toolkit for Service Ping reporting
`gitlab_product_usage_data_enabled`	String	If `true`, product usage data collection is enabled. When the `GITLAB_PRODUCT_USAGE_DATA_ENABLED` environment variable is set, the API returns the

Property	Type	Description
		effective value from the environment variable
`gitlab_shell_operation_limit`	String	Maximum number of Git operations per minute a user can perform. Default: `600`
`gitpod_enabled`	Boolean	If `true`, enables Ona integration. Requires `gitpod_url`
`gitpod_url`	String	Ona instance URL for the integration
`global_search_block_anonymous_searches_enabled`	String	—
`global_search_groups_enabled`	String	—
`global_search_merge_requests_enabled`	String	—
`global_search_snippet_titles_enabled`	String	—
`global_search_users_enabled`	String	—
`global_search_work_items_enabled`	String	—
`grafana_enabled`	Boolean	If `true`, enables Grafana
`grafana_url`	String	Grafana URL
`granular_tokens_enforced_after`	String	—
`gravatar_enabled`	Boolean	If `true`, enables the Gravatar service
`group_api_limit`	String	—
`group_archive_unarchive_api_limit`	String	—

Property	Type	Description
`group_create_limit`	String	—
`group_download_export_limit`	String	—
`group_export_limit`	String	—
`group_import_limit`	String	—
`group_invited_groups_api_limit`	String	—
`group_projects_api_limit`	String	—
`group_runner_token_expiration_interval`	Integer	Set the expiration time (in seconds) of authentication tokens of newly registered group runners. Minimum value is 7200 seconds. For more information, see Automatically rotate authentication tokens
`group_shared_groups_api_limit`	String	—
`groups_api_limit`	String	—
`hashed_storage_enabled`	String	Create new projects using hashed storage paths: Enable immutable, hash-based paths and repository names to store repositories on disk. This prevents repositories from having to be moved or renamed when the Project URL changes and may improve disk I/O performance. (Always enabled in GitLab versions 13.0 and later, configuration is scheduled for removal in 14.0)
`helm_max_packages_count`	String	—
	String	

Property	Type	Description
`help_page_documentation_base_url`		Alternate documentation pages URL
`help_page_hide_commercial_content`	Boolean	If `true`, hides marketing-related entries from help
`help_page_support_url`	String	Alternate support URL for help page and help dropdown
`help_page_text`	String	Custom text displayed on the help page
`hide_third_party_offers`	String	If `true`, hides offers from third parties in GitLab
`home_page_url`	String	Users who are not signed in are redirected to this page
`housekeeping_enabled`	Boolean	Enable automatic repository housekeeping (git repack, git gc)
`housekeeping_full_repack_period`	Integer	Number of Git pushes after which a full `git repack` is run. Deprecated. Use `housekeeping_optimize_repository_period` instead. All or none of `housekeeping_full_repack_period`, `housekeeping_gc_period`, `housekeeping_incremental_repack_period` must be provided
`housekeeping_gc_period`	Integer	Number of Git pushes after which `git gc` is run. Deprecated. Use `housekeeping_optimize_repository_period` instead. All or none of `housekeeping_full_repack_period`, `housekeeping_gc_period`, `housekeeping_incremental_repack_period` must be provided

Property	Type	Description
`housekeeping_incremental_repack_period`	Integer	Number of Git pushes after which an incremental `git repack` is run. Deprecated. Use `housekeeping_optimize_repository_period` instead. Exactly one of `housekeeping_incremental_repack_period`, `housekeeping_optimize_repository_period` must be provided. All or none of `housekeeping_full_repack_period`, `housekeeping_gc_period`, `housekeeping_incremental_repack_period` must be provided
`housekeeping_optimize_repository_period`	Integer	Number of Git pushes after which Gitaly is asked to optimize a repository. Exactly one of `housekeeping_incremental_repack_period`, `housekeeping_optimize_repository_period` must be provided
`html_emails_enabled`	Boolean	If `true`, sends emails in both HTML and plain text formats. If `false`, sends emails in plain text only. Defaults to `true`
`iframe_rendering_allowlist`	Array of strings	IDs of embed providers users can embed content from. Enter multiple entries separated by commas or on separate lines
`iframe_rendering_allowlist_raw`	String	Raw newline- or comma-separated list of IDs of embed providers users can embed content from
`iframe_rendering_enabled`	Boolean	If `true`, allows rendering of iframes in Markdown

Property	Type	Description
`import_jobs_concurrency_limit`	String	—
`import_sources`	Array of strings	Sources to allow project import from. OmniAuth must be configured for GitHub, Bitbucket, and GitLab.com
`inactive_projects_delete_after_months`	String	If `delete_inactive_projects` is `true`, the time (in months) to wait before deleting dormant projects. Default is `2`
`inactive_projects_min_size_mb`	String	If `delete_inactive_projects` is `true`, the minimum repository size, in MB, for projects to be checked for inactivity. Default is `0`
`inactive_projects_send_warning_email_after_months`	String	If `delete_inactive_projects` is `true`, sets the time (in months) to wait before emailing Maintainers that the project is scheduled to be deleted because it is dormant. Default is `1`
`inactive_resource_access_tokens_delete_after_days`	String	Specifies retention period for inactive project and group access tokens. Default is `30`
`include_optional_metrics_in_service_ping`	String	If `true`, enables optional metrics in Service Ping
`instance_token_prefix`	String	—
`invisible_captcha_enabled`	Boolean	If `true`, enables Invisible CAPTCHA spam detection during account creation
`invitation_flow_enforcement`	String	—
`issues_create_limit`	Integer	

Property	Type	Description
		Maximum number of issue creation requests allowed per minute per user. Set to 0 for unlimited requests per minute
`jira_connect_additional_audience_url`	String	—
`jira_connect_application_key`	String	ID of the OAuth application used to authenticate with the GitLab for Jira Cloud app
`jira_connect_proxy_url`	String	URL of the GitLab instance used as a proxy for the GitLab for Jira Cloud app
`jira_connect_public_key_storage_enabled`	Boolean	If `true`, enables public key storage for the GitLab for Jira Cloud app
`jira_forge_app_id`	String	Atlassian Forge app ID (ARI) of the GitLab for Jira Cloud app, used to verify inbound Forge Invocation Tokens
`job_erase_limit_per_user_project`	Integer	Maximum number of job erasures allowed per minute for a given project
`job_play_limit_per_user_project`	Integer	Maximum number of manual job runs allowed per minute for a given project
`job_retry_limit_per_user_project`	Integer	Maximum number of job retries allowed per minute for a given project
`keep_latest_artifact`	String	If `true`, prevents the deletion of the artifacts from the most recent successful jobs, regardless of the expiry time. Defaults to `true`
	String	

Property	Type	Description
`kroki_diagram_proxy_enabled`		Enable Kroki diagram proxy. Default is `false`
`kroki_enabled`	Boolean	If `true`, enables Kroki integration. Requires `kroki_url`
`kroki_formats`	String	Additional formats supported by the Kroki instance. Possible values are `true` or `false` for formats `bpmn`, `blockdiag`, `excalidraw`, and `mermaid` in the format `<format>: true` or `<format>: false`
`kroki_url`	String	Kroki server URL
`local_markdown_version`	Integer	Local Markdown version. Increase this value to invalidate all cached Markdown
`lock_require_sha_for_merge`	String	If `true`, enforces the `require_sha_for_merge` setting for all groups on the instance. Introduced in GitLab 19.2
`lock_resource_access_token_notify_inherited`	String	—
`logging_field_dual_emit_target`	Integer	Version to dual-emit alongside schema_version. Must be strictly greater than schema_version, or omit/null to disable Allowed values: `1`
`logging_field_schema_version`	Integer	Logging field schema version (v0, v1, …). Cannot be downgraded Allowed values: `0`, `1`
`login_recaptcha_protection_enabled`	Boolean	Helps prevent brute-force attacks

Property	Type	Description
`mailgun_events_enabled`	Boolean	Enable Mailgun event receiver
`mailgun_signing_key`	String	Mailgun HTTP webhook signing key for receiving events from webhook
`math_rendering_limits_enabled`	String	—
`max_artifacts_content_include_size`	String	—
`max_artifacts_size`	Integer	Maximum file size, in MB, for each job's artifacts
`max_attachment_size`	Integer	Maximum attachment size in MB
`max_decompressed_archive_size`	Integer	Maximum decompressed file size for imported archives in MB. Set to `0` for unlimited. Default is `25600`
`max_export_size`	Integer	Maximum export size in MB
`max_github_response_json_value_count`	Integer	Maximum object count for Git Hub API responses. Set to `0` for unlimited. The count is an estimate based on the number of `:`, `,`, `{`, and `[` characters in the response
`max_github_response_size_limit`	Integer	Maximum size in MB for Git Hub API responses. Set to `0` for unlimited size
`max_http_decompressed_size`	String	Maximum allowed size in MiB for Gzip-compressed HTTP responses from outbound requests after decompression. Set to `0` for unlimited
`max_http_response_csv_structural_chars`	String	Maximum allowed object count in CSV HTTP responses from

Property	Type	Description
		outbound requests. Count is an estimate based on the number of `,`, `;`, `\t`, and `\n` occurrences in the response. Introduced in GitLab 18.4
`max_http_response_json_depth`	String	Maximum allowed nesting depth in JSON HTTP responses from outbound requests
`max_http_response_json_structural_chars`	String	Maximum allowed object count in JSON HTTP responses from outbound requests. Count is an estimate based on the number of `:`, `,`, `{`, and `[` occurrences in the response. Introduced in GitLab 18.4
`max_http_response_size_limit`	String	Maximum allowed size in MiB for HTTP responses from outbound requests. Set to `0` for unlimited. Applicable for integrations, importers, and webhooks. Introduced in Git Lab 18.4
`max_http_response_xml_structural_chars`	String	Maximum allowed object count in XML HTTP responses from outbound requests. Count is an estimate based on the number of `<`, and `=` occurrences in the response. Introduced in Git Lab 18.4
`max_import_remote_file_size`	Integer	Maximum remote file size in MB for imports from external object storages
`max_import_size`	Integer	Maximum import size in MB
`max_login_attempts`	String	

Property	Type	Description
		Maximum number of sign-in attempts before locking out the user
`max_pages_custom_domains_per_project`	Integer	Maximum number of GitLab Pages custom domains per project
`max_pages_size`	Integer	Maximum size of each GitLab Pages site in MB. Set to `0` to allow up to 1 TB
`max_terraform_state_size_bytes`	Integer	Maximum size of the Terraform state file in bytes. Set to `0` for unlimited size
`max_yaml_depth`	String	Maximum depth of nested CI/CD configuration added with the `include` keyword. Default: `100`
`max_yaml_size_bytes`	String	Maximum size in bytes of a single CI/CD configuration file. Default: `2097152`
`mcp_server_enabled`	String	—
`members_delete_limit`	String	—
`metrics_method_call_threshold`	Integer	Method call is only tracked if it takes longer than this threshold, in milliseconds
`minimum_language_server_version`	String	Minimum language server version to accept requests from
`minimum_password_length`	String	Indicates whether passwords require a minimum length. Premium and Ultimate only
`mirror_available`	String	If `true`, project Maintainers can configure repository mirroring. If `false`, only

Property	Type	Description
		administrators can configure repository mirroring
`namespace_aggregation_schedule_lease_duration_in_seconds`	Integer	Maximum duration, in seconds, between refreshes of namespace statistics. Defaults to `300`
`notes_create_limit`	String	—
`notes_create_limit_allowlist_raw`	String	—
`notify_on_unknown_sign_in`	String	If `true`, sends a notification when a sign-in from an unknown IP address occurs
`nuget_skip_metadata_url_validation`	String	—
`oauth_access_token_expires_in`	Integer	Maximum lifetime in seconds of all new OAuth access tokens issued by the instance. Minimum value: `300` (5 minutes). Default value: `7200` (2 hours). If blank or `null`, uses default value. Does not affect existing OAuth access tokens
`observability_backend_ssl_verification_enabled`	String	—
`offline_transfer_exports_enabled`	String	If `true`, enables exporting Git Lab groups and projects by offline transfer. Introduced in GitLab 19.3
`offline_transfer_imports_enabled`	String	If `true`, enables importing Git Lab groups and projects by offline transfer. Introduced in GitLab 19.3
	String	—

Property	Type	Description
`organization_cluster_agent_authorization_enabled`		
`outbound_local_requests_allowlist_raw`	String	—
`outbound_local_requests_whitelist`	Array of strings	List of trusted domains or IP addresses to which local requests are allowed when local requests for webhooks and integrations are disabled
`package_registry_allow_anyone_to_pull_option`	String	If `true`, makes the option to allow anyone to pull from the package registry visible and changeable in project settings
`package_registry_cleanup_policies_worker_capacity`	String	Number of workers assigned to the packages cleanup policies
`pages_domain_verification_enabled`	String	If `true`, requires users to prove ownership of custom domains. Domain verification is an essential security measure for public GitLab sites. Users are required to demonstrate they control a domain before it is enabled
`pages_extra_deployments_default_expiry_seconds`	String	—
`pages_unique_domain_default_enabled`	String	If `true`, enables unique domains by default for Pages sites to avoid cookie sharing between sites under a given namespace. Defaults to `true`
`password_authentication_enabled`	Boolean	Flag indicating if password authentication is enabled for the web interface. Mutually exclusive with `password_`

Property	Type	Description
		`authentication_enabled_for_web`, `signin_enabled`
`password_authentication_enabled_for_git`	Boolean	If `true`, enables authentication for Git over HTTP(S) via a GitLab account password. Defaults to `true`
`password_authentication_enabled_for_web`	Boolean	If `true`, enables authentication for the web interface via a GitLab account password. Defaults to `true`. Mutually exclusive with `password_authentication_enabled`, `signin_enabled`
`performance_bar_allowed_group_id`	String	Path of the group that is allowed to toggle the performance bar. Deprecated. Use `performance_bar_allowed_group_path` instead
`performance_bar_allowed_group_path`	String	Path of the group that is allowed to toggle the performance bar
`performance_bar_enabled`	String	Deprecated: Pass `performance_bar_allowed_group_path: nil` instead. Allow enabling the performance
`personal_access_token_prefix`	String	Prefix to prepend to all personal access tokens
`pg_ash_sample_interval_seconds`	String	—
`pg_ash_sampling_enabled`	String	—
`pipeline_cancel_limit_per_user_project`	Integer	Maximum number of pipeline cancellations allowed per minute for a given project
	Integer	

Property	Type	Description
`pipeline_delete_limit_per_user_project`		Maximum number of pipeline deletions allowed per minute for a given project
`pipeline_limit_per_project_user_sha`	Integer	Maximum number of pipeline creation requests per minute per user and commit. Set to `0` for unlimited requests. Disabled by default
`pipeline_limit_per_user`	Integer	Maximum number of pipeline creation requests allowed per minute per user. Set to `0` for unlimited requests
`pipeline_retry_limit_per_user_project`	Integer	Maximum number of pipeline retries allowed per minute for a given project
`plantuml_diagram_proxy_enabled`	String	Enable PlantUML diagram proxy. Default is `false`
`plantuml_enabled`	Boolean	If `true`, enables PlantUML integration. Requires `plantuml_url`
`plantuml_url`	String	The PlantUML server URL
`polling_interval_multiplier`	Number	Interval multiplier used by endpoints that perform polling. Set to `0` to disable polling
`project_api_limit`	String	—
`project_create_limit`	String	—
`project_download_export_limit`	String	—
`project_export_enabled`	Boolean	If `true`, enables project export
`project_export_limit`	String	—
`project_import_limit`	String	—
	String	—

Property	Type	Description
`project_invited_groups_api_limit`		
`project_jobs_api_rate_limit`	Integer	Maximum authenticated requests to /project/:id/jobs per minute
`project_members_api_limit`	String	—
`project_repositories_blobs_batch_limit`	String	—
`project_runner_token_expiration_interval`	Integer	Set the expiration time (in seconds) of authentication tokens of newly registered project runners. Minimum value is 7200 seconds. For more information, see Automatically rotate authentication tokens
`projects_api_limit`	String	—
`projects_api_rate_limit_unauthenticated`	String	Maximum number of requests per 10 minutes per IP address for unauthenticated requests to the list all projects API. Default: `400`. To disable throttling, set to `0`
`prometheus_metrics_enabled`	Boolean	If `true`, enables Prometheus metrics
`protected_ci_variables`	String	If `true`, CI/CD variables are protected by default
`protected_paths_for_get_request_raw`	String	—
`protected_paths_raw`	String	—
`push_event_activities_limit`	Integer	Maximum number of changes (branches or tags) in a single push above which a bulk push

Property	Type	Description
		event is created. Setting to `0` does not disable throttling
`push_event_hooks_limit`	Integer	Maximum number of changes (branches or tags) in a single push above which webhooks and integrations are not triggered. Setting to `0` does not disable throttling. Default: `3`
`rate_limiting_response_text`	String	When rate limiting is enabled via the `throttle_*` settings, send this plain text response when a rate limit is exceeded. `Retry later` is sent if this is blank
`raw_blob_request_limit`	Integer	Maximum number of requests per minute for each raw path (default is `300`). Set to `0` to disable throttling
`raw_blob_request_limit_unauthenticated`	Integer	Maximum number of unauthenticated requests per minute across all raw paths in a project (default is `800`). Set to `0` to disable throttling
`recaptcha_enabled`	Boolean	If `true`, enables reCAPTCHA. Requires `recaptcha_private_key` and `recaptcha_site_key`
`recaptcha_private_key`	String	Generate private key at http://www.google.com/recaptcha
`recaptcha_site_key`	String	Generate site key at http://www.google.com/recaptcha
`receive_max_input_size`	String	Maximum push size (MB)
`reindexing_minimum_index_size`	String	—

Property	Type	Description
`reindexing_minimum_relative_bloat_size`	String	—
`relation_export_batch_size`	String	Size of each batch when exporting batched relations. Introduced in GitLab 18.2
`remember_me_enabled`	String	If `true`, enables the **Remember me** setting
`repository_checks_enabled`	Boolean	If `true`, GitLab periodically runs `git fsck` in all project and wiki repositories to look for silent disk corruption issues
`repository_storages_weighted`	Object	Hash of names of storages taken from `gitlab.yml` to weights ranging from 0 to 100. New projects are created in one of these stores, chosen by a weighted random selection
`require_admin_approval_after_user_signup`	Boolean	If `true`, requires explicit administrator approval for new signups
`require_admin_two_factor_authentication`	String	Allow administrators to require 2FA for all administrators on the instance
`require_email_verification_on_account_locked`	String	If `true`, all users on the instance must verify their identity after suspicious sign-in activity is detected
`require_personal_access_token_expiry`	Boolean	If `true`, requires an expiration date on group and project access tokens, and on personal access tokens not owned by service accounts
`require_sha_for_merge`	String	If `true`, requires a valid commit `sha` for calls to the merge a merge request

Property	Type	Description
		endpoint. Sets the instance default. Introduced in GitLab 19.2
`require_two_factor_authentication`	Boolean	If `true`, requires all users to set up two-factor authentication. Requires `two_factor_grace_period`
`resource_access_token_notify_inherited`	String	—
`resource_usage_limits`	Object	Definition for resource usage limits enforced in Sidekiq workers
`restricted_visibility_levels`	Array of strings	Selected levels cannot be used by non-administrator users for groups, projects, or snippets. Can take `private`, `internal`, and `public` as a parameter. If the `public` level is restricted, user profiles are only visible to signed-in users. If empty, no levels are restricted. Cannot select levels that are set as `default_project_visibility` and `default_group_visibility`
`root_moved_permanently_redirection`	String	—
`rsa_key_restriction`	Integer	Restrictions on the complexity of uploaded RSA keys. A value of -1 disables all RSA keys Allowed values: `0`, `1024`, `2048`, `3072`, `4096`, `-1`
`runner_jobs_endpoints_api_limit`	String	Maximum number of requests per minute per job token for requests to the `/jobs/*` runner jobs API endpoints.

Property	Type	Description
		Default: `200`. To disable throttling, set to `0`. Introduced in GitLab 18.5
`runner_jobs_patch_trace_api_limit`	String	Maximum number of requests per minute per runner token for requests to the `PATCH /jobs/:id/trace` runner jobs API endpoint. Default: 2000. To disable throttling, set to 0. Introduced in GitLab 18.5
`runner_jobs_request_api_limit`	String	Maximum number of requests per minute per runner token for requests to the `/jobs/request` runner jobs API endpoint. Default: `2000`. To disable throttling, set to `0`. Introduced in GitLab 18.5
`runner_token_expiration_interval`	Integer	Set the expiration time (in seconds) of authentication tokens of newly registered instance runners. Minimum value is 7200 seconds. For more information, see Automatically rotate authentication tokens
`search_rate_limit`	String	Maximum number of requests per minute for performing a search while authenticated. Default: `30`. To disable throttling, set to `0`
`search_rate_limit_allowlist_raw`	String	—
`search_rate_limit_unauthenticated`	String	Maximum number of requests per minute for performing a search while unauthenticated.

Property	Type	Description
		Default: `10`. To disable throttling, set to `0`
`security_policy_global_group_approvers_enabled`	Boolean	If `true`, looks up merge request approval policy approval groups globally. If `false`, looks them up within project hierarchies
`security_txt_content`	String	Public security contact information made available at https://gitlab.example.com/.well-known/security.txt
`sentry_clientside_dsn`	String	—
`sentry_clientside_traces_sample_rate`	String	—
`sentry_dsn`	String	—
`sentry_enabled`	String	—
`sentry_environment`	String	—
`session_expire_delay`	Integer	Session duration in minutes. Requires a GitLab restart to apply changes
`session_expire_from_init`	Boolean	If `true`, sessions expire a number of minutes after the session was created rather than after the last activity. The session lifetime is defined by `session_expire_delay`
`shared_runners_enabled`	Boolean	Enable shared runners for new projects
`shared_runners_text`	String	Shared runners text
`show_migrate_from_jenkins_banner`	Boolean	If `true`, shows the Jenkins migration banner
	String	

Property	Type	Description
`sidekiq_job_limiter_compression_threshold_bytes`		Threshold in bytes at which Sidekiq jobs are compressed before being stored in Redis. Default: 100,000 bytes (100 KB)
`sidekiq_job_limiter_limit_bytes`	String	Threshold in bytes at which Sidekiq jobs are rejected. Default: 0 bytes (doesn't reject any job)
`sidekiq_job_limiter_mode`	String	Sets the behavior for Sidekiq job size limits. Default: 'compress' Allowed values: `track`, `compress`
`sidekiq_timezone_override`	String	IANA timezone identifier (for example, `America/Chicago`) applied to all Sidekiq cron jobs. When blank, no override is applied and cron jobs use the Rails application timezone
`sign_in_restrictions`	String	Application sign in restrictions
`signin_enabled`	Boolean	If `true`, enables password authentication for the web interface. Deprecated. Use `password_authentication_enabled_for_web` instead. Mutually exclusive with `password_authentication_enabled_for_web`, `password_authentication_enabled`
`signup_enabled`	Boolean	If `true`, enables registration. Defaults to `true`
`silent_admin_exports_enabled`	String	If `true`, enables Silent administrator exports. Default is `false`

Property	Type	Description
`silent_mode_enabled`	String	If `true`, enables Silent mode. Default is `false`
`slack_app_enabled`	Boolean	If `true`, enables the GitLab for Slack app. Requires `slack_app_id`, `slack_app_secret`, `slack_app_signing_secret`, and `slack_app_verification_token`
`slack_app_id`	String	Client ID of the GitLab for Slack app
`slack_app_secret`	String	Client secret of the GitLab for Slack app. Used for authenticating OAuth requests from the app
`slack_app_signing_secret`	String	Signing secret of the GitLab for Slack app. Used for authenticating API requests from the app
`slack_app_verification_token`	String	Verification token of the GitLab for Slack app. This method of authentication is deprecated by Slack and used only for authenticating slash commands from the app
`snippet_size_limit`	String	Maximum snippet content size in bytes. Default: 52428800 bytes (50 MB)
`snowplow_app_id`	String	Snowplow site name or application ID, for example `gitlab`
`snowplow_collector_hostname`	String	The Snowplow collector hostname
`snowplow_cookie_domain`	String	

Property	Type	Description
		Snowplow cookie domain, for example `.gitlab.com`
`snowplow_database_collector_hostname`	String	The Snowplow collector for database events hostname. (for example, `your-db-snowplow-collector.example.com`)
`snowplow_enabled`	Boolean	Enable Snowplow tracking
`sourcegraph_enabled`	Boolean	If `true`, enables Sourcegraph integration. Requires `sourcegraph_url`
`sourcegraph_public_only`	Boolean	If `true`, blocks Sourcegraph from being loaded on private and internal projects. Defaults to `true`
`sourcegraph_url`	String	Configured Sourcegraph instance URL
`spam_check_api_key`	String	API key used by GitLab for accessing the Spam Check service endpoint
`spam_check_endpoint_enabled`	Boolean	Enable Spam Check via external API endpoint
`spam_check_endpoint_url`	String	URL of the external Spamcheck service endpoint. Valid URI schemes are `grpc` or `tls`. Specifying `tls` forces communication to be encrypted
`static_objects_external_storage_auth_token`	String	Authentication token for the external storage linked in `static_objects_external_storage_url`
`static_objects_external_storage_url`	String	URL to an external storage for repository static objects

Property	Type	Description
`tags_create_limit`	String	—
`terminal_max_session_time`	Integer	Maximum time for a web terminal WebSocket connection, in seconds. Set to `0` for unlimited time
`terms`	String	(**Required by**: `enforce_terms`) Markdown content for the ToS
`terraform_state_encryption_enabled`	Boolean	Enable encryption for Terraform state files
`throttle_authenticated_api_enabled`	String	If `true`, enables the authenticated API request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). Requires `throttle_authenticated_api_period_in_seconds` and `throttle_authenticated_api_requests_per_period`
`throttle_authenticated_api_period_in_seconds`	String	Rate limit period in seconds
`throttle_authenticated_api_requests_per_period`	String	Maximum requests per period per user
`throttle_authenticated_dependency_proxy_enabled`	String	If `true`, enforces the authenticated dependency proxy request rate limit. Default value: `false`
`throttle_authenticated_dependency_proxy_period_in_seconds`	String	Rate limit period in seconds. `throttle_authenticated_dependency_proxy_enabled` must be `true`. Default value: `15`
	String	Maximum requests per period per user. `throttle_`

Property	Type	Description
`throttle_authenticated_dependency_proxy_requests_per_period`		`authenticated_dependency_proxy_enabled` must be `true`. Default value: `1000`
`throttle_authenticated_deprecated_api_enabled`	String	—
`throttle_authenticated_deprecated_api_period_in_seconds`	String	—
`throttle_authenticated_deprecated_api_requests_per_period`	String	—
`throttle_authenticated_files_api_enabled`	String	—
`throttle_authenticated_files_api_period_in_seconds`	String	—
`throttle_authenticated_files_api_requests_per_period`	String	—
`throttle_authenticated_git_http_enabled`	String	If `true`, enforces the authenticated Git HTTP request rate limit. Default value: `false`
`throttle_authenticated_git_http_period_in_seconds`	String	Rate limit period in seconds. `throttle_authenticated_git_http_enabled` must be `true`. Default value: `3600`
`throttle_authenticated_git_http_requests_per_period`	String	Maximum requests per period per user. `throttle_authenticated_git_http_enabled` must be `true`. Default value: `3600`
`throttle_authenticated_git_lfs_enabled`	String	—
	String	—

Property	Type	Description
`throttle_authenticated_git_lfs_period_in_seconds`		
`throttle_authenticated_git_lfs_requests_per_period`	String	—
`throttle_authenticated_mcp_enabled`	String	If `true`, enforces the rate limit on requests to the MCP Server endpoint, `/api/v4/mcp`, for any HTTP method. Does not apply to other MCP endpoints. Default value: `false`
`throttle_authenticated_mcp_period_in_seconds`	String	Rate limit period in seconds. `throttle_authenticated_mcp_enabled` must be `true`. Default value: `60`
`throttle_authenticated_mcp_requests_per_period`	String	Maximum requests per period per user. `throttle_authenticated_mcp_enabled` must be `true`. Default value: `600`
`throttle_authenticated_packages_api_enabled`	String	If `true`, enables the authenticated package registry API request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). View package registry rate limits for more details. Requires `throttle_authenticated_packages_api_period_in_seconds` and `throttle_authenticated_packages_api_requests_per_period`
`throttle_authenticated_packages_api_period_in_seconds`	String	Rate limit period in seconds. View package registry rate limits for more details
	String	

Property	Type	Description
`throttle_authenticated_packages_api_requests_per_period`		Maximum requests per period per user. View package registry rate limits for more details
`throttle_authenticated_web_enabled`	String	If `true`, enables the authenticated web request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). Requires `throttle_authenticated_web_period_in_seconds` and `throttle_authenticated_web_requests_per_period`
`throttle_authenticated_web_period_in_seconds`	String	Rate limit period in seconds
`throttle_authenticated_web_requests_per_period`	String	Maximum requests per period per user
`throttle_protected_paths_enabled`	String	—
`throttle_protected_paths_period_in_seconds`	String	—
`throttle_protected_paths_requests_per_period`	String	—
`throttle_unauthenticated_api_enabled`	String	If `true`, enables the unauthenticated API request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). Requires `throttle_unauthenticated_api_period_in_seconds` and `throttle_unauthenticated_api_requests_per_period`
`throttle_unauthenticated_api_period_in_seconds`	String	Rate limit period in seconds

Property	Type	Description
`throttle_unauthenticated_api_requests_per_period`	String	Maximum requests per period per IP
`throttle_unauthenticated_deprecated_api_enabled`	String	—
`throttle_unauthenticated_deprecated_api_period_in_seconds`	String	—
`throttle_unauthenticated_deprecated_api_requests_per_period`	String	—
`throttle_unauthenticated_enabled`	String	If `true`, enables the unauthenticated web request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). Requires `throttle_unauthenticated_period_in_seconds` and `throttle_unauthenticated_requests_per_period`. Deprecated in Git Lab 14.3. Use `throttle_unauthenticated_web_enabled` or `throttle_unauthenticated_api_enabled` instead
`throttle_unauthenticated_files_api_enabled`	String	—
`throttle_unauthenticated_files_api_period_in_seconds`	String	—
`throttle_unauthenticated_files_api_requests_per_period`	String	—
`throttle_unauthenticated_git_http_enabled`	String	If `true`, enforces the unauthenticated Git HTTP

Property	Type	Description
		request rate limit. Default value: `false`
`throttle_unauthenticated_git_http_period_in_seconds`	String	Rate limit period in seconds. `throttle_unauthenticated_git_http_enabled` must be `true`. Default value: `3600`
`throttle_unauthenticated_git_http_requests_per_period`	String	Maximum requests per period per IP. `throttle_unauthenticated_git_http_enabled` must be `true`. Default value: `3600`
`throttle_unauthenticated_packages_api_enabled`	String	If `true`, enables the unauthenticated package registry API request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). View package registry rate limits for more details. Requires `throttle_unauthenticated_packages_api_period_in_seconds` and `throttle_unauthenticated_packages_api_requests_per_period`
`throttle_unauthenticated_packages_api_period_in_seconds`	String	Rate limit period in seconds. View package registry rate limits for more details
`throttle_unauthenticated_packages_api_requests_per_period`	String	Maximum requests per period per IP. View package registry rate limits for more details
`throttle_unauthenticated_period_in_seconds`	String	Rate limit period in seconds. Deprecated in GitLab 14.3. Use `throttle_unauthenticated_web_period_in_seconds` or `throttle_`

Property	Type	Description
		`unauthenticated_api_period_in_seconds` instead
`throttle_unauthenticated_requests_per_period`	String	Maximum requests per period per IP. Deprecated in GitLab 14.3. Use `throttle_unauthenticated_web_requests_per_period` or `throttle_unauthenticated_api_requests_per_period` instead
`throttle_unauthenticated_web_enabled`	String	If `true`, enables the unauthenticated web request rate limit. Helps reduce request volume (for example, from crawlers or abusive bots). Requires `throttle_unauthenticated_web_period_in_seconds` and `throttle_unauthenticated_web_requests_per_period`
`throttle_unauthenticated_web_period_in_seconds`	String	Rate limit period in seconds
`throttle_unauthenticated_web_requests_per_period`	String	Maximum requests per period per IP
`time_tracking_limit_to_hours`	String	If `true`, limits display of time tracking units to hours. Default is `false`
`top_level_group_creation_enabled`	String	If `true`, users can create top-level groups with the API. Defaults to `true`
`two_factor_grace_period`	Integer	Amount of time, in hours, that users can skip enforced two-factor authentication setup
`unique_ips_limit_enabled`	String	If `true`, limits sign-ins from multiple IP addresses. Requires `unique_ips_limit_`

Property	Type	Description
		`per_user` and `unique_ips_limit_time_window`
`unique_ips_limit_per_user`	String	Maximum number of IPs per user
`unique_ips_limit_time_window`	String	How many seconds an IP is counted towards the limit
`update_runner_versions_enabled`	String	If `true`, fetches GitLab Runner release version data from GitLab.com. For more information, see how to determine which runners need to be upgraded
`usage_ping_enabled`	Boolean	Every week GitLab will report license usage back to GitLab, Inc
`usage_ping_features_enabled`	String	—
`usage_ping_generation_enabled`	String	—
`use_clickhouse_for_analytics`	String	If `true`, enables ClickHouse as a data source for analytics reports. ClickHouse must be configured for this setting to take effect. Premium and Ultimate only
`user_contributed_projects_api_limit`	String	—
`user_deactivation_emails_enabled`	Boolean	If `true`, sends emails to users when their account is deactivated
`user_default_external`	String	If `true`, newly registered users are external by default
	String	

Property	Type	Description
`user_default_internal_regex`		Specify an email address regex pattern to identify default internal users
`user_defaults_to_private_profile`	String	If `true`, newly created users have a private profile by default. Defaults to `false`
`user_oauth_applications`	String	If `true`, allows users to register any application to use GitLab as an OAuth provider. This setting does not affect group-level OAuth applications
`user_projects_api_limit`	String	—
`user_show_add_ssh_key_message`	String	When set to `false` disable the `You won't be able to pull or push repositories via SSH until you add an SSH key to your profile` warning shown to users with no uploaded SSH key
`user_starred_projects_api_limit`	String	—
`users_api_limit_followers`	String	Maximum number of requests per minute, per user or IP address. Default: 100. Set to `0` to disable limits. Introduced in GitLab 17.10
`users_api_limit_following`	String	Maximum number of requests per minute, per user or IP address. Default: 100. Set to `0` to disable limits. Introduced in GitLab 17.10
`users_api_limit_gpg_key`	String	Maximum number of requests per minute, per user or IP address. Default: 120. Set to

Property	Type	Description
		`0` to disable limits. Introduced in GitLab 17.10
`users_api_limit_gpg_keys`	String	Maximum number of requests per minute, per user or IP address. Default: 120. Set to `0` to disable limits. Introduced in GitLab 17.10
`users_api_limit_ssh_key`	String	—
`users_api_limit_ssh_keys`	String	—
`users_api_limit_status`	String	Maximum number of requests per minute, per user or IP address. Default: 240. Set to `0` to disable limits. Introduced in GitLab 17.10
`users_get_by_id_limit`	Integer	Maximum number of calls to the `/users/:id` API per 10 minutes per user. Set to `0` for unlimited requests
`users_get_by_id_limit_allowlist_raw`	String	—
`valid_runner_registrars`	Array of strings	List of types which are allowed to register a GitLab Runner. Can be `[]`, `['group']`, `['project']`, or `['group', 'project']`
`version_check_enabled`	String	If `true`, GitLab informs you when an update is available
`vscode_extension_marketplace`	Object	Settings for VS Code Extension Marketplace
`vscode_extension_marketplace.custom_values`	Object	VS Code Extension Marketplace URL's when preset is 'custom'
	Boolean	

Property	Type	Description
`vscode_extension_marketplace.enabled`		If `true`, enables the VS Code extension marketplace for the Web IDE and Workspaces
`vscode_extension_marketplace.preset`	String	The preset configuration of URL's for the VS Code Extension Marketplace
`vscode_extension_marketplace_enabled`	String	—
`vscode_extension_marketplace_extension_host_domain`	String	—
`vscode_extension_marketplace_single_origin_fallback_enabled`	String	—
`web_hook_event_resend_limit`	String	Maximum number of webhook event resend requests per minute, per user, for a given project or group. Default: `5`. Set to `0` to disable limits. Introduced in GitLab 19.3
`web_hook_test_limit`	String	Maximum number of webhook test requests per minute, per user, for a given project or group. Default: `5`. Set to `0` to disable limits. Introduced in Git Lab 19.3
`whats_new_variant`	String	Which features What's new shows. `all_tiers` shows features for all tiers, `current_tier` shows only features for the instance's tier, and `disabled` hides What's new Allowed values: `all_tiers`, `current_tier`, `disabled`
`wiki_asciidoc_allow_uri_includes`	Boolean	If `true`, allows URI includes for AsciiDoc wiki pages

Property	Type	Description
`wiki_page_max_content_bytes`	Integer	Maximum wiki page content size in bytes. Default: 5242880 bytes (5 MB). The minimum value is 1024 bytes

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplicationSetting
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—

Retrieve application statistics

`GET /api/v4/application/statistics`

Retrieves the current application statistics for this GitLab instance.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplicationStatistics
`401`	Unauthorized	—
`403`	Forbidden	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesAppearance

Property	Type	Description
`description`	String	Example: `Open source software to collaborate on code`
`email_header_and_footer_enabled`	Boolean	Example: `false`
`favicon`	String	Example: `/uploads/-/system/appearance/favicon/1/favicon.png`
`footer_message`	String	Example: `This is a footer message`
`header_logo`	String	Example: `/uploads/-/system/appearance/header_logo/1/header.png`
`header_message`	String	Example: `This is a header message`
`logo`	String	Example: `/uploads/-/system/appearance/logo/1/logo.png`
`member_guidelines`	String	Example: `Please read the member guidelines.`
`message_background_color`	String	Example: `#e75e40`
`message_font_color`	String	Example: `#ffffff`
`new_project_guidelines`	String	Example: `Please read the FAQs for help.`
`profile_image_guidelines`	String	Example: `Custom profile image guidelines`
`pwa_description`	String	Example: `GitLab as PWA`
`pwa_icon`	String	Example: `/uploads/-/system/appearance/pwa_icon/1/icon.png`
`pwa_name`	String	Example: `GitLab`
`pwa_short_name`	String	Example: `GitLab`

Property	Type	Description
`site_name`	String	Example: `GitLab`
`title`	String	Example: `GitLab Community Edition`

APIEntitiesApplicationSetting

Property	Type	Description
`abuse_notification_email`	String	—
`admin_mode`	String	—
`after_sign_out_path`	String	—
`after_sign_up_text`	String	—
`ai_action_api_rate_limit`	String	—
`akismet_api_key`	String	—
`akismet_enabled`	String	—
`allow_account_deletion`	String	—
`allow_application_default_credentials_for_offline_transfer`	String	—
`allow_bypass_placeholder_confirmation`	String	—
`allow_contribution_mapping_to_admins`	String	—
`allow_deploy_tokens_and_keys_with_external_authn`	String	—
`allow_local_requests_from_hooks_and_services`	String	—
`allow_local_requests_from_system_hooks`	String	—
`allow_local_requests_from_web_hooks_and_services`	String	—
`allow_possible_spam`	String	—
`allow_project_creation_for_guest_and_below`	String	—
`allow_runner_registration_token`	String	—
	String	—

Property	Type	Description
`allow_s3_compatible_storage_for_offline_transfer`		
`anonymous_searches_allowed`	String	—
`archive_builds_in_human_readable`	String	—
`asciidoc_max_includes`	String	—
`asset_proxy_allowlist`	String	—
`asset_proxy_enabled`	String	—
`asset_proxy_url`	String	—
`asset_proxy_whitelist`	String	—
`authn_data_retention_cleanup_enabled`	String	—
`authorized_keys_enabled`	String	—
`auto_accept_awarded_achievements`	String	—
`auto_devops_domain`	String	—
`auto_devops_enabled`	String	—
`autocomplete_users_limit`	String	—
`autocomplete_users_unauthenticated_limit`	String	—
`background_operations_max_jobs`	String	—
`block_jwt_for_reclaimed_paths`	String	—
`bulk_import_concurrent_pipeline_batch_limit`	String	—
`bulk_import_enabled`	String	—
	String	—

Property	Type	Description
`bulk_import_max_download_file_size`		
`can_create_group`	String	—
`can_create_organization`	String	—
`ci_delete_pipelines_in_seconds_limit_human_readable`	String	—
`ci_job_live_trace_enabled`	String	—
`ci_job_trace_update_interval`	Integer	—
`ci_job_trace_update_interval_when_being_watched`	Integer	—
`ci_lint_limit_per_user`	String	—
`ci_max_caches_per_job`	String	—
`ci_max_includes`	String	—
`ci_max_total_yaml_size_bytes`	String	—
`ci_partitions_in_seconds_limit_human_readable`	String	—
`code_dropdown_custom_clients`	String	—
`code_suggestions_api_rate_limit`	String	—
`commit_email_hostname`	String	—
`concurrent_bitbucket_import_jobs_limit`	String	—
`concurrent_bitbucket_server_import_jobs_limit`	String	—
`concurrent_github_import_jobs_limit`	String	—

Property	Type	Description
`concurrent_pull_request_import_jobs_limit`	String	—
`concurrent_relation_batch_export_limit`	String	—
`concurrent_relation_export_limit`	String	—
`container_expiration_policies_enable_historic_entries`	String	—
`container_registry_cleanup_tags_service_max_list_size`	String	—
`container_registry_delete_tags_service_timeout`	String	—
`container_registry_expiration_policies_caching`	String	—
`container_registry_expiration_policies_worker_capacity`	String	—
`container_registry_import_created_before`	String	—
`container_registry_import_max_retries`	String	—
`container_registry_import_max_step_duration`	String	—
`container_registry_import_max_tags_count`	String	—
`container_registry_import_start_max_retries`	String	—
`container_registry_import_target_plan`	String	—
`container_registry_import_timeout`	String	—

Property	Type	Description
`container_registry_pre_import_tags_rate`	String	—
`container_registry_pre_import_timeout`	String	—
`container_registry_token_expire_delay`	String	—
`create_organization_api_limit`	String	—
`custom_http_clone_url_root`	String	—
`deactivate_dormant_users`	String	—
`deactivate_dormant_users_period`	String	—
`deactivation_email_additional_text`	String	—
`decompress_archive_file_timeout`	String	—
`default_artifacts_expire_in`	String	—
`default_branch_name`	String	—
`default_branch_protection`	String	—
`default_branch_protection_defaults`	String	—
`default_ci_config_path`	String	—
`default_dark_syntax_highlighting_theme`	String	—
`default_group_visibility`	String	—
`default_preferred_language`	String	—
`default_project_creation`	String	—
`default_project_visibility`	String	—

Property	Type	Description
`default_projects_limit`	String	—
`default_search_scope`	String	—
`default_snippet_visibility`	String	—
`default_syntax_highlighting_theme`	String	—
`delay_user_account_self_deletion`	String	—
`delete_inactive_projects`	String	—
`deletion_adjourned_period`	String	—
`deny_all_requests_except_allowed`	String	—
`dependency_management_settings`	String	—
`description_and_note_max_size`	String	—
`diagramsnet_enabled`	String	—
`diagramsnet_url`	String	—
`diff_max_commits`	String	—
`diff_max_files`	String	—
`diff_max_lines`	String	—
`diff_max_patch_bytes`	String	—
`diff_max_versions`	String	—
`disable_admin_oauth_scopes`	String	—
`disable_feed_token`	String	—
`disable_password_authentication_for_users_with_sso_identities`	String	—
	String	—

Property	Type	Description
`disabled_oauth_sign_in_sources`		
`dns_rebinding_protection_enabled`	String	—
`domain_allowlist`	String	—
`domain_allowlist_raw`	String	—
`domain_denylist`	String	—
`domain_denylist_enabled`	String	—
`domain_denylist_raw`	String	—
`downstream_pipeline_trigger_limit_per_project_user_sha`	String	—
`dsa_key_restriction`	String	—
`dynamic_client_registration_enabled`	String	—
`ecdsa_key_restriction`	String	—
`ecdsa_sk_key_restriction`	String	—
`ed25519_key_restriction`	String	—
`ed25519_sk_key_restriction`	String	—
`eks_access_key_id`	String	—
`eks_account_id`	String	—
`eks_integration_enabled`	String	—
`email_author_in_body`	String	—
`email_confirmation_setting`	String	—
`email_otp_enabled`	String	—
`email_restrictions`	String	—
`email_restrictions_enabled`	String	—

Property	Type	Description
`enable_artifact_external_redirect_warning_page`	String	—
`enable_language_server_restrictions`	String	—
`enabled_git_access_protocol`	String	—
`enforce_ci_inbound_job_token_scope_enabled`	String	—
`enforce_email_subaddress_restrictions`	String	—
`enforce_granular_tokens`	String	—
`enforce_terms`	String	—
`error_tracking_api_url`	String	—
`error_tracking_enabled`	String	—
`external_auth_client_cert`	String	—
`external_auth_client_key`	String	—
`external_auth_client_key_pass`	String	—
`external_authorization_service_default_label`	String	—
`external_authorization_service_enabled`	String	—
`external_authorization_service_timeout`	String	—
`external_authorization_service_url`	String	—
`external_pipeline_validation_service_timeout`	String	—
	String	—

Property	Type	Description
`external_pipeline_validation_service_token`		
`external_pipeline_validation_service_url`	String	—
`failed_login_attempts_unlock_period_in_minutes`	String	—
`fe_application_settings_attributes`	Object	—
`first_day_of_week`	String	—
`floc_enabled`	String	—
`force_pages_access_control`	String	—
`git_push_pipeline_limit`	String	—
`gitaly_timeout_default`	String	—
`gitaly_timeout_fast`	String	—
`gitaly_timeout_medium`	String	—
`gitlab_dedicated_instance`	String	—
`gitlab_environment_toolkit_instance`	String	—
`gitlab_product_usage_data_enabled`	String	—
`gitlab_product_usage_data_source`	String	—
`gitlab_shell_operation_limit`	String	—
`gitpod_enabled`	String	—
`gitpod_url`	String	—
`global_search_block_anonymous_searches_enabled`	String	—
	String	—

Property	Type	Description
`global_search_groups_enabled`		
`global_search_merge_requests_enabled`	String	—
`global_search_snippet_titles_enabled`	String	—
`global_search_users_enabled`	String	—
`global_search_work_items_enabled`	String	—
`grafana_enabled`	String	—
`grafana_url`	String	—
`granular_tokens_enforced_after`	String	—
`gravatar_enabled`	String	—
`group_api_limit`	String	—
`group_archive_unarchive_api_limit`	String	—
`group_create_limit`	String	—
`group_download_export_limit`	String	—
`group_export_limit`	String	—
`group_import_limit`	String	—
`group_invited_groups_api_limit`	String	—
`group_projects_api_limit`	String	—
`group_runner_token_expiration_interval`	String	—
	String	—

Property	Type	Description
`group_shared_groups_api_limit`		
`groups_api_limit`	String	—
`hashed_storage_enabled`	String	—
`helm_max_packages_count`	String	—
`help_page_documentation_base_url`	String	—
`help_page_hide_commercial_content`	String	—
`help_page_support_url`	String	—
`help_page_text`	String	—
`hide_third_party_offers`	String	—
`home_page_url`	String	—
`housekeeping_bitmaps_enabled`	String	—
`housekeeping_enabled`	String	—
`housekeeping_full_repack_period`	String	—
`housekeeping_gc_period`	String	—
`housekeeping_incremental_repack_period`	String	—
`housekeeping_optimize_repository_period`	String	—
`html_emails_enabled`	String	—
`id`	String	—
`iframe_rendering_allowlist`	String	—
`iframe_rendering_allowlist_raw`	String	—

Property	Type	Description
`iframe_rendering_enabled`	String	—
`import_jobs_concurrency_limit`	String	—
`import_sources`	String	—
`inactive_projects_delete_after_months`	String	—
`inactive_projects_min_size_mb`	String	—
`inactive_projects_send_warning_email_after_months`	String	—
`inactive_resource_access_tokens_delete_after_days`	String	—
`include_optional_metrics_in_service_ping`	String	—
`instance_token_prefix`	String	—
`invisible_captcha_enabled`	String	—
`invitation_flow_enforcement`	String	—
`issues_create_limit`	String	—
`jira_connect_additional_audience_url`	String	—
`jira_connect_application_key`	String	—
`jira_connect_proxy_url`	String	—
`jira_connect_public_key_storage_enabled`	String	—
`jira_forge_app_id`	String	—
`job_erase_limit_per_user_project`	String	—

Property	Type	Description
`job_play_limit_per_user_project`	String	—
`job_retry_limit_per_user_project`	String	—
`keep_latest_artifact`	String	—
`kroki_diagram_proxy_enabled`	String	—
`kroki_enabled`	String	—
`kroki_formats`	String	—
`kroki_url`	String	—
`local_markdown_version`	String	—
`lock_require_sha_for_merge`	String	—
`lock_resource_access_token_notify_inherited`	String	—
`logging_field_dual_emit_target`	Integer	—
`logging_field_schema_version`	Integer	—
`login_recaptcha_protection_enabled`	String	—
`mailgun_events_enabled`	String	—
`mailgun_signing_key`	String	—
`math_rendering_limits_enabled`	String	—
`max_artifacts_content_include_size`	String	—
`max_artifacts_size`	String	—
`max_attachment_size`	String	—

Property	Type	Description
`max_decompressed_archive_size`	String	—
`max_export_size`	String	—
`max_github_response_json_value_count`	String	—
`max_github_response_size_limit`	String	—
`max_http_decompressed_size`	String	—
`max_http_response_csv_structural_chars`	String	—
`max_http_response_json_depth`	String	—
`max_http_response_json_structural_chars`	String	—
`max_http_response_size_limit`	String	—
`max_http_response_xml_structural_chars`	String	—
`max_import_remote_file_size`	String	—
`max_import_size`	String	—
`max_login_attempts`	String	—
`max_pages_custom_domains_per_project`	String	—
`max_pages_size`	String	—
`max_terraform_state_size_bytes`	String	—
`max_yaml_depth`	String	—
`max_yaml_size_bytes`	String	—

Property	Type	Description
`mcp_server_enabled`	String	—
`members_delete_limit`	String	—
`metrics_method_call_threshold`	String	—
`minimum_language_server_version`	String	—
`minimum_password_length`	String	—
`mirror_available`	String	—
`namespace_aggregation_schedule_lease_duration_in_seconds`	String	—
`notes_create_limit`	String	—
`notes_create_limit_allowlist_raw`	String	—
`notify_on_unknown_sign_in`	String	—
`nuget_skip_metadata_url_validation`	String	—
`oauth_access_token_expires_in`	String	—
`observability_backend_ssl_verification_enabled`	String	—
`offline_transfer_exports_enabled`	String	—
`offline_transfer_imports_enabled`	String	—
`organization_cluster_agent_authorization_enabled`	String	—
`outbound_local_requests_allowlist_raw`	String	—

Property	Type	Description
`outbound_local_requests_whitelist`	String	—
`package_registry_allow_anyone_to_pull_option`	String	—
`package_registry_cleanup_policies_worker_capacity`	String	—
`pages_domain_verification_enabled`	String	—
`pages_extra_deployments_default_expiry_seconds`	String	—
`pages_unique_domain_default_enabled`	String	—
`password_authentication_enabled`	String	—
`password_authentication_enabled_for_git`	String	—
`password_authentication_enabled_for_web`	String	—
`performance_bar_allowed_group_id`	String	—
`personal_access_token_prefix`	String	—
`pg_ash_sample_interval_seconds`	String	—
`pg_ash_sampling_enabled`	String	—
`pipeline_cancel_limit_per_user_project`	String	—
`pipeline_delete_limit_per_user_project`	String	—
`pipeline_limit_per_project_user_sha`	String	—

Property	Type	Description
`pipeline_limit_per_user`	String	—
`pipeline_retry_limit_per_user_project`	String	—
`plantuml_diagram_proxy_enabled`	String	—
`plantuml_enabled`	String	—
`plantuml_url`	String	—
`polling_interval_multiplier`	String	—
`project_api_limit`	String	—
`project_create_limit`	String	—
`project_download_export_limit`	String	—
`project_export_enabled`	String	—
`project_export_limit`	String	—
`project_import_limit`	String	—
`project_invited_groups_api_limit`	String	—
`project_jobs_api_rate_limit`	String	—
`project_members_api_limit`	String	—
`project_repositories_blobs_batch_limit`	String	—
`project_runner_token_expiration_interval`	String	—
`projects_api_limit`	String	—
`projects_api_rate_limit_unauthenticated`	String	—

Property	Type	Description
`prometheus_metrics_enabled`	String	—
`protected_ci_variables`	String	—
`protected_paths_for_get_request_raw`	String	—
`protected_paths_raw`	String	—
`push_event_activities_limit`	String	—
`push_event_hooks_limit`	String	—
`rate_limiting_response_text`	String	—
`raw_blob_request_limit`	String	—
`raw_blob_request_limit_unauthenticated`	String	—
`recaptcha_enabled`	String	—
`recaptcha_private_key`	String	—
`recaptcha_site_key`	String	—
`receive_max_input_size`	String	—
`reindexing_minimum_index_size`	String	—
`reindexing_minimum_relative_bloat_size`	String	—
`relation_export_batch_size`	String	—
`remember_me_enabled`	String	—
`repository_checks_enabled`	String	—
`repository_storages_weighted`	String	—
`require_admin_approval_after_user_signup`	String	—

Property	Type	Description
`require_admin_two_factor_authentication`	String	—
`require_email_verification_on_account_locked`	String	—
`require_personal_access_token_expiry`	String	—
`require_sha_for_merge`	String	—
`require_two_factor_authentication`	String	—
`resource_access_token_notify_inherited`	String	—
`resource_usage_limits`	String	—
`restricted_visibility_levels`	String	—
`root_moved_permanently_redirection`	String	—
`rsa_key_restriction`	String	—
`runner_jobs_endpoints_api_limit`	String	—
`runner_jobs_patch_trace_api_limit`	String	—
`runner_jobs_request_api_limit`	String	—
`runner_token_expiration_interval`	String	—
`search_rate_limit`	String	—
`search_rate_limit_allowlist_raw`	String	—
`search_rate_limit_unauthenticated`	String	—

Property	Type	Description
`security_txt_content`	String	—
`sentry_clientside_dsn`	String	—
`sentry_clientside_traces_sample_rate`	String	—
`sentry_dsn`	String	—
`sentry_enabled`	String	—
`sentry_environment`	String	—
`session_expire_delay`	String	—
`session_expire_from_init`	String	—
`shared_runners_enabled`	String	—
`shared_runners_text`	String	—
`show_migrate_from_jenkins_banner`	String	—
`sidekiq_job_limiter_compression_threshold_bytes`	String	—
`sidekiq_job_limiter_limit_bytes`	String	—
`sidekiq_job_limiter_mode`	String	—
`sidekiq_timezone_override`	String	—
`sign_in_restrictions`	String	—
`signin_enabled`	String	—
`signup_enabled`	String	—
`silent_admin_exports_enabled`	String	—
`silent_mode_enabled`	String	—
`slack_app_enabled`	String	—
`slack_app_id`	String	—

Property	Type	Description
`slack_app_secret`	String	—
`slack_app_signing_secret`	String	—
`slack_app_verification_token`	String	—
`snippet_size_limit`	String	—
`snowplow_app_id`	String	—
`snowplow_collector_hostname`	String	—
`snowplow_cookie_domain`	String	—
`snowplow_database_collector_hostname`	String	—
`snowplow_enabled`	String	—
`sourcegraph_enabled`	String	—
`sourcegraph_public_only`	String	—
`sourcegraph_url`	String	—
`spam_check_api_key`	String	—
`spam_check_endpoint_enabled`	String	—
`spam_check_endpoint_url`	String	—
`static_objects_external_storage_auth_token`	String	—
`static_objects_external_storage_url`	String	—
`tags_create_limit`	String	—
`terminal_max_session_time`	String	—
`terms`	String	—
`terraform_state_encryption_enabled`	String	—

Property	Type	Description
`throttle_authenticated_api_enabled`	String	—
`throttle_authenticated_api_period_in_seconds`	String	—
`throttle_authenticated_api_requests_per_period`	String	—
`throttle_authenticated_dependency_proxy_enabled`	String	—
`throttle_authenticated_dependency_proxy_period_in_seconds`	String	—
`throttle_authenticated_dependency_proxy_requests_per_period`	String	—
`throttle_authenticated_deprecated_api_enabled`	String	—
`throttle_authenticated_deprecated_api_period_in_seconds`	String	—
`throttle_authenticated_deprecated_api_requests_per_period`	String	—
`throttle_authenticated_files_api_enabled`	String	—
`throttle_authenticated_files_api_period_in_seconds`	String	—
`throttle_authenticated_files_api_requests_per_period`	String	—
`throttle_authenticated_git_http_enabled`	String	—
`throttle_authenticated_git_http_period_in_seconds`	String	—

Property	Type	Description
`throttle_authenticated_git_http_requests_per_period`	String	—
`throttle_authenticated_git_lfs_enabled`	String	—
`throttle_authenticated_git_lfs_period_in_seconds`	String	—
`throttle_authenticated_git_lfs_requests_per_period`	String	—
`throttle_authenticated_mcp_enabled`	String	—
`throttle_authenticated_mcp_period_in_seconds`	String	—
`throttle_authenticated_mcp_requests_per_period`	String	—
`throttle_authenticated_packages_api_enabled`	String	—
`throttle_authenticated_packages_api_period_in_seconds`	String	—
`throttle_authenticated_packages_api_requests_per_period`	String	—
`throttle_authenticated_web_enabled`	String	—
`throttle_authenticated_web_period_in_seconds`	String	—
`throttle_authenticated_web_requests_per_period`	String	—
`throttle_protected_paths_enabled`	String	—
`throttle_protected_paths_period_in_seconds`	String	—

Property	Type	Description
`throttle_protected_paths_requests_per_period`	String	—
`throttle_unauthenticated_api_enabled`	String	—
`throttle_unauthenticated_api_period_in_seconds`	String	—
`throttle_unauthenticated_api_requests_per_period`	String	—
`throttle_unauthenticated_deprecated_api_enabled`	String	—
`throttle_unauthenticated_deprecated_api_period_in_seconds`	String	—
`throttle_unauthenticated_deprecated_api_requests_per_period`	String	—
`throttle_unauthenticated_enabled`	String	—
`throttle_unauthenticated_files_api_enabled`	String	—
`throttle_unauthenticated_files_api_period_in_seconds`	String	—
`throttle_unauthenticated_files_api_requests_per_period`	String	—
`throttle_unauthenticated_git_http_enabled`	String	—
`throttle_unauthenticated_git_http_period_in_seconds`	String	—
`throttle_unauthenticated_git_http_requests_per_period`	String	—

Property	Type	Description
`throttle_unauthenticated_packages_api_enabled`	String	—
`throttle_unauthenticated_packages_api_period_in_seconds`	String	—
`throttle_unauthenticated_packages_api_requests_per_period`	String	—
`throttle_unauthenticated_period_in_seconds`	String	—
`throttle_unauthenticated_requests_per_period`	String	—
`throttle_unauthenticated_web_enabled`	String	—
`throttle_unauthenticated_web_period_in_seconds`	String	—
`throttle_unauthenticated_web_requests_per_period`	String	—
`time_tracking_limit_to_hours`	String	—
`top_level_group_creation_enabled`	String	—
`two_factor_grace_period`	String	—
`unique_ips_limit_enabled`	String	—
`unique_ips_limit_per_user`	String	—
`unique_ips_limit_time_window`	String	—
`update_runner_versions_enabled`	String	—
`usage_ping_enabled`	String	—

Property	Type	Description
`usage_ping_features_enabled`	String	—
`usage_ping_generation_enabled`	String	—
`use_clickhouse_for_analytics`	String	—
`user_contributed_projects_api_limit`	String	—
`user_deactivation_emails_enabled`	String	—
`user_default_external`	String	—
`user_default_internal_regex`	String	—
`user_defaults_to_private_profile`	String	—
`user_oauth_applications`	String	—
`user_projects_api_limit`	String	—
`user_show_add_ssh_key_message`	String	—
`user_starred_projects_api_limit`	String	—
`users_api_limit_followers`	String	—
`users_api_limit_following`	String	—
`users_api_limit_gpg_key`	String	—
`users_api_limit_gpg_keys`	String	—
`users_api_limit_ssh_key`	String	—
`users_api_limit_ssh_keys`	String	—
`users_api_limit_status`	String	—

Property	Type	Description
`users_get_by_id_limit`	String	—
`users_get_by_id_limit_allowlist_raw`	String	—
`valid_runner_registrars`	String	—
`version_check_enabled`	String	—
`vscode_extension_marketplace`	String	—
`vscode_extension_marketplace_enabled`	String	—
`vscode_extension_marketplace_extension_host_domain`	String	—
`vscode_extension_marketplace_single_origin_fallback_enabled`	String	—
`web_hook_event_resend_limit`	String	—
`web_hook_test_limit`	String	—
`whats_new_variant`	String	—
`wiki_asciidoc_allow_uri_includes`	String	—
`wiki_page_max_content_bytes`	String	—

APIEntitiesApplicationStatistics

Property	Type	Description
`active_users`	Integer	Number of active users Example: `21`
`forks`	Integer	Approximate number of repo forks Example: `6`
`groups`	Integer	Approximate number of projects Example: `1`
`issues`	Integer	Approximate number of issues Example: `121`
`merge_requests`	Integer	Approximate number of merge requests Example: `49`
`milestones`	Integer	Approximate number of milestones Example: `3`
`notes`	Integer	Approximate number of notes Example: `6`
`projects`	Integer	Approximate number of projects Example: `4`
`snippets`	Integer	Approximate number of snippets Example: `4`
`ssh_keys`	Integer	Approximate number of SSH keys Example: `11`
`users`	Integer	Approximate number of users Example: `22`

Integrations

Use this API to manage integrations for a project, or for a group and its subgroups.

Managing integrations requires the Maintainer or Owner role for the project or group.

List all active integrations

`GET /api/v4/groups/{id}/integrations`

Lists all active integrations.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesIntegrationBasic
`401`	Unauthorized	—
`404`	Not found	—

Create or update the Apple App Store integration

`PUT /api/v4/groups/{id}/integrations/apple-app-store`

Creates or updates the Apple App Store integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`app_store_issuer_id` Required	String	Apple App Store Connect issuer ID
`app_store_key_id` Required	String	Apple App Store Connect key ID
`app_store_private_key` Required	String	Apple App Store Connect private key
`app_store_private_key_file_name` Required	String	Apple App Store Connect private key file name
`app_store_protected_refs`	Boolean	Set variables on protected branches and tags only
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Asana integration

`PUT /api/v4/groups/{id}/integrations/asana`

Creates or updates the Asana integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	User API token. The user must have access to the task. All comments are attributed to this user
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to be automatically inspected. Leave blank to include all branches
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Assembla integration

`PUT /api/v4/groups/{id}/integrations/assembla`

Creates or updates the Assembla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`subdomain`	String	The subdomain setting
`token` Required	String	The authentication token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bamboo integration

`PUT /api/v4/groups/{id}/integrations/bamboo`

Creates or updates the Bamboo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`bamboo_url` Required	String	Bamboo root URL (for example, `https://bamboo.example.com`)
`build_key` Required	String	Bamboo build plan key (for example, `KEY`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`password` Required	String	Password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	User with API access to the Bamboo server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bugzilla integration

`PUT /api/v4/groups/{id}/integrations/bugzilla`

Creates or updates the Bugzilla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Buildkite integration

`PUT /api/v4/groups/{id}/integrations/buildkite`

Creates or updates the Buildkite integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	DEPRECATED: This parameter has no effect because SSL verification is always enabled
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`project_url` Required	String	Pipeline URL (for example, `https://buildkite.com/example/pipeline`)
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Token you get after you create a Buildkite pipeline with a Deckhouse Code repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Campfire integration

`PUT /api/v4/groups/{id}/integrations/campfire`

Creates or updates the Campfire integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`room`	String	ID portion of the Campfire room URL
`subdomain`	String	`.campfirenow.com` subdomain when you're signed in
`token` Required	String	API authentication token from Campfire. To get the token, sign in to Campfire and select **My info**
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Clickup integration

`PUT /api/v4/groups/{id}/integrations/clickup`

Creates or updates the Clickup integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Code Scoring integration

`PUT /api/v4/groups/{id}/integrations/code-scoring`

Creates or updates the Code Scoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`codescoring_agent_image`	String	Optional. The image the Code Scoring scan jobs run in, with registry and tag or digest. It needs curl, ca-certificates, jq, ruby and git; the johnny agent is downloaded into it at run time. Left empty, the jobs use the default image. For an image in a private registry, set the registry login and password variables in the scan execution policy
`codescoring_bom_format`	String	Optional. Which CycloneDX version the bill of materials is written in. They all list the same components: pick an older version for a tool downstream that needs one, or the extended 1.6 for CodeScoring's own GOST properties. A Scan Execution Policy can name a format of its own, which wins. Left empty here and in the policy, the scan jobs write CycloneDX 1.6 JSON
`codescoring_ca_cert`	String	PEM certificate of the authority that signed the server's certificate. Used only when the trust setting above says so
`codescoring_cloud_resolve`	Boolean	Let the CodeScoring server work out the dependency tree instead of the runner. The only resolver that needs no package manager in the scan job image, and it finds the transitive packages a manifest read misses. Known cost: the server currently returns the

Property	Type	Description
		vulnerability report empty when this is on, so findings reach the tests summary but not the vulnerability report. Leave it off unless the dependency list matters more
`codescoring_extra_args`	String	Optional. Passed to the agent as written, for options this form does not name, such as per-manager arguments. A setting above wins over the same option written here
`codescoring_ignore_paths`	String	Optional. One glob per line. Keeps test fixtures and vendored trees out of the results
`codescoring_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`codescoring_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`codescoring_scan_mode`	String	What to scan: a source directory, a container image, or

Property	Type	Description
		an existing CycloneDX document. Defaults to the source directory
`codescoring_scan_target`	String	Optional. Directory, image reference or SBOM path, depending on the scan mode. Defaults to the whole checkout
`codescoring_stage`	String	The policy stage this scan reports as. The agent takes one, so the scan reports as one. Defaults to “build”
`codescoring_tls_mode`	String	How the scan job decides whether to trust the Code Scoring server. It applies to the requests the job makes with curl – fetching the agent, and the connection test above. The johnny agent itself does not verify the certificate at all and offers no way to make it, so nothing here changes what the scan itself trusts
`codescoring_token` Required	String	API token to authenticate and download the Johnny agent
`codescoring_url` Required	String	Base URL of the CodeScoring instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Confluence integration

`PUT /api/v4/groups/{id}/integrations/confluence`

Creates or updates the Confluence integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`confluence_url` Required	String	URL of the Confluence Workspace hosted on `atlassian.net`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Custom Issue Tracker integration

`PUT /api/v4/groups/{id}/integrations/custom-issue-tracker`

Creates or updates the Custom Issue Tracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Datadog integration

`PUT /api/v4/groups/{id}/integrations/datadog`

Creates or updates the Datadog integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	API key used for authentication with Datadog
`api_url`	String	Full URL of your Datadog site. Only required if you do not use a standard Datadog site
`archive_trace_events`	Boolean	When enabled, job logs are collected by Datadog and displayed along with pipeline execution traces
`build_events`	Boolean	Trigger event when a build is created
`datadog_ci_visibility`	Boolean	Enable CI Visibility
`datadog_env`	String	For self-managed deployments, `env` tag for all the data sent to Datadog
`datadog_service`	String	Tag all pipeline data from this Deckhouse Code instance in Datadog. Can be used when managing several self-managed deployments
`datadog_site`	String	Datadog site to send data to. Learn more about Datadog sites in the documentation
`datadog_tags`	String	Custom tags in Datadog. Specify one tag per line in the format `key:value\nkey2:value2`
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`project_events`	Boolean	—
`push_events`	Boolean	Trigger event for pushes to the repository
`subgroup_events`	Boolean	—
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Defect Dojo integration

`PUT /api/v4/groups/{id}/integrations/defect-dojo`

Creates or updates the Defect Dojo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	Token used to authenticate with the DefectDojo API
`auto_create_context`	Boolean	—
`ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`close_old_findings`	Boolean	—
`engagement_name`	String	DefectDojo engagement. Leave blank to use the branch name
`findings_active`	Boolean	—
`findings_verified`	Boolean	—
`minimum_severity`	String	Ignore findings below this severity on import
`product_name`	String	DefectDojo product. Leave blank to use the project full path
`product_type_name`	String	DefectDojo product type, used when a product is auto-created
`tls_mode`	String	How this instance decides whether to trust the DefectDojo server. Unlike a scan, this connection is made from here rather than from a job, so verification can be turned off – which leaves the upload open to interception, and the reports carry everything the scanners found
`url` Required	String	Base URL of your DefectDojo instance
`use_inherited_settings`	Boolean	

Property	Type	Description
		Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Diffblue Cover integration

`PUT /api/v4/groups/{id}/integrations/diffblue-cover`

Creates or updates the Diffblue Cover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`diffblue_access_token_name` Required	String	Access token name used by Diffblue Cover in pipelines
`diffblue_access_token_secret` Required	String	Access token secret used by Diffblue Cover in pipelines
`diffblue_license_key` Required	String	Diffblue Cover license key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Discord integration

`PUT /api/v4/groups/{id}/integrations/discord`

Creates or updates the Discord integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`issue_channel`	String	The name of the channel to receive issues_events notifications

Property	Type	Description
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`webhook` Required	String	Discord webhook (for example, `https://discord.com/api/webhooks/…`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Drone Ci integration

`PUT /api/v4/groups/{id}/integrations/drone-ci`

Creates or updates the Drone Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`drone_url` Required	String	Drone CI URL (for example, `http://drone.example.com`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Drone CI token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Emails On Push integration

`PUT /api/v4/groups/{id}/integrations/emails-on-push`

Creates or updates the Emails On Push integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`disable_diffs`	Boolean	Disable code diffs
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Emails separated by whitespace
`send_from_committer_email`	Boolean	Send from committer
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ewm integration

`PUT /api/v4/groups/{id}/integrations/ewm`

Creates or updates the Ewm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the External Wiki integration

`PUT /api/v4/groups/{id}/integrations/external-wiki`

Creates or updates the External Wiki integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`external_wiki_url` Required	String	URL of the external wiki
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Gitlab Slack Application integration

`PUT /api/v4/groups/{id}/integrations/gitlab-slack-application`

Creates or updates the Gitlab Slack Application integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Google Play integration

`PUT /api/v4/groups/{id}/integrations/google-play`

Creates or updates the Google Play integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`google_play_protected_refs`	Boolean	Set variables on protected branches and tags only
`package_name` Required	String	Package name of the app in Google Play
`service_account_key` Required	String	Google Play service account key
`service_account_key_file_name` Required	String	File name of the Google Play service account key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Hangouts Chat integration

`PUT /api/v4/groups/{id}/integrations/hangouts-chat`

Creates or updates the Hangouts Chat integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Hangouts Chat webhook (for example, `https://chat.googleapis.com/v1/spaces...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Harbor integration

`PUT /api/v4/groups/{id}/integrations/harbor`

Creates or updates the Harbor integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`password` Required	String	The password of the user
`project_name` Required	String	The name of the project in the Harbor instance. For example, `testproject`
`url` Required	String	The base URL to the Harbor instance linked to the Deckhouse Code project. For example, `https://demo.goharbor.io`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	The username created in the Harbor interface

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Irker integration

`PUT /api/v4/groups/{id}/integrations/irker`

Creates or updates the Irker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`colorize_messages`	Boolean	Colorize messages
`default_irc_uri`	String	URI to add before each recipient. The default value is `irc://irc.network.net:6697/`
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Comma-separated list of channels or email addresses
`server_host`	String	irker daemon hostname. The default value is `localhost`
`server_port`	Integer	irker daemon port. The default value is `6659`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jaga integration

`PUT /api/v4/groups/{id}/integrations/jaga`

Creates or updates the Jaga integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	If different from the Web URL followed by /backend
`comment_detail`	String	—
`commit_events`	Boolean	—
`issues_events`	Boolean	—
`jaga_issue_prefix`	String	Text that must come right before a task code
`jaga_issue_regex`	String	Regular expression in RE2 syntax that matches Jaga task codes. If empty, codes like ASDF-1 are matched
`jaga_issue_transition_automatic`	Boolean	—
`jaga_issue_transition_statuses`	String	Status names in workflow order, separated by commas. The task is moved along them, as far as it can go at each step, until it is done. Used when the automatic transition is off. Leave both empty to keep task statuses unchanged
`jaga_password` Required	String	Password of that Jaga user
`jaga_smart_commit_comment`	Boolean	—
`jaga_smart_commit_status`	Boolean	—
`jaga_smart_commit_time`	Boolean	—
`jaga_smart_commits_enabled`	Boolean	—
`jaga_time_activity_type`	String	Jaga activity type code the time is logged with, for example develop
	String	

Property	Type	Description
`jaga_username` Required		Email of the Jaga user the integration acts as
`merge_requests_events`	Boolean	—
`url` Required	String	Base URL of Jaga. With http://, the password and the tokens are sent unencrypted
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jenkins integration

`PUT /api/v4/groups/{id}/integrations/jenkins`

Creates or updates the Jenkins integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`jenkins_url` Required	String	URL of the Jenkins server
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password`	String	Password of the Jenkins server
`project_name` Required	String	Name of the Jenkins project
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Username of the Jenkins server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira integration

`PUT /api/v4/groups/{id}/integrations/jira`

Creates or updates the Jira integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	The base URL to the Jira instance API. Web URL value is used if not set (for example, `https://jira-api.example.com`)
`comment_on_event_enabled`	Boolean	Enable comments inside Jira issues on each GitLab event (commit / merge request)
`commit_events`	Boolean	Trigger event when a commit is created or updated
`issues_enabled`	String	Enable viewing Jira issues in Deckhouse Code
`jira_auth_type`	Integer	The authentication method to use with Jira. Use `0` for basic authentication, `1` for Jira personal access token, and `2` for Jira Cloud service accounts. Defaults to `0`
`jira_issue_prefix`	String	Prefix to match Jira issue keys
`jira_issue_regex`	String	Regular expression to match Jira issue keys
`jira_issue_transition_id`	String	The ID of one or more transitions for custom issue transitions.Ignored when `jira_issue_transition_automatic` is enabled. Defaults to a blank string,which disables custom transitions
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The Jira API token, password, or personal access token to use with Jira. When using Basic Authentication (`jira_`

Property	Type	Description
		`auth_type` is `0`), use an API token for Jira Cloud, and a password for Jira Data Center or Jira Server. For a Jira personal access token (`jira_auth_type` is `1`), use the personal access token
`project_keys`	Array of strings	Keys of Jira projects to display. When `issues_enabled` is `true`, this setting filters which Jira projects are shown in Deckhouse Code. It does not restrict the API token's access
`url` Required	String	The URL to the Jira project which is being linked to this Deckhouse Code project (for example, `https://jira.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	The email or username to use with Jira. Use an email for Jira Cloud, and a username for Jira Data Center and Jira Server. Required when using basic authentication (`jira_auth_type` is `0`)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira Cloud App integration

`PUT /api/v4/groups/{id}/integrations/jira-cloud-app`

Creates or updates the Jira Cloud App integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`jira_cloud_app_deployment_gating_environments`	String	Enter the environment (production,staging,testing, development) where you want to enable deployment gating. Use comma (,) to separate multiple environments
`jira_cloud_app_enable_deployment_gating`	Boolean	Enable to approve or reject blocked Deckhouse Code deployments from Jira Service Management
`jira_cloud_app_service_ids`	String	Copy and paste your JSM Service ID here. Use comma (,) to separate multiple IDs
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Linear integration

`PUT /api/v4/groups/{id}/integrations/linear`

Creates or updates the Linear integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`workspace_url` Required	String	Linear workspace URL (for example, https://linear.app/example)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Matrix integration

`PUT /api/v4/groups/{id}/integrations/matrix`

Creates or updates the Matrix integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	Custom hostname of the Matrix server. The default value is `https://matrix-client.matrix.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target room (in the format `!qPKKM111FFKKsfoCVy:matrix.org`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	The Matrix access token (for example, `syt-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost integration

`PUT /api/v4/groups/{id}/integrations/mattermost`

Creates or updates the Mattermost integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`incident_events`	Boolean	

Property	Type	Description
		Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Mattermost notifications username
`webhook` Required	String	Mattermost notifications webhook (for example, `http://mattermost.example.com/hooks/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost Slash Commands integration

`PUT /api/v4/groups/{id}/integrations/mattermost-slash-commands`

Creates or updates the Mattermost Slash Commands integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`token` Required	String	The Mattermost token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Microsoft Teams integration

`PUT /api/v4/groups/{id}/integrations/microsoft-teams`

Creates or updates the Microsoft Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Microsoft Teams webhook (for example, `https://outlook.office.com/webhook/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Ci integration

`PUT /api/v4/groups/{id}/integrations/mock-ci`

Creates or updates the Mock Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`mock_service_url` Required	String	URL of the Mock CI integration
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Monitoring integration

`PUT /api/v4/groups/{id}/integrations/mock-monitoring`

Creates or updates the Mock Monitoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Packagist integration

`PUT /api/v4/groups/{id}/integrations/packagist`

Creates or updates the Packagist integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`server`	String	URL of the Packagist server. The default value is `https://packagist.org`
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	API token of the Packagist server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	Username of a Packagist account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Phorge integration

`PUT /api/v4/groups/{id}/integrations/phorge`

Creates or updates the Phorge integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pipelines Email integration

`PUT /api/v4/groups/{id}/integrations/pipelines-email`

Creates or updates the Pipelines Email integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`notify_child_pipelines`	Boolean	Send notifications for child pipelines
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_default_branch`	Boolean	Send notifications for the default branch
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`recipients` Required	String	Comma-separated list of recipient email addresses
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pivotaltracker integration

`PUT /api/v4/groups/{id}/integrations/pivotaltracker`

Creates or updates the Pivotaltracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to automatically inspect. Leave blank to include all branches
`token` Required	String	The Pivotal Tracker token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ptai integration

`PUT /api/v4/groups/{id}/integrations/ptai`

Creates or updates the Ptai integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`ptai_ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`ptai_image` Required	String	The image the PT AI scan job runs in: the whole reference, with a tag or a digest. A new integration is filled in with the image this product ships. It needs aictl, ruby and mktemp, and curl, python3 or wget when the policy reads its settings file from another project. For an image in a private registry, set the registry login and password variables below
`ptai_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`ptai_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`ptai_server_url` Required	String	Base URL of the PT Application Inspector server. Changing it

Property	Type	Description
		clears the saved API token: enter the token again
`ptai_tls_mode`	String	Use the certificates of the job image when they already cover the server. Otherwise paste the certificate authority below
`ptai_token` Required	String	API token used to authenticate with the PT Application Inspector server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pumble integration

`PUT /api/v4/groups/{id}/integrations/pumble`

Creates or updates the Pumble integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Pumble webhook (for example, `https://api.pumble.com/workspaces/x/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pushover integration

`PUT /api/v4/groups/{id}/integrations/pushover`

Creates or updates the Pushover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	The application key
`device`	String	Leave blank for all active devices
`priority` Required	String	The priority
`push_events`	Boolean	Trigger event for pushes to the repository
`sound`	String	The sound of the notification
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`user_key` Required	String	The user key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Redmine integration

`PUT /api/v4/groups/{id}/integrations/redmine`

Creates or updates the Redmine integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Slack integration

`PUT /api/v4/groups/{id}/integrations/slack`

Creates or updates the Slack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Slack notifications username
`webhook` Required	String	Slack notifications webhook (for example, `https://hooks.slack.com/services/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Squash Tm integration

`PUT /api/v4/groups/{id}/integrations/squash-tm`

Creates or updates the Squash Tm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`token`	String	Secret token
`url` Required	String	URL of the Squash TM webhook
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Teamcity integration

`PUT /api/v4/groups/{id}/integrations/teamcity`

Creates or updates the Teamcity integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`build_type` Required	String	The build configuration ID of the TeamCity project
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`teamcity_url` Required	String	TeamCity root URL (for example, `https://teamcity.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	A user with permissions to trigger a manual build

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Telegram integration

`PUT /api/v4/groups/{id}/integrations/telegram`

Creates or updates the Telegram integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	URL of the Telegram API server, including `http://` or `https://`. The default value is `https://api.telegram.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target chat or the username of the target channel (in the format `@channelusername`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`thread`	String	Unique identifier for the target message thread (topic in a forum supergroup)
`token` Required	String	The Telegram bot token (for example, `123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Unify Circuit integration

`PUT /api/v4/groups/{id}/integrations/unify-circuit`

Creates or updates the Unify Circuit integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Unify Circuit webhook (for example, `https://circuit.com/rest/v2/webhooks/incoming/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Webex Teams integration

`PUT /api/v4/groups/{id}/integrations/webex-teams`

Creates or updates the Webex Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Webex Teams webhook. For example, https://api.ciscospark.com/v1/webhooks/incoming/…
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Youtrack integration

`PUT /api/v4/groups/{id}/integrations/youtrack`

Creates or updates the Youtrack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Zentao integration

`PUT /api/v4/groups/{id}/integrations/zentao`

Creates or updates the Zentao integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	—
`api_url`	String	If different from Web URL
`url` Required	String	Base URL of the ZenTao instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`zentao_product_xid` Required	String	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve integration settings

`GET /api/v4/groups/{id}/integrations/{slug}`

Retrieves the settings for a specified integration.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegration
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Disable an integration

`DELETE /api/v4/groups/{id}/integrations/{slug}`

Disables a specified integration. Integration settings are preserved.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Receive Slack events

`POST /api/v4/integrations/slack/events`

Request body (`application/json`)

Property	Type	Description
`api_app_id`	String	ID of the Slack app
`authed_users`	Array of strings	Array of Slack user IDs. Deprecated by Slack
`event`	Object	Event object with variable properties
`event_id`	String	ID of the event
`event_time`	Integer	Unix timestamp, in seconds, of when the event was dispatched
`team_id`	String	ID of the Slack workspace where the event occurred
`token`	String	Request token. Not used by Git Lab. Deprecated by Slack
`type`	String	Type of the event, usually `event_callback`

Responses

Code	Description	Schema
`200`	Successfully processed event	—
`204`	Failed to process event	—
`400`	Bad Request	—
`401`	Unauthorized	—

Process Slack interaction events

`POST /api/v4/integrations/slack/interactions`

Processes interaction events from Slack

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Get Slack interactive component options

`POST /api/v4/integrations/slack/options`

Retrieves options for Slack interactive components

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

List all active integrations

`GET /api/v4/projects/{id}/integrations`

Lists all active integrations.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesIntegration Basic
`401`	Unauthorized	—
`404`	Not found	—

Create or update the Apple App Store integration

`PUT /api/v4/projects/{id}/integrations/apple-app-store`

Creates or updates the Apple App Store integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`app_store_issuer_id` Required	String	Apple App Store Connect issuer ID
`app_store_key_id` Required	String	Apple App Store Connect key ID
`app_store_private_key` Required	String	Apple App Store Connect private key
`app_store_private_key_file_name` Required	String	Apple App Store Connect private key file name
`app_store_protected_refs`	Boolean	Set variables on protected branches and tags only
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Asana integration

`PUT /api/v4/projects/{id}/integrations/asana`

Creates or updates the Asana integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	User API token. The user must have access to the task. All comments are attributed to this user
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to be automatically inspected. Leave blank to include all branches
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Assembla integration

`PUT /api/v4/projects/{id}/integrations/assembla`

Creates or updates the Assembla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`subdomain`	String	The subdomain setting
`token` Required	String	The authentication token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bamboo integration

`PUT /api/v4/projects/{id}/integrations/bamboo`

Creates or updates the Bamboo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`bamboo_url` Required	String	Bamboo root URL (for example, `https://bamboo.example.com`)
`build_key` Required	String	Bamboo build plan key (for example, `KEY`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`password` Required	String	Password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	User with API access to the Bamboo server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bugzilla integration

`PUT /api/v4/projects/{id}/integrations/bugzilla`

Creates or updates the Bugzilla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Buildkite integration

`PUT /api/v4/projects/{id}/integrations/buildkite`

Creates or updates the Buildkite integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	DEPRECATED: This parameter has no effect because SSL verification is always enabled
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`project_url` Required	String	Pipeline URL (for example, `https://buildkite.com/example/pipeline`)
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Token you get after you create a Buildkite pipeline with a Deckhouse Code repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Campfire integration

`PUT /api/v4/projects/{id}/integrations/campfire`

Creates or updates the Campfire integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`room`	String	ID portion of the Campfire room URL
`subdomain`	String	`.campfirenow.com` subdomain when you're signed in
`token` Required	String	API authentication token from Campfire. To get the token, sign in to Campfire and select **My info**
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Clickup integration

`PUT /api/v4/projects/{id}/integrations/clickup`

Creates or updates the Clickup integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Code Scoring integration

`PUT /api/v4/projects/{id}/integrations/code-scoring`

Creates or updates the Code Scoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`codescoring_agent_image`	String	Optional. The image the Code Scoring scan jobs run in, with registry and tag or digest. It needs curl, ca-certificates, jq, ruby and git; the johnny agent is downloaded into it at run time. Left empty, the jobs use the default image. For an image in a private registry, set the registry login and password variables in the scan execution policy
`codescoring_bom_format`	String	Optional. Which CycloneDX version the bill of materials is written in. They all list the same components: pick an older version for a tool downstream that needs one, or the extended 1.6 for CodeScoring's own GOST properties. A Scan Execution Policy can name a format of its own, which wins. Left empty here and in the policy, the scan jobs write CycloneDX 1.6 JSON
`codescoring_ca_cert`	String	PEM certificate of the authority that signed the server's certificate. Used only when the trust setting above says so
`codescoring_cloud_resolve`	Boolean	Let the CodeScoring server work out the dependency tree instead of the runner. The only resolver that needs no package manager in the scan job image, and it finds the transitive packages a manifest read misses. Known cost: the server currently returns the

Property	Type	Description
		vulnerability report empty when this is on, so findings reach the tests summary but not the vulnerability report. Leave it off unless the dependency list matters more
`codescoring_extra_args`	String	Optional. Passed to the agent as written, for options this form does not name, such as per-manager arguments. A setting above wins over the same option written here
`codescoring_ignore_paths`	String	Optional. One glob per line. Keeps test fixtures and vendored trees out of the results
`codescoring_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`codescoring_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`codescoring_scan_mode`	String	What to scan: a source directory, a container image, or

Property	Type	Description
		an existing CycloneDX document. Defaults to the source directory
`codescoring_scan_target`	String	Optional. Directory, image reference or SBOM path, depending on the scan mode. Defaults to the whole checkout
`codescoring_stage`	String	The policy stage this scan reports as. The agent takes one, so the scan reports as one. Defaults to “build”
`codescoring_tls_mode`	String	How the scan job decides whether to trust the Code Scoring server. It applies to the requests the job makes with curl – fetching the agent, and the connection test above. The johnny agent itself does not verify the certificate at all and offers no way to make it, so nothing here changes what the scan itself trusts
`codescoring_token` Required	String	API token to authenticate and download the Johnny agent
`codescoring_url` Required	String	Base URL of the CodeScoring instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Confluence integration

`PUT /api/v4/projects/{id}/integrations/confluence`

Creates or updates the Confluence integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`confluence_url` Required	String	URL of the Confluence Workspace hosted on `atlassian.net`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Custom Issue Tracker integration

`PUT /api/v4/projects/{id}/integrations/custom-issue-tracker`

Creates or updates the Custom Issue Tracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Datadog integration

`PUT /api/v4/projects/{id}/integrations/datadog`

Creates or updates the Datadog integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	API key used for authentication with Datadog
`api_url`	String	Full URL of your Datadog site. Only required if you do not use a standard Datadog site
`archive_trace_events`	Boolean	When enabled, job logs are collected by Datadog and displayed along with pipeline execution traces
`build_events`	Boolean	Trigger event when a build is created
`datadog_ci_visibility`	Boolean	Enable CI Visibility
`datadog_env`	String	For self-managed deployments, `env` tag for all the data sent to Datadog
`datadog_service`	String	Tag all pipeline data from this Deckhouse Code instance in Datadog. Can be used when managing several self-managed deployments
`datadog_site`	String	Datadog site to send data to. Learn more about Datadog sites in the documentation
`datadog_tags`	String	Custom tags in Datadog. Specify one tag per line in the format `key:value\nkey2:value2`
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`project_events`	Boolean	—
`push_events`	Boolean	Trigger event for pushes to the repository
`subgroup_events`	Boolean	—
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Defect Dojo integration

`PUT /api/v4/projects/{id}/integrations/defect-dojo`

Creates or updates the Defect Dojo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	Token used to authenticate with the DefectDojo API
`auto_create_context`	Boolean	—
`ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`close_old_findings`	Boolean	—
`engagement_name`	String	DefectDojo engagement. Leave blank to use the branch name
`findings_active`	Boolean	—
`findings_verified`	Boolean	—
`minimum_severity`	String	Ignore findings below this severity on import
`product_name`	String	DefectDojo product. Leave blank to use the project full path
`product_type_name`	String	DefectDojo product type, used when a product is auto-created
`tls_mode`	String	How this instance decides whether to trust the DefectDojo server. Unlike a scan, this connection is made from here rather than from a job, so verification can be turned off – which leaves the upload open to interception, and the reports carry everything the scanners found
`url` Required	String	Base URL of your DefectDojo instance
`use_inherited_settings`	Boolean	

Property	Type	Description
		Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Diffblue Cover integration

`PUT /api/v4/projects/{id}/integrations/diffblue-cover`

Creates or updates the Diffblue Cover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`diffblue_access_token_name` Required	String	Access token name used by Diffblue Cover in pipelines
`diffblue_access_token_secret` Required	String	Access token secret used by Diffblue Cover in pipelines
`diffblue_license_key` Required	String	Diffblue Cover license key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Discord integration

`PUT /api/v4/projects/{id}/integrations/discord`

Creates or updates the Discord integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`issue_channel`	String	The name of the channel to receive issues_events notifications

Property	Type	Description
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`webhook` Required	String	Discord webhook (for example, `https://discord.com/api/webhooks/…`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Drone Ci integration

`PUT /api/v4/projects/{id}/integrations/drone-ci`

Creates or updates the Drone Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`drone_url` Required	String	Drone CI URL (for example, `http://drone.example.com`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Drone CI token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Emails On Push integration

`PUT /api/v4/projects/{id}/integrations/emails-on-push`

Creates or updates the Emails On Push integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`disable_diffs`	Boolean	Disable code diffs
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Emails separated by whitespace
`send_from_committer_email`	Boolean	Send from committer
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ewm integration

`PUT /api/v4/projects/{id}/integrations/ewm`

Creates or updates the Ewm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the External Wiki integration

`PUT /api/v4/projects/{id}/integrations/external-wiki`

Creates or updates the External Wiki integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`external_wiki_url` Required	String	URL of the external wiki
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Gitlab Slack Application integration

`PUT /api/v4/projects/{id}/integrations/gitlab-slack-application`

Creates or updates the Gitlab Slack Application integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Google Play integration

`PUT /api/v4/projects/{id}/integrations/google-play`

Creates or updates the Google Play integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`google_play_protected_refs`	Boolean	Set variables on protected branches and tags only
`package_name` Required	String	Package name of the app in Google Play
`service_account_key` Required	String	Google Play service account key
`service_account_key_file_name` Required	String	File name of the Google Play service account key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Hangouts Chat integration

`PUT /api/v4/projects/{id}/integrations/hangouts-chat`

Creates or updates the Hangouts Chat integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Hangouts Chat webhook (for example, `https://chat.googleapis.com/v1/spaces...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Harbor integration

`PUT /api/v4/projects/{id}/integrations/harbor`

Creates or updates the Harbor integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`password` Required	String	The password of the user
`project_name` Required	String	The name of the project in the Harbor instance. For example, `testproject`
`url` Required	String	The base URL to the Harbor instance linked to the Deckhouse Code project. For example, `https://demo.goharbor.io`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	The username created in the Harbor interface

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Irker integration

`PUT /api/v4/projects/{id}/integrations/irker`

Creates or updates the Irker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`colorize_messages`	Boolean	Colorize messages
`default_irc_uri`	String	URI to add before each recipient. The default value is `irc://irc.network.net:6697/`
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Comma-separated list of channels or email addresses
`server_host`	String	irker daemon hostname. The default value is `localhost`
`server_port`	Integer	irker daemon port. The default value is `6659`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jaga integration

`PUT /api/v4/projects/{id}/integrations/jaga`

Creates or updates the Jaga integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	If different from the Web URL followed by /backend
`comment_detail`	String	—
`commit_events`	Boolean	—
`issues_events`	Boolean	—
`jaga_issue_prefix`	String	Text that must come right before a task code
`jaga_issue_regex`	String	Regular expression in RE2 syntax that matches Jaga task codes. If empty, codes like ASDF-1 are matched
`jaga_issue_transition_automatic`	Boolean	—
`jaga_issue_transition_statuses`	String	Status names in workflow order, separated by commas. The task is moved along them, as far as it can go at each step, until it is done. Used when the automatic transition is off. Leave both empty to keep task statuses unchanged
`jaga_password` Required	String	Password of that Jaga user
`jaga_smart_commit_comment`	Boolean	—
`jaga_smart_commit_status`	Boolean	—
`jaga_smart_commit_time`	Boolean	—
`jaga_smart_commits_enabled`	Boolean	—
`jaga_time_activity_type`	String	Jaga activity type code the time is logged with, for example develop
	String	

Property	Type	Description
`jaga_username` Required		Email of the Jaga user the integration acts as
`merge_requests_events`	Boolean	—
`url` Required	String	Base URL of Jaga. With http://, the password and the tokens are sent unencrypted
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jenkins integration

`PUT /api/v4/projects/{id}/integrations/jenkins`

Creates or updates the Jenkins integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`jenkins_url` Required	String	URL of the Jenkins server
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password`	String	Password of the Jenkins server
`project_name` Required	String	Name of the Jenkins project
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Username of the Jenkins server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira integration

`PUT /api/v4/projects/{id}/integrations/jira`

Creates or updates the Jira integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	The base URL to the Jira instance API. Web URL value is used if not set (for example, `https://jira-api.example.com`)
`comment_on_event_enabled`	Boolean	Enable comments inside Jira issues on each GitLab event (commit / merge request)
`commit_events`	Boolean	Trigger event when a commit is created or updated
`issues_enabled`	String	Enable viewing Jira issues in Deckhouse Code
`jira_auth_type`	Integer	The authentication method to use with Jira. Use `0` for basic authentication, `1` for Jira personal access token, and `2` for Jira Cloud service accounts. Defaults to `0`
`jira_issue_prefix`	String	Prefix to match Jira issue keys
`jira_issue_regex`	String	Regular expression to match Jira issue keys
`jira_issue_transition_id`	String	The ID of one or more transitions for custom issue transitions.Ignored when `jira_issue_transition_automatic` is enabled. Defaults to a blank string,which disables custom transitions
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The Jira API token, password, or personal access token to use with Jira. When using Basic Authentication (`jira_`

Property	Type	Description
		`auth_type` is `0`), use an API token for Jira Cloud, and a password for Jira Data Center or Jira Server. For a Jira personal access token (`jira_auth_type` is `1`), use the personal access token
`project_keys`	Array of strings	Keys of Jira projects to display. When `issues_enabled` is `true`, this setting filters which Jira projects are shown in Deckhouse Code. It does not restrict the API token's access
`url` Required	String	The URL to the Jira project which is being linked to this Deckhouse Code project (for example, `https://jira.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	The email or username to use with Jira. Use an email for Jira Cloud, and a username for Jira Data Center and Jira Server. Required when using basic authentication (`jira_auth_type` is `0`)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira Cloud App integration

`PUT /api/v4/projects/{id}/integrations/jira-cloud-app`

Creates or updates the Jira Cloud App integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`jira_cloud_app_deployment_gating_environments`	String	Enter the environment (production,staging,testing, development) where you want to enable deployment gating. Use comma (,) to separate multiple environments
`jira_cloud_app_enable_deployment_gating`	Boolean	Enable to approve or reject blocked Deckhouse Code deployments from Jira Service Management
`jira_cloud_app_service_ids`	String	Copy and paste your JSM Service ID here. Use comma (,) to separate multiple IDs
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Linear integration

`PUT /api/v4/projects/{id}/integrations/linear`

Creates or updates the Linear integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`workspace_url` Required	String	Linear workspace URL (for example, https://linear.app/example)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Matrix integration

`PUT /api/v4/projects/{id}/integrations/matrix`

Creates or updates the Matrix integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	Custom hostname of the Matrix server. The default value is `https://matrix-client.matrix.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target room (in the format `!qPKKM111FFKKsfoCVy:matrix.org`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	The Matrix access token (for example, `syt-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost integration

`PUT /api/v4/projects/{id}/integrations/mattermost`

Creates or updates the Mattermost integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`incident_events`	Boolean	

Property	Type	Description
		Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Mattermost notifications username
`webhook` Required	String	Mattermost notifications webhook (for example, `http://mattermost.example.com/hooks/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost Slash Commands integration

`PUT /api/v4/projects/{id}/integrations/mattermost-slash-commands`

Creates or updates the Mattermost Slash Commands integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`token` Required	String	The Mattermost token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Trigger a slash command for mattermost-slash-commands

`POST /api/v4/projects/{id}/integrations/mattermost_slash_commands/trigger`

Triggers a slash command for mattermost-slash-commands.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`token` Required	String	The Mattermost token

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create or update the Microsoft Teams integration

`PUT /api/v4/projects/{id}/integrations/microsoft-teams`

Creates or updates the Microsoft Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Microsoft Teams webhook (for example, `https://outlook.office.com/webhook/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Ci integration

`PUT /api/v4/projects/{id}/integrations/mock-ci`

Creates or updates the Mock Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`mock_service_url` Required	String	URL of the Mock CI integration
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Monitoring integration

`PUT /api/v4/projects/{id}/integrations/mock-monitoring`

Creates or updates the Mock Monitoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Packagist integration

`PUT /api/v4/projects/{id}/integrations/packagist`

Creates or updates the Packagist integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`server`	String	URL of the Packagist server. The default value is `https://packagist.org`
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	API token of the Packagist server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	Username of a Packagist account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Phorge integration

`PUT /api/v4/projects/{id}/integrations/phorge`

Creates or updates the Phorge integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pipelines Email integration

`PUT /api/v4/projects/{id}/integrations/pipelines-email`

Creates or updates the Pipelines Email integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`notify_child_pipelines`	Boolean	Send notifications for child pipelines
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_default_branch`	Boolean	Send notifications for the default branch
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`recipients` Required	String	Comma-separated list of recipient email addresses
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pivotaltracker integration

`PUT /api/v4/projects/{id}/integrations/pivotaltracker`

Creates or updates the Pivotaltracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to automatically inspect. Leave blank to include all branches
`token` Required	String	The Pivotal Tracker token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ptai integration

`PUT /api/v4/projects/{id}/integrations/ptai`

Creates or updates the Ptai integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`ptai_ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`ptai_image` Required	String	The image the PT AI scan job runs in: the whole reference, with a tag or a digest. A new integration is filled in with the image this product ships. It needs aictl, ruby and mktemp, and curl, python3 or wget when the policy reads its settings file from another project. For an image in a private registry, set the registry login and password variables below
`ptai_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`ptai_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`ptai_server_url` Required	String	Base URL of the PT Application Inspector server. Changing it

Property	Type	Description
		clears the saved API token: enter the token again
`ptai_tls_mode`	String	Use the certificates of the job image when they already cover the server. Otherwise paste the certificate authority below
`ptai_token` Required	String	API token used to authenticate with the PT Application Inspector server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pumble integration

`PUT /api/v4/projects/{id}/integrations/pumble`

Creates or updates the Pumble integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Pumble webhook (for example, `https://api.pumble.com/workspaces/x/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pushover integration

`PUT /api/v4/projects/{id}/integrations/pushover`

Creates or updates the Pushover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	The application key
`device`	String	Leave blank for all active devices
`priority` Required	String	The priority
`push_events`	Boolean	Trigger event for pushes to the repository
`sound`	String	The sound of the notification
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`user_key` Required	String	The user key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Redmine integration

`PUT /api/v4/projects/{id}/integrations/redmine`

Creates or updates the Redmine integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Slack integration

`PUT /api/v4/projects/{id}/integrations/slack`

Creates or updates the Slack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Slack notifications username
`webhook` Required	String	Slack notifications webhook (for example, `https://hooks.slack.com/services/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Squash Tm integration

`PUT /api/v4/projects/{id}/integrations/squash-tm`

Creates or updates the Squash Tm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`token`	String	Secret token
`url` Required	String	URL of the Squash TM webhook
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Teamcity integration

`PUT /api/v4/projects/{id}/integrations/teamcity`

Creates or updates the Teamcity integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`build_type` Required	String	The build configuration ID of the TeamCity project
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`teamcity_url` Required	String	TeamCity root URL (for example, `https://teamcity.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	A user with permissions to trigger a manual build

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Telegram integration

`PUT /api/v4/projects/{id}/integrations/telegram`

Creates or updates the Telegram integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	URL of the Telegram API server, including `http://` or `https://`. The default value is `https://api.telegram.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target chat or the username of the target channel (in the format `@channelusername`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`thread`	String	Unique identifier for the target message thread (topic in a forum supergroup)
`token` Required	String	The Telegram bot token (for example, `123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Unify Circuit integration

`PUT /api/v4/projects/{id}/integrations/unify-circuit`

Creates or updates the Unify Circuit integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Unify Circuit webhook (for example, `https://circuit.com/rest/v2/webhooks/incoming/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Webex Teams integration

`PUT /api/v4/projects/{id}/integrations/webex-teams`

Creates or updates the Webex Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Webex Teams webhook. For example, https://api.ciscospark.com/v1/webhooks/incoming/…
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Youtrack integration

`PUT /api/v4/projects/{id}/integrations/youtrack`

Creates or updates the Youtrack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Zentao integration

`PUT /api/v4/projects/{id}/integrations/zentao`

Creates or updates the Zentao integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	—
`api_url`	String	If different from Web URL
`url` Required	String	Base URL of the ZenTao instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`zentao_product_xid` Required	String	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve integration settings

`GET /api/v4/projects/{id}/integrations/{slug}`

Retrieves the settings for a specified integration.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegration
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Disable an integration

`DELETE /api/v4/projects/{id}/integrations/{slug}`

Disables a specified integration. Integration settings are preserved.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

List all active integrations

`GET /api/v4/projects/{id}/services`

Lists all active integrations.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesIntegrationBasic
`401`	Unauthorized	—
`404`	Not found	—

Create or update the Apple App Store integration

`PUT /api/v4/projects/{id}/services/apple-app-store`

Creates or updates the Apple App Store integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`app_store_issuer_id` Required	String	Apple App Store Connect issuer ID
`app_store_key_id` Required	String	Apple App Store Connect key ID
`app_store_private_key` Required	String	Apple App Store Connect private key
`app_store_private_key_file_name` Required	String	Apple App Store Connect private key file name
`app_store_protected_refs`	Boolean	Set variables on protected branches and tags only
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Asana integration

`PUT /api/v4/projects/{id}/services/asana`

Creates or updates the Asana integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	User API token. The user must have access to the task. All comments are attributed to this user
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to be automatically inspected. Leave blank to include all branches
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Assembla integration

`PUT /api/v4/projects/{id}/services/assembla`

Creates or updates the Assembla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`subdomain`	String	The subdomain setting
`token` Required	String	The authentication token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bamboo integration

`PUT /api/v4/projects/{id}/services/bamboo`

Creates or updates the Bamboo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`bamboo_url` Required	String	Bamboo root URL (for example, `https://bamboo.example.com`)
`build_key` Required	String	Bamboo build plan key (for example, `KEY`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`password` Required	String	Password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	User with API access to the Bamboo server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Bugzilla integration

`PUT /api/v4/projects/{id}/services/bugzilla`

Creates or updates the Bugzilla integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Buildkite integration

`PUT /api/v4/projects/{id}/services/buildkite`

Creates or updates the Buildkite integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	DEPRECATED: This parameter has no effect because SSL verification is always enabled
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`project_url` Required	String	Pipeline URL (for example, `https://buildkite.com/example/pipeline`)
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Token you get after you create a Buildkite pipeline with a Deckhouse Code repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Campfire integration

`PUT /api/v4/projects/{id}/services/campfire`

Creates or updates the Campfire integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`room`	String	ID portion of the Campfire room URL
`subdomain`	String	`.campfirenow.com` subdomain when you're signed in
`token` Required	String	API authentication token from Campfire. To get the token, sign in to Campfire and select **My info**
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Clickup integration

`PUT /api/v4/projects/{id}/services/clickup`

Creates or updates the Clickup integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Code Scoring integration

`PUT /api/v4/projects/{id}/services/code-scoring`

Creates or updates the Code Scoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`codescoring_agent_image`	String	Optional. The image the Code Scoring scan jobs run in, with registry and tag or digest. It needs curl, ca-certificates, jq, ruby and git; the johnny agent is downloaded into it at run time. Left empty, the jobs use the default image. For an image in a private registry, set the registry login and password variables in the scan execution policy
`codescoring_bom_format`	String	Optional. Which CycloneDX version the bill of materials is written in. They all list the same components: pick an older version for a tool downstream that needs one, or the extended 1.6 for CodeScoring's own GOST properties. A Scan Execution Policy can name a format of its own, which wins. Left empty here and in the policy, the scan jobs write CycloneDX 1.6 JSON
`codescoring_ca_cert`	String	PEM certificate of the authority that signed the server's certificate. Used only when the trust setting above says so
`codescoring_cloud_resolve`	Boolean	Let the CodeScoring server work out the dependency tree instead of the runner. The only resolver that needs no package manager in the scan job image, and it finds the transitive packages a manifest read misses. Known cost: the server currently returns the

Property	Type	Description
		vulnerability report empty when this is on, so findings reach the tests summary but not the vulnerability report. Leave it off unless the dependency list matters more
`codescoring_extra_args`	String	Optional. Passed to the agent as written, for options this form does not name, such as per-manager arguments. A setting above wins over the same option written here
`codescoring_ignore_paths`	String	Optional. One glob per line. Keeps test fixtures and vendored trees out of the results
`codescoring_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`codescoring_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`codescoring_scan_mode`	String	What to scan: a source directory, a container image, or

Property	Type	Description
		an existing CycloneDX document. Defaults to the source directory
`codescoring_scan_target`	String	Optional. Directory, image reference or SBOM path, depending on the scan mode. Defaults to the whole checkout
`codescoring_stage`	String	The policy stage this scan reports as. The agent takes one, so the scan reports as one. Defaults to “build”
`codescoring_tls_mode`	String	How the scan job decides whether to trust the Code Scoring server. It applies to the requests the job makes with curl – fetching the agent, and the connection test above. The johnny agent itself does not verify the certificate at all and offers no way to make it, so nothing here changes what the scan itself trusts
`codescoring_token` Required	String	API token to authenticate and download the Johnny agent
`codescoring_url` Required	String	Base URL of the CodeScoring instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Confluence integration

`PUT /api/v4/projects/{id}/services/confluence`

Creates or updates the Confluence integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`confluence_url` Required	String	URL of the Confluence Workspace hosted on `atlassian.net`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Custom Issue Tracker integration

`PUT /api/v4/projects/{id}/services/custom-issue-tracker`

Creates or updates the Custom Issue Tracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Datadog integration

`PUT /api/v4/projects/{id}/services/datadog`

Creates or updates the Datadog integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	API key used for authentication with Datadog
`api_url`	String	Full URL of your Datadog site. Only required if you do not use a standard Datadog site
`archive_trace_events`	Boolean	When enabled, job logs are collected by Datadog and displayed along with pipeline execution traces
`build_events`	Boolean	Trigger event when a build is created
`datadog_ci_visibility`	Boolean	Enable CI Visibility
`datadog_env`	String	For self-managed deployments, `env` tag for all the data sent to Datadog
`datadog_service`	String	Tag all pipeline data from this Deckhouse Code instance in Datadog. Can be used when managing several self-managed deployments
`datadog_site`	String	Datadog site to send data to. Learn more about Datadog sites in the documentation
`datadog_tags`	String	Custom tags in Datadog. Specify one tag per line in the format `key:value\nkey2:value2`
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`project_events`	Boolean	—
`push_events`	Boolean	Trigger event for pushes to the repository
`subgroup_events`	Boolean	—
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Defect Dojo integration

`PUT /api/v4/projects/{id}/services/defect-dojo`

Creates or updates the Defect Dojo integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	Token used to authenticate with the DefectDojo API
`auto_create_context`	Boolean	—
`ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`close_old_findings`	Boolean	—
`engagement_name`	String	DefectDojo engagement. Leave blank to use the branch name
`findings_active`	Boolean	—
`findings_verified`	Boolean	—
`minimum_severity`	String	Ignore findings below this severity on import
`product_name`	String	DefectDojo product. Leave blank to use the project full path
`product_type_name`	String	DefectDojo product type, used when a product is auto-created
`tls_mode`	String	How this instance decides whether to trust the DefectDojo server. Unlike a scan, this connection is made from here rather than from a job, so verification can be turned off – which leaves the upload open to interception, and the reports carry everything the scanners found
`url` Required	String	Base URL of your DefectDojo instance
`use_inherited_settings`	Boolean	

Property	Type	Description
		Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Diffblue Cover integration

`PUT /api/v4/projects/{id}/services/diffblue-cover`

Creates or updates the Diffblue Cover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`diffblue_access_token_name` Required	String	Access token name used by Diffblue Cover in pipelines
`diffblue_access_token_secret` Required	String	Access token secret used by Diffblue Cover in pipelines
`diffblue_license_key` Required	String	Diffblue Cover license key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Discord integration

`PUT /api/v4/projects/{id}/services/discord`

Creates or updates the Discord integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`issue_channel`	String	The name of the channel to receive issues_events notifications

Property	Type	Description
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`webhook` Required	String	Discord webhook (for example, `https://discord.com/api/webhooks/…`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Drone Ci integration

`PUT /api/v4/projects/{id}/services/drone-ci`

Creates or updates the Drone Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`drone_url` Required	String	Drone CI URL (for example, `http://drone.example.com`)
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	Drone CI token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Emails On Push integration

`PUT /api/v4/projects/{id}/services/emails-on-push`

Creates or updates the Emails On Push integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`disable_diffs`	Boolean	Disable code diffs
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Emails separated by whitespace
`send_from_committer_email`	Boolean	Send from committer
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ewm integration

`PUT /api/v4/projects/{id}/services/ewm`

Creates or updates the Ewm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the External Wiki integration

`PUT /api/v4/projects/{id}/services/external-wiki`

Creates or updates the External Wiki integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`external_wiki_url` Required	String	URL of the external wiki
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Gitlab Slack Application integration

`PUT /api/v4/projects/{id}/services/gitlab-slack-application`

Creates or updates the Gitlab Slack Application integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Google Play integration

`PUT /api/v4/projects/{id}/services/google-play`

Creates or updates the Google Play integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`google_play_protected_refs`	Boolean	Set variables on protected branches and tags only
`package_name` Required	String	Package name of the app in Google Play
`service_account_key` Required	String	Google Play service account key
`service_account_key_file_name` Required	String	File name of the Google Play service account key
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Hangouts Chat integration

`PUT /api/v4/projects/{id}/services/hangouts-chat`

Creates or updates the Hangouts Chat integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Hangouts Chat webhook (for example, `https://chat.googleapis.com/v1/spaces...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Harbor integration

`PUT /api/v4/projects/{id}/services/harbor`

Creates or updates the Harbor integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`password` Required	String	The password of the user
`project_name` Required	String	The name of the project in the Harbor instance. For example, `testproject`
`url` Required	String	The base URL to the Harbor instance linked to the Deckhouse Code project. For example, `https://demo.goharbor.io`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	The username created in the Harbor interface

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Irker integration

`PUT /api/v4/projects/{id}/services/irker`

Creates or updates the Irker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`colorize_messages`	Boolean	Colorize messages
`default_irc_uri`	String	URI to add before each recipient. The default value is `irc://irc.network.net:6697/`
`push_events`	Boolean	Trigger event for pushes to the repository
`recipients` Required	String	Comma-separated list of channels or email addresses
`server_host`	String	irker daemon hostname. The default value is `localhost`
`server_port`	Integer	irker daemon port. The default value is `6659`
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jaga integration

`PUT /api/v4/projects/{id}/services/jaga`

Creates or updates the Jaga integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	If different from the Web URL followed by /backend
`comment_detail`	String	—
`commit_events`	Boolean	—
`issues_events`	Boolean	—
`jaga_issue_prefix`	String	Text that must come right before a task code
`jaga_issue_regex`	String	Regular expression in RE2 syntax that matches Jaga task codes. If empty, codes like ASDF-1 are matched
`jaga_issue_transition_automatic`	Boolean	—
`jaga_issue_transition_statuses`	String	Status names in workflow order, separated by commas. The task is moved along them, as far as it can go at each step, until it is done. Used when the automatic transition is off. Leave both empty to keep task statuses unchanged
`jaga_password` Required	String	Password of that Jaga user
`jaga_smart_commit_comment`	Boolean	—
`jaga_smart_commit_status`	Boolean	—
`jaga_smart_commit_time`	Boolean	—
`jaga_smart_commits_enabled`	Boolean	—
`jaga_time_activity_type`	String	Jaga activity type code the time is logged with, for example develop
	String	

Property	Type	Description
`jaga_username` Required		Email of the Jaga user the integration acts as
`merge_requests_events`	Boolean	—
`url` Required	String	Base URL of Jaga. With http://, the password and the tokens are sent unencrypted
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jenkins integration

`PUT /api/v4/projects/{id}/services/jenkins`

Creates or updates the Jenkins integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`jenkins_url` Required	String	URL of the Jenkins server
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password`	String	Password of the Jenkins server
`project_name` Required	String	Name of the Jenkins project
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Username of the Jenkins server

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira integration

`PUT /api/v4/projects/{id}/services/jira`

Creates or updates the Jira integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_url`	String	The base URL to the Jira instance API. Web URL value is used if not set (for example, `https://jira-api.example.com`)
`comment_on_event_enabled`	Boolean	Enable comments inside Jira issues on each GitLab event (commit / merge request)
`commit_events`	Boolean	Trigger event when a commit is created or updated
`issues_enabled`	String	Enable viewing Jira issues in Deckhouse Code
`jira_auth_type`	Integer	The authentication method to use with Jira. Use `0` for basic authentication, `1` for Jira personal access token, and `2` for Jira Cloud service accounts. Defaults to `0`
`jira_issue_prefix`	String	Prefix to match Jira issue keys
`jira_issue_regex`	String	Regular expression to match Jira issue keys
`jira_issue_transition_id`	String	The ID of one or more transitions for custom issue transitions.Ignored when `jira_issue_transition_automatic` is enabled. Defaults to a blank string,which disables custom transitions
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The Jira API token, password, or personal access token to use with Jira. When using Basic Authentication (`jira_`

Property	Type	Description
		`auth_type` is `0`), use an API token for Jira Cloud, and a password for Jira Data Center or Jira Server. For a Jira personal access token (`jira_auth_type` is `1`), use the personal access token
`project_keys`	Array of strings	Keys of Jira projects to display. When `issues_enabled` is `true`, this setting filters which Jira projects are shown in Deckhouse Code. It does not restrict the API token's access
`url` Required	String	The URL to the Jira project which is being linked to this Deckhouse Code project (for example, `https://jira.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	The email or username to use with Jira. Use an email for Jira Cloud, and a username for Jira Data Center and Jira Server. Required when using basic authentication (`jira_auth_type` is `0`)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Jira Cloud App integration

`PUT /api/v4/projects/{id}/services/jira-cloud-app`

Creates or updates the Jira Cloud App integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`jira_cloud_app_deployment_gating_environments`	String	Enter the environment (production,staging,testing, development) where you want to enable deployment gating. Use comma (,) to separate multiple environments
`jira_cloud_app_enable_deployment_gating`	Boolean	Enable to approve or reject blocked Deckhouse Code deployments from Jira Service Management
`jira_cloud_app_service_ids`	String	Copy and paste your JSM Service ID here. Use comma (,) to separate multiple IDs
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Linear integration

`PUT /api/v4/projects/{id}/services/linear`

Creates or updates the Linear integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`workspace_url` Required	String	Linear workspace URL (for example, https://linear.app/example)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Matrix integration

`PUT /api/v4/projects/{id}/services/matrix`

Creates or updates the Matrix integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	Custom hostname of the Matrix server. The default value is `https://matrix-client.matrix.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target room (in the format `!qPKKM111FFKKsfoCVy:matrix.org`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	The Matrix access token (for example, `syt-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost integration

`PUT /api/v4/projects/{id}/services/mattermost`

Creates or updates the Mattermost integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	The name of the channel to receive incident_events notifications
`incident_events`	Boolean	

Property	Type	Description
		Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	

Property	Type	Description
		Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Mattermost notifications username
`webhook` Required	String	Mattermost notifications webhook (for example, `http://mattermost.example.com/hooks/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mattermost Slash Commands integration

`PUT /api/v4/projects/{id}/services/mattermost-slash-commands`

Creates or updates the Mattermost Slash Commands integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`token` Required	String	The Mattermost token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Trigger a slash command for mattermost-slash-commands

`POST /api/v4/projects/{id}/services/mattermost_slash_commands/trigger`

Triggers a slash command for mattermost-slash-commands.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`token` Required	String	The Mattermost token

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create or update the Microsoft Teams integration

`PUT /api/v4/projects/{id}/services/microsoft-teams`

Creates or updates the Microsoft Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Microsoft Teams webhook (for example, `https://outlook.office.com/webhook/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Ci integration

`PUT /api/v4/projects/{id}/services/mock-ci`

Creates or updates the Mock Ci integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`mock_service_url` Required	String	URL of the Mock CI integration
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Mock Monitoring integration

`PUT /api/v4/projects/{id}/services/mock-monitoring`

Creates or updates the Mock Monitoring integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Packagist integration

`PUT /api/v4/projects/{id}/services/packagist`

Creates or updates the Packagist integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`push_events`	Boolean	Trigger event for pushes to the repository
`server`	String	URL of the Packagist server. The default value is `https://packagist.org`
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`token` Required	String	API token of the Packagist server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	Username of a Packagist account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Phorge integration

`PUT /api/v4/projects/{id}/services/phorge`

Creates or updates the Phorge integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pipelines Email integration

`PUT /api/v4/projects/{id}/services/pipelines-email`

Creates or updates the Pipelines Email integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`notify_child_pipelines`	Boolean	Send notifications for child pipelines
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_default_branch`	Boolean	Send notifications for the default branch
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`recipients` Required	String	Comma-separated list of recipient email addresses
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pivotaltracker integration

`PUT /api/v4/projects/{id}/services/pivotaltracker`

Creates or updates the Pivotaltracker integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`push_events`	Boolean	Trigger event for pushes to the repository
`restrict_to_branch`	String	Comma-separated list of branches to automatically inspect. Leave blank to include all branches
`token` Required	String	The Pivotal Tracker token
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Ptai integration

`PUT /api/v4/projects/{id}/services/ptai`

Creates or updates the Ptai integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`ptai_ca_cert`	String	PEM certificate of the authority that signed the server’s certificate. Used only when the trust setting above says so
`ptai_image` Required	String	The image the PT AI scan job runs in: the whole reference, with a tag or a digest. A new integration is filled in with the image this product ships. It needs aictl, ruby and mktemp, and curl, python3 or wget when the policy reads its settings file from another project. For an image in a private registry, set the registry login and password variables below
`ptai_registry_login_variable`	String	Optional. The name of a CI/CD variable of the project or its group that holds the login for the registry of the scanner image, such as $SCANNER_REGISTRY_USER. Use regular variables, not file variables. Fill in both variables, or neither when the runner or the project can already pull the image
`ptai_registry_password_variable`	String	Optional. The name of the CI/CD variable that holds the password or token for that registry, such as $SCANNER_REGISTRY_PASSWORD. Mask the variable. The password cannot contain a " or \ character
`ptai_server_url` Required	String	Base URL of the PT Application Inspector server. Changing it

Property	Type	Description
		clears the saved API token: enter the token again
`ptai_tls_mode`	String	Use the certificates of the job image when they already cover the server. Otherwise paste the certificate authority below
`ptai_token` Required	String	API token used to authenticate with the PT Application Inspector server
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pumble integration

`PUT /api/v4/projects/{id}/services/pumble`

Creates or updates the Pumble integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Pumble webhook (for example, `https://api.pumble.com/workspaces/x/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Pushover integration

`PUT /api/v4/projects/{id}/services/pushover`

Creates or updates the Pushover integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_key` Required	String	The application key
`device`	String	Leave blank for all active devices
`priority` Required	String	The priority
`push_events`	Boolean	Trigger event for pushes to the repository
`sound`	String	The sound of the notification
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`user_key` Required	String	The user key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Redmine integration

`PUT /api/v4/projects/{id}/services/redmine`

Creates or updates the Redmine integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`new_issue_url` Required	String	URL of the new issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Slack integration

`PUT /api/v4/projects/{id}/services/slack`

Creates or updates the Slack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`alert_channel`	String	The name of the channel to receive alert_events notifications
`alert_events`	Boolean	Trigger event when a new, unique alert is recorded
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`channel`	String	Default channel to use if no other channel is configured
`confidential_issue_channel`	String	The name of the channel to receive confidential_issues_ events notifications
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_channel`	String	The name of the channel to receive confidential_note_ events notifications
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`deployment_channel`	String	The name of the channel to receive deployment_events notifications
`deployment_events`	Boolean	Trigger event when a deployment starts or finishes
`incident_channel`	String	

Property	Type	Description
		The name of the channel to receive incident_events notifications
`incident_events`	Boolean	Trigger event when an incident is created
`issue_channel`	String	The name of the channel to receive issues_events notifications
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`labels_to_be_notified`	String	Labels to send notifications for. Leave blank to receive notifications for all events
`labels_to_be_notified_behavior`	String	Labels to be notified for. Valid options are `match_any` and `match_all`. The default value is `match_any`
`merge_request_channel`	String	The name of the channel to receive merge_requests_ events notifications
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_channel`	String	The name of the channel to receive note_events notifications
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_channel`	String	

Property	Type	Description
		The name of the channel to receive pipeline_events notifications
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_channel`	String	The name of the channel to receive push_events notifications
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_channel`	String	The name of the channel to receive tag_push_events notifications
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username`	String	Slack notifications username
`webhook` Required	String	Slack notifications webhook (for example, `https://hooks.slack.com/services/...`)
`wiki_page_channel`	String	The name of the channel to receive wiki_page_events notifications
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Squash Tm integration

`PUT /api/v4/projects/{id}/services/squash-tm`

Creates or updates the Squash Tm integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`token`	String	Secret token
`url` Required	String	URL of the Squash TM webhook
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Teamcity integration

`PUT /api/v4/projects/{id}/services/teamcity`

Creates or updates the Teamcity integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`build_type` Required	String	The build configuration ID of the TeamCity project
`enable_ssl_verification`	Boolean	Enable SSL verification. Defaults to `true` (enabled)
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`password` Required	String	The password of the user
`push_events`	Boolean	Trigger event for pushes to the repository
`teamcity_url` Required	String	TeamCity root URL (for example, `https://teamcity.example.com`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`username` Required	String	A user with permissions to trigger a manual build

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Telegram integration

`PUT /api/v4/projects/{id}/services/telegram`

Creates or updates the Telegram integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`hostname`	String	URL of the Telegram API server, including `http://` or `https://`. The default value is `https://api.telegram.org`
`incident_events`	Boolean	Trigger event when an incident is created
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	

Property	Type	Description
		Trigger event for pushes to the repository
`room` Required	String	Unique identifier for the target chat or the username of the target channel (in the format `@channelusername`)
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`thread`	String	Unique identifier for the target message thread (topic in a forum supergroup)
`token` Required	String	The Telegram bot token (for example, `123456:ABC-DEF1234ghIkl-zyx57W2v1u123ew11`)
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Unify Circuit integration

`PUT /api/v4/projects/{id}/services/unify-circuit`

Creates or updates the Unify Circuit integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Unify Circuit webhook (for example, `https://circuit.com/rest/v2/webhooks/incoming/...`)
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Webex Teams integration

`PUT /api/v4/projects/{id}/services/webex-teams`

Creates or updates the Webex Teams integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`branches_to_be_notified`	String	Branches to send notifications for. Valid options are `all`, `default`, `protected`, and `default_and_protected`. The default value is `default`
`confidential_issues_events`	Boolean	Trigger event when a confidential work item is created, updated, or closed
`confidential_note_events`	Boolean	Trigger event for new comments on confidential work items
`confidential_work_item_events`	Boolean	—
`issues_events`	Boolean	Trigger event when a work item is created, updated, or closed
`merge_requests_events`	Boolean	Trigger event when a merge request is created, updated, or merged
`note_events`	Boolean	Trigger event for new comments
`notify_only_broken_pipelines`	Boolean	Send notifications for broken pipelines
`notify_only_when_pipeline_status_changes`	Boolean	Send notifications only when the pipeline status changes
`pipeline_events`	Boolean	Trigger event when a pipeline status changes
`push_events`	Boolean	Trigger event for pushes to the repository
`tag_push_events`	Boolean	Trigger event for new tags pushed to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Property	Type	Description
`webhook` Required	String	The Webex Teams webhook. For example, https://api.ciscospark.com/v1/webhooks/incoming/…
`wiki_page_events`	Boolean	Trigger event when a wiki page is created or updated
`work_item_events`	Boolean	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Youtrack integration

`PUT /api/v4/projects/{id}/services/youtrack`

Creates or updates the Youtrack integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`issues_url` Required	String	URL of the issue
`project_url` Required	String	URL of the project
`push_events`	Boolean	Trigger event for pushes to the repository
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create or update the Zentao integration

`PUT /api/v4/projects/{id}/services/zentao`

Creates or updates the Zentao integration.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`api_token` Required	String	—
`api_url`	String	If different from Web URL
`url` Required	String	Base URL of the ZenTao instance
`use_inherited_settings`	Boolean	Indicates whether to inherit the default settings. Defaults to `false`
`zentao_product_xid` Required	String	—

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegrationBasic
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve integration settings

`GET /api/v4/projects/{id}/services/{slug}`

Retrieves the settings for a specified integration.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIntegration
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Disable an integration

`DELETE /api/v4/projects/{id}/services/{slug}`

Disables a specified integration. Integration settings are preserved.

Parameters

Name	Type	Description
`slug` Path, required	String	The name of the integration Allowed values: `apple-app-store`, `asana`, `assembla`, `bamboo`, `bugzilla`, `buildkite`, `campfire`, `confluence`, `custom-issue-tracker`, `datadog`, `diffblue-cover`, `discord`, `drone-ci`, `emails-on-push`, `external-wiki`, `gitlab-slack-application`, `google-play`, `hangouts-chat`, `harbor`, `irker`, `jenkins`, `jira`, `jira-cloud-app`, `linear`, `matrix`, `mattermost-slash-commands`, `packagist`, `phorge`, `pipelines-email`, `pivotaltracker`, `pumble`, `pushover`, `redmine`, `ewm`, `youtrack`, `clickup`, `slack`, `microsoft-teams`, `mattermost`, `teamcity`, `telegram`, `unify-circuit`, `webex-teams`, `zentao`, `squash-tm`, `defect-dojo`, `ptai`, `code-scoring`, `jaga`, `mock-ci`, `mock-monitoring` Minimum length: `1`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Trigger a global slack command

`POST /api/v4/slack/trigger`

Triggers a global slack command.

Request body (`application/json`)

Property	Type	Description
`text` Required	String	Text of the slack command

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesIntegration

Property	Type	Description
`active`	Boolean	—
`alert_events`	Boolean	—
`comment_on_event_enabled`	Boolean	—
`commit_events`	Boolean	—
`confidential_issues_events`	Boolean	—
`confidential_note_events`	Boolean	—
`created_at`	String (date-time)	Example: `2019-11-20T11:20:25.297Z`
`deployment_events`	Boolean	—
`id`	Integer (int64)	Example: `75`
`incident_events`	Boolean	—
`inherited`	Boolean	—
`issues_events`	Boolean	—
`job_events`	Boolean	—
`merge_requests_events`	Boolean	—
`note_events`	Boolean	—
`pipeline_events`	Boolean	—
`properties`	Object	Example: `{"token":"secr3t"}`
`push_events`	Boolean	—
`slug`	String	Example: `jenkins`
`tag_push_events`	Boolean	—
`title`	String	Example: `Jenkins CI`
`updated_at`	String (date-time)	Example: `2019-11-20T12:24:37.498Z`
`wiki_page_events`	Boolean	—

APIEntitiesIntegrationBasic

Property	Type	Description
`active`	Boolean	—
`alert_events`	Boolean	—
`comment_on_event_enabled`	Boolean	—
`commit_events`	Boolean	—
`confidential_issues_events`	Boolean	—
`confidential_note_events`	Boolean	—
`created_at`	String (date-time)	Example: `2019-11-20T11:20:25.297Z`
`deployment_events`	Boolean	—
`id`	Integer (int64)	Example: `75`
`incident_events`	Boolean	—
`inherited`	Boolean	—
`issues_events`	Boolean	—
`job_events`	Boolean	—
`merge_requests_events`	Boolean	—
`note_events`	Boolean	—
`pipeline_events`	Boolean	—
`push_events`	Boolean	—
`slug`	String	Example: `jenkins`
`tag_push_events`	Boolean	—
`title`	String	Example: `Jenkins CI`
`updated_at`	String (date-time)	Example: `2019-11-20T12:24:37.498Z`
`wiki_page_events`	Boolean	—

Internal operations

Operations related to internal operations.

Process Slack interaction events

`POST /api/v4/integrations/slack/interactions`

Processes interaction events from Slack

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Get Slack interactive component options

`POST /api/v4/integrations/slack/options`

Retrieves options for Slack interactive components

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Invitations

Use this API to manage invitations and add users to a group or project.

List all pending invitations for a group

`GET /api/v4/groups/{id}/invitations`

Lists all pending invitations for a specified group viewable by the authenticated user. Returns invitations to direct members only, and not through inherited ancestor groups. This function takes pagination parameters `page` and `per_page` to restrict the list of members.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`query` Query	String	Return only invitations sent to this email address. Must match exactly. If omitted, returns all invitations

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Add a member to a group

`POST /api/v4/groups/{id}/invitations`

Adds a member to a group. You can specify a user ID or invite a user by email.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	Access level to grant to the invited user. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`email`	Array of strings	Email address of the member to invite, or multiple email addresses separated by commas. Required if `user_id` is not provided
`expires_at`	String (date-time)	Date when the invitation expires
`invite_source`	String	Source of the invitation that starts the member creation process Default: `invitations-api`
`user_id`	Array of strings	ID of the user to invite, or multiple IDs separated by commas. Required if `email` is not provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Update an invitation to a group

`PUT /api/v4/groups/{id}/invitations/{email}`

Updates a pending invitation to a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`email` Path, required	String	Email address the invitation was previously sent to

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	Access level to grant to the invited user. For a list of values, see roles Allowed values: `10` , `15` , `20` , `25` , `30` , `40` , `50`
`expires_at`	String (date-time)	Date when the invitation expires

Responses

Code	Description	Schema
`200`	OK	APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Delete an invitation to a group

`DELETE /api/v4/groups/{id}/invitations/{email}`

Deletes a pending invitation to a specified email address for a group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`email` Path, required	String	Email address the invitation was previously sent to

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Could not delete invitation	—

List all pending invitations for a project

`GET /api/v4/projects/{id}/invitations`

Lists all pending invitations for a specified project viewable by the authenticated user. Returns invitations to direct members only, and not through inherited ancestor groups. This function takes pagination parameters `page` and `per_page` to restrict the list of members.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`query` Query	String	Return only invitations sent to this email address. Must match exactly. If omitted, returns all invitations

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Add a member to a project

`POST /api/v4/projects/{id}/invitations`

Adds a member to a project. You can specify a user ID or invite a user by email.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	Access level to grant to the invited user. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`email`	Array of strings	Email address of the member to invite, or multiple email addresses separated by commas. Required if `user_id` is not provided
`expires_at`	String (date-time)	Date when the invitation expires
`invite_source`	String	Source of the invitation that starts the member creation process Default: `invitations-api`
`user_id`	Array of strings	ID of the user to invite, or multiple IDs separated by commas. Required if `email` is not provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Update an invitation to a project

`PUT /api/v4/projects/{id}/invitations/{email}`

Updates a pending invitation to a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`email` Path, required	String	Email address the invitation was previously sent to

Request body (`application/json`)

Property	Type	Description
`access_level`	Integer	Access level to grant to the invited user. For a list of values, see roles Allowed values: `10` , `15` , `20` , `25` , `30` , `40` , `50`
`expires_at`	String (date-time)	Date when the invitation expires

Responses

Code	Description	Schema
`200`	OK	APIEntitiesInvitation
`400`	Bad Request	—
`404`	Not Found	—

Delete an invitation to a project

`DELETE /api/v4/projects/{id}/invitations/{email}`

Deletes a pending invitation to a specified email address for a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`email` Path, required	String	Email address the invitation was previously sent to

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Could not delete invitation	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesInvitation

Property	Type	Description
`access_level`	String	—
`created_at`	String	—
`created_by_name`	String	—
`expires_at`	String	—
`invite_email`	String	—
`invite_token`	String	—
`user_name`	String	—

Issues

Use this API to manage issues, their links to related issues, and issue statistics. You can:

- Create, update, and delete issues.
- Manage issue metadata, like assignees, labels, milestones, and time tracking.
- Cross-reference issues and merge requests.
- Track issue movement and promotion between projects and epics.
- Retrieve statistics about issues.

If a user is not a member of a private project, a `GET` request on that project results in a `404` status code.

> The `references.relative` attribute is relative to the group or project of the issue being requested. When an issue is fetched from its project, the `relative` format is the same as the `short` format. When requested across groups or projects, it's expected to be the same as the `full` format.

List all issues for the currently authenticated user

`GET /api/v4/issues`

Lists all issues accessible by the currently authenticated user. By default, returns only issues created by the current user. To list all issues, use parameter `scope=all`.

Parameters

Name	Type	Description
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`state` Query	String	Filter issues by state Allowed values: `opened`, `closed`, `all` Default: `all`
`closed_by_id` Query	Integer	Return issues closed by the user with the given ID
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `due_date`, `label_priority`, `milestone_due`, `popularity`, `priority`, `relative_position`, `title`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`due_date` Query	String	Return issues that have no due date, are overdue, or whose due date is this week, this month, or between two weeks ago and next month Allowed values: `0`, `any`, `today`, `tomorrow`, `overdue`, `week`, `month`, `next_month_and_previous_two_weeks`, ``
`issue_type` Query	String	The type of the issue. Accepts: issue, incident, test_case,

Name	Type	Description
		requirement, task, ticket Allowed values: `issue`, `incident`, `test_case`, `requirement`, `task`, `ticket`
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for `Upcoming` and `Started` differs from the logic used in

Name	Type	Description
		the GraphQL API. Mutually exclusive with `milestone` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid parameter error is returned. When multiple usernames are

Name	Type	Description
		given, only issues assigned to all of them are returned. Mutually exclusive with `assignee_id`
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`
	String	Return issues not authored by the user with the given

Name	Type	Description
`not[author_username]` Query		username. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer	Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `all` Default: `created_by_me`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
	Boolean	

Name	Type	Description
`non_archived` Query		If `true`, returns only issues from non-archived projects. If `false`, returns issues from both archived and non-archived projects Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—

Retrieve an issue

`GET /api/v4/issues/{id}`

Retrieves a specified issue. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	String	ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Retrieve issues statistics for the currently authenticated user

`GET /api/v4/issues_statistics`

Retrieves statistics for issues accessible by the currently authenticated user. By default, returns only issues created by the current user. To get all issues, set the `scope` attribute to `all`.

Parameters

Name	Type	Description
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for `Upcoming` and `Started` differs from the logic used in the GraphQL API. Mutually exclusive with `milestone` Allowed values: `Any`, `None`, `Upcoming`, `Started`

Name	Type	Description
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid parameter error is returned. When multiple usernames are given, only issues assigned to all of them are returned. Mutually exclusive with `assignee_id`

Name	Type	Description
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	Return issues not authored by the user with the given username. Mutually exclusive with `not[author_id]`
	Integer	

Name	Type	Description
`not[assignee_id]` Query		Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created_by_me`, `assigned_to_me`, `all` Default: `created_by_me`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—

Add spent time for an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/add_spent_time`

Adds spent time for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`duration` Required	String	The duration in human format

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all issue links

`GET /api/v4/projects/{id}/issues/{issue_iid}/links`

Lists all linked issues for a specified issue, sorted by the relationship creation datetime (ascending). Issues are filtered according to the user authorizations.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesRelated Issue
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create an issue link

`POST /api/v4/projects/{id}/issues/{issue_iid}/links`

Creates a two-way relationship between two issues. The user must be allowed to update both issues to succeed.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`link_type`	String	The type of the relation Allowed values: `relates_to` Default: `relates_to`
`target_issue_iid` Required	String or integer	Internal ID of the issue in the target project
`target_project_id` Required	String or integer	ID or URL-encoded path of the target project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssueLink
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve an issue link

`GET /api/v4/projects/{id}/issues/{issue_iid}/links/{issue_link_id}`

Retrieves a specified issue link.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue
`issue_link_id` Path, required	String or integer	ID of the issue relationship

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssueLink
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete an issue link

`DELETE /api/v4/projects/{id}/issues/{issue_iid}/links/{issue_link_id}`

Deletes a specified issue link, removing the two-way relationship.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue
`issue_link_id` Path, required	String or integer	ID of the issue relationship

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssueLink
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all participants in an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/participants`

Lists all users that are participants in a specified issue. If the project is private or the issue is confidential, you need to provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserBasic
`400`	Bad Request	—
`404`	Not Found	—

List all merge requests related to an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/related_merge_requests`

Lists all merge requests that are related to a specified issue. If the project is private or the issue is confidential, you need to provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`404`	Not Found	—

Reset spent time for an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/reset_spent_time`

Resets the total spent time for a specified issue to `0` seconds.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of the issue

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Reset the estimated time for an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/reset_time_estimate`

Resets the estimated time for a specified issue to `0` seconds.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of the issue

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Set the estimated time for an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/time_estimate`

Sets an estimated time of work for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`duration` Required	String	The duration in human format Example: `3h30m`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve time tracking stats for an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/time_stats`

Retrieves time tracking stats for a specified issue, including time estimate and time spent in seconds and human-readable format (for example, `1h 30m`).

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve user agent details for an issue

`GET /api/v4/projects/{id}/issues/{issue_iid}/user_agent_detail`

Retrieves user agent details for an issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserAgentDetail
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesIssue

Property	Type	Description
`_links`	Object	—
`_links.award_emoji`	String	Example: `http://example.com/api/v4/projects/1/issues/2/award_emoji`
`_links.closed_as_duplicate_of`	String	Example: `http://example.com/api/v4/projects/1/issues/75`
`_links.notes`	String	Example: `http://example.com/api/v4/projects/1/issues/2/notes`
`_links.project`	String	Example: `http://example.com/api/v4/projects/1`
`_links.self`	String	Example: `http://example.com/api/v4/projects/1/issues/2`
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`

Property	Type	Description
`has_tasks`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`moved_to_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `4`
`references`	APIEntitiesIssuableReferences	—
`service_desk_reply_to`	String	Example: `user@example.com`
`severity`	String	The severity of the issue Allowed values: `UNKNOWN`, `LOW`, `MEDIUM`, `HIGH`, `CRITICAL`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`subscribed`	Boolean	Example: `false`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`task_status`	String	Example: `2 of 4 tasks completed`
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`

Property	Type	Description
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesIssueBasic

Property	Type	Description
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`project_id`	Integer (int64)	Example: `4`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—

Property	Type	Description
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesIssueLink

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`link_type`	String	Example: `relates_to`
`source_issue`	APIEntitiesIssueBasic	—
`target_issue`	APIEntitiesIssueBasic	—

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesRelatedIssue

Property	Type	Description
`_links`	Object	—
`_links.award_emoji`	String	Example: `http://example.com/api/v4/projects/1/issues/2/award_emoji`
`_links.closed_as_duplicate_of`	String	Example: `http://example.com/api/v4/projects/1/issues/75`
`_links.notes`	String	Example: `http://example.com/api/v4/projects/1/issues/2/notes`
`_links.project`	String	Example: `http://example.com/api/v4/projects/1`
`_links.self`	String	Example: `http://example.com/api/v4/projects/1/issues/2`
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`

Property	Type	Description
`has_tasks`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`issue_link_id`	Integer (int64)	Example: `1`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`link_created_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`link_type`	String	Example: `relates_to`
`link_updated_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`moved_to_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `4`
`references`	APIEntitiesIssuableReferences	—
`service_desk_reply_to`	String	Example: `user@example.com`
`severity`	String	The severity of the issue Allowed values: `UNKNOWN`, `LOW`, `MEDIUM`, `HIGH`, `CRITICAL`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`subscribed`	Boolean	Example: `false`
`task_completion_status`		—

Property	Type	Description
	APIEntitiesTaskCompletionStatus	
`task_status`	String	Example: `2 of 4 tasks completed`
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserAgentDetail

Property	Type	Description
`akismet_submitted`	Boolean	Example: `false`
`ip_address`	String	Example: `127.0.0.1`
`user_agent`	String	Example: `AppleWebKit/537.36`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Jira Connect subscriptions

Use this API to subscribe a namespace to an installation of the GitLab for Jira Cloud app.

Subscribe a namespace to a JiraConnectInstallation

`POST /api/v4/integrations/jira_connect/subscriptions`

Subscribes the namespace to the JiraConnectInstallation

Request body (`application/json`)

Property	Type	Description
`jwt` Required	String	JWT token for authorization with the Jira Connect installation
`namespace_path` Required	String	Path for the namespace that should be subscribed

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBasicSuccess
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicSuccess

Property	Type	Description
`success`	Object	—

Jira forge installation

Use this API to configure a GitLab for Jira (Forge) installation, by setting the GitLab instance URL it points to and registering the system token it uses to synchronize development information. For more information, see the Jira integrations.

Update the GitLab for Jira (Forge) installation instance URL

`PUT /api/v4/integrations/jira_forge/installation`

Sets the GitLab instance the installation points at. Omit instance_url (or send null) for GitLab.com. Requires a Jira site or organization admin.

Request body (`application/json`)

Property	Type	Description
`instance_url`	String	Base URL of the self-managed GitLab instance; null for Git Lab.com Maximum length: `1024`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicSuccess
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`422`	Unprocessable entity	—

Destroy the GitLab for Jira (Forge) installation on app uninstall

`DELETE /api/v4/integrations/jira_forge/installation`

Called by the Forge app preUninstall hook. A pure-native install fires no Connect uninstalled event, so this is what removes the installation, its subscriptions, and deactivates the linked jira_cloud_app integrations. FIT authentication carries the same trust as the signed Connect lifecycle event, so no Jira admin check is needed. Succeeds with no change when the installation is already gone.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicSuccess
`401`	Unauthorized	—
`422`	Unprocessable entity	—

Register the GitLab for Jira (Forge) system token for direct dev-info sync

`POST /api/v4/integrations/jira_forge/installation/forge_token`

Stores the Forge app system OAuth token (X-Forge-Oauth-System header) and the Jira apiBaseUrl (from the FIT), so GitLab pushes dev-info directly to Jira. See Atlassian::Forge::SystemTokenClient.

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBasicSuccess
`400`	Bad Request	—
`401`	Unauthorized	—
`422`	Unprocessable entity	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicSuccess

Property	Type	Description
`success`	Object	—

Jira forge subscriptions

Use this API to list, create, and delete the namespace subscriptions of a GitLab for Jira (Forge) installation. For more information, see the Jira integrations.

List GitLab for Jira (Forge) namespace subscriptions

`GET /api/v4/integrations/jira_forge/subscriptions`

Lists the GitLab namespaces subscribed to the Forge installation.

Responses

Code	Description	Schema
`200`	OK	JiraConnectSubscriptionEntity
`401`	Unauthorized	—

Create a GitLab for Jira (Forge) namespace subscription

`POST /api/v4/integrations/jira_forge/subscriptions`

Subscribes a GitLab namespace to the Forge installation so its development data syncs to Jira. Authenticated by the Forge Invocation Token (app-context call); the GitLab user is bound via the signed delegation token from the user_delegation endpoint, and the Jira site-admin check runs against Jira with the app system token riding the call.

Request body (`application/json`)

Property	Type	Description
`namespace_path` Required	String	Path of the namespace to subscribe Maximum length: `255`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBasicSuccess
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a GitLab for Jira (Forge) namespace subscription

`DELETE /api/v4/integrations/jira_forge/subscriptions/{id}`

Unsubscribes a GitLab namespace from the Forge installation.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the subscription

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicSuccess
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicSuccess

Property	Type	Description
`success`	Object	—

JiraConnectGroupEntity

Property	Type	Description
`avatar_url`	String	—
`description`	String	—
`full_name`	String	—
`name`	String	—

JiraConnectSubscriptionEntity

Property	Type	Description
`created_at`	String	—
`group`	JiraConnectGroupEntity	—
`unlink_path`	String	—

Jira forge user delegation

Operations related to jira forge user delegation.

Create a GitLab for Jira (Forge) user delegation token

`POST /api/v4/integrations/jira_forge/user_delegation`

Mints a short-lived GitLab-signed token identifying the signed-in GitLab user, scoped to the Jira account and site it is minted for. Accepts only the OAuth token issued to the Forge app at sign-in. The Forge app stores it at config-page sign-in and presents it on the FIT-authenticated subscribe call, which cannot carry the user OAuth token (Forge attaches the FIT only to app-context calls); the scope claims must match that call's FIT.

Request body (`application/json`)

Property	Type	Description
`account_id` Required	String	Jira account id the delegation is scoped to Maximum length: `255`
`cloud_id` Required	String	Jira site (cloud) id the delegation is scoped to Maximum length: `255`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBasicSuccess
`400`	Bad Request	—
`401`	Unauthorized	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicSuccess

Property	Type	Description
`success`	Object	—

Job artifacts

Use this API to download, keep, and delete job artifacts.

Delete all job artifacts in a project

`DELETE /api/v4/projects/{id}/artifacts`

Deletes job artifacts from all jobs in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`409`	Conflict	—

Retrieve job artifacts

`GET /api/v4/projects/{id}/jobs/artifacts/{ref_name}/download`

Retrieves the artifacts archive for the latest successful job on a specified branch or tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`ref_name` Path, required	String	Branch or tag name in the repository. `HEAD` or SHA references are not supported. For merge request pipelines, use `refs/merge-requests/:iid/head` instead of the branch name
`job` Query, required	String	Name of the job
`job_token` Query	String	CI/CD job token for multi-project pipelines. Premium and Ultimate only
`search_recent_successful_pipelines` Query	Boolean	If `true`, searches across recent successful pipelines instead of just the latest one. Introduced in GitLab 18.7 with a feature flag named `ci_search_recent_successful_pipelines`. Disabled by default Default: `false`
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy`, `direct`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Download a specific file from artifacts archive from a ref

`GET /api/v4/projects/{id}/jobs/artifacts/{ref_name}/raw/{artifact_path}`

This feature was introduced in GitLab 11.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`ref_name` Path, required	String	Branch or tag name in the repository. `HEAD` or SHA references are not supported. For merge request pipelines, use `refs/merge-requests/:iid/head` instead of the branch name
`job` Query, required	String	Name of the job
`artifact_path` Path, required	String	Path to a file inside the artifacts archive
`job_token` Query	String	CI/CD job token for multi-project pipelines. Premium and Ultimate only
`search_recent_successful_pipelines` Query	Boolean	If `true`, searches across recent successful pipelines instead of just the latest one. Introduced in GitLab 18.7 with a feature flag named `ci_search_recent_successful_pipelines`. Disabled by default Default: `false`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Download an artifact from a job

`GET /api/v4/projects/{id}/jobs/{job_id}/artifacts`

This feature was introduced in GitLab 8.5. The `file_type` attribute was added in GitLab 19.4.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job
`file_type` Query	String	Type of artifact to download. Set a specific report type to download that report file directly instead of the full artifacts archive. `performance` and `browser_performance` share a report group. If the job uploaded both, the requested type is returned, and if it uploaded only one, that artifact is returned whichever type was requested. Introduced in Git Lab 19.4 Allowed values: `accessibility`, `api_fuzzing`, `archive`, `cobertura`, `jacoco`, `codequality`, `container_scanning`, `dast`, `dependency_scanning`, `dotenv`, `junit`, `license_scanning`, `lsif`, `metrics`, `performance`, `browser_performance`, `load_performance`, `sast`, `secret_detection`, `requirements`, `requirements_v2`, `cluster_image_scanning`, `cyclonedx`, `sarif` Default: `archive`
`job_token` Query	String	CI/CD job token for multi-project pipelines. Premium and Ultimate only

Name	Type	Description
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy` , `direct`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete job artifacts

`DELETE /api/v4/projects/{id}/jobs/{job_id}/artifacts`

Deletes job artifacts from a specified job in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`409`	Conflict	—

Retain job artifacts

`POST /api/v4/projects/{id}/jobs/{job_id}/artifacts/keep`

Retains job artifacts. Prevents artifacts for a job from being automatically deleted when they reach their expiration date.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all files in an artifacts archive

`GET /api/v4/projects/{id}/jobs/{job_id}/artifacts/tree`

Lists all files in a specified artifacts archive without extracting them.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job Example: `42`
`path` Query	String	Path to browse in the artifacts archive. Defaults to the root directory Default: `` Example: `coverage/reports`
`recursive` Query	Boolean	If `true`, returns all entries recursively Default: `false` Example: `false`
`job_token` Query	String	CI/CD job token for multi-project pipelines. Premium and Ultimate only
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiJobArtifactEntry
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Download a specific file from artifacts archive

`GET /api/v4/projects/{id}/jobs/{job_id}/artifacts/{artifact_path}`

This feature was introduced in GitLab 10.0

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`job_id` Path, required	Integer	ID of the job
`artifact_path` Path, required	String	Path to a file inside the artifacts archive
`job_token` Query	String	CI/CD job token for multi-project pipelines. Premium and Ultimate only

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactEntry

Property	Type	Description
`mode`	String	Example: `100644`
`name`	String	Example: `index.html`
`path`	String	Example: `coverage/index.html`
`size`	Integer	Example: `12345`
`type`	String	Allowed values: `file`, `directory` Example: `file`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Jobs

Use this API to run CI/CD jobs on a runner: request the next job, update its state, append to its log, and transfer its artifacts.

GitLab Runner uses these endpoints itself.

Request a job

`POST /api/v4/jobs/request`

Requests a job for a runner to execute.

Request body (`application/json`)

Property	Type	Description
`info`	Object	Runner’s metadata
`info.architecture`	String	Runner’s architecture
`info.config`	Object	Runner’s config
`info.config.gpus`	String	GPUs enabled
`info.executor`	String	Runner’s executor
`info.features`	Object	Runner’s features
`info.labels`	Object	Runner’s labels
`info.name`	String	Runner’s name
`info.platform`	String	Runner’s platform
`info.revision`	String	Runner’s revision
`info.version`	String	Runner’s version
`last_update`	String	`last_update` token of the runner queue
`session`	Object	Session data of the runner
`session.authorization`	String	Authorization of the session
`session.certificate`	String	Certificate of the session
`session.url`	String	URL of the session
`system_id`	String	Runner’s system identifier
`token` Required	String	Runner’s authentication token

Responses

Code	Description	Schema
`201`	Job was scheduled	APIEntitiesCiJobRequestResponse
`204`	No job for Runner	—
`400`	Bad Request	—
`403`	Forbidden	—
`409`	Conflict	—
`422`	Runner is orphaned	—
`429`	Too Many Requests	—

Update a job

`PUT /api/v4/jobs/{id}`

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job

Request body (`application/json`)

Property	Type	Description
`checksum`	String	CRC32 checksum of the job trace
`exit_code`	Integer	Exit code of the job
`failure_reason`	String	Failure reason of the job
`output`	Object	Build log state
`output.bytesize`	Integer	Size of the job's trace output, in bytes
`output.checksum`	String	CRC32 checksum of the job trace
`runtime_environment_key`	String	Runtime environment key emitted by the runner on job suspension Maximum length: `512`
`state`	String	Status of the job: `running`, `success`, or `failed`
`token` Required	String	Authentication token of the job

Responses

Code	Description	Schema
`200`	Job was updated	—
`202`	Update accepted	—
`400`	Unknown parameters	—
`403`	Forbidden	—
`404`	Not Found	—
`409`	Conflict	—
`429`	Too Many Requests	—

Download job artifacts

`GET /api/v4/jobs/{id}/artifacts`

Downloads artifacts for a specified job.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job
`token` Query	String	Authentication token of the job
`direct_download` Query	Boolean	If `true`, performs a direct download of artifacts from remote storage instead of proxying them Default: `false`
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy`, `direct`

Responses

Code	Description	Schema
`200`	Download allowed	String (binary) (`application/octet-stream`)
`302`	Found	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Artifact not found	—
`429`	Too Many Requests	—

Upload job artifacts

`POST /api/v4/jobs/{id}/artifacts`

Uploads artifacts for a specified job.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job

Request body (`multipart/form-data`)

Property	Type	Description
`accessibility`	String	Specify accessibility level of artifact private/public
`artifact_format`	String	The format of artifact Allowed values: `raw`, `zip`, `gzip` Default: `zip`
`artifact_type`	String	Type of artifact Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Default: `archive`
`expire_in`	String	Specify when artifact should expire
`file` Required	String (binary)	The artifact file to store (generated by Multipart middleware)
`metadata`	String (binary)	The artifact metadata to store (generated by Multipart middleware)

Property	Type	Description
`token`	String	Authentication token of the job

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad request	—
`403`	Forbidden	—
`404`	Not Found	—
`405`	Artifacts support not enabled	—
`413`	File too large	—
`429`	Too Many Requests	—

Authorize artifacts upload

`POST /api/v4/jobs/{id}/artifacts/authorize`

Authorizes uploading artifacts for a specified job.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job

Request body (`application/json`)

Property	Type	Description
`artifact_type`	String	Type of artifact Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Default: `archive`
`filesize`	Integer	Size of the artifact file
`token`	String	Authentication token of the job

Responses

Code	Description	Schema
`200`	Upload allowed	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—
`405`	Artifacts support not enabled	—
`413`	File too large	—
`429`	Too Many Requests	—

Get the runtime environment key for a job

`GET /api/v4/jobs/{id}/runtime_environment_key`

Retrieves the runtime environment key linked to a job, if the job is resuming a suspended environment.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job
`token` Query	String	Authentication token of the job

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRuntimeEnvironmentKey
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too Many Requests	—

Append a patch to the job trace

`PATCH /api/v4/jobs/{id}/trace`

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the job

Request body (`application/json`)

Property	Type	Description
`debug_trace`	Boolean	If `true` , enables debug tracing for the job
`token`	String	Authentication token of the job

Responses

Code	Description	Schema
`202`	Trace was patched	—
`400`	Missing Content-Range header	—
`403`	Forbidden	—
`404`	Not Found	—
`416`	Range not satisfiable	—
`429`	Too Many Requests	—

List all jobs processed by a runner

`GET /api/v4/runners/{id}/jobs`

Lists all jobs that are being processed or were processed by a specified runner. The list of jobs is limited to projects where the user has the Reporter, Developer, Maintainer, or Owner role.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner
`system_id` Query	String	System ID associated with the runner manager
`status` Query	String	Status of the job Allowed values: `created`, `waiting_for_resource`, `preparing`, `waiting_for_callback`, `pending`, `running`, `success`, `failed`, `canceling`, `canceled`, `skipped`, `manual`, `scheduled`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`
`sort` Query	String	Sort results in ascending or descending order. Specify `order_by` as well, including for `id` Allowed values: `asc`, `desc` Default: `desc`
`cursor` Query	String	Cursor for obtaining the next set of records
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJobBasicWithProject
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiJobBasicWithProject

Property	Type	Description
`allow_failure`	Boolean	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	APIEntitiesProjectIdentity	—
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`user`	APIEntitiesUser	—
`web_url`	String	

Property	Type	Description
		Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobRequestArtifacts

Property	Type	Description
`artifact_format`	String	—
`artifact_type`	String	—
`exclude`	String	—
`expire_in`	String	—
`name`	String	—
`paths`	String	—
`untracked`	String	—
`when`	String	—

APIEntitiesCiJobRequestCache

Property	Type	Description
`fallback_keys`	String	—
`key`	String	—
`paths`	String	—
`policy`	String	—
`untracked`	String	—
`when`	String	—

APIEntitiesCiJobRequestCredentials

Property	Type	Description
`password`	String	—
`type`	String	—
`url`	String	—
`username`	String	—

APIEntitiesCiJobRequestGitInfo

Property	Type	Description
`before_sha`	String	—
`depth`	String	—
`protected`	String	—
`ref`	String	—
`ref_type`	String	—
`refspecs`	String	—
`repo_object_format`	String	—
`repo_url`	String	—
`sha`	String	—

APIEntitiesCiJobRequestHook

Property	Type	Description
`name`	String	—
`script`	String	—

APIEntitiesCiJobRequestImage

Property	Type	Description
`entrypoint`	String	—
`executor_opts`	String	—
`name`	String	—
`ports`	APIEntitiesCiJobRequestPort	—
`pull_policy`	String	—

APIEntitiesCiJobRequestJobInfo

Property	Type	Description
`id`	String	—
`instance_id`	String	—
`instance_uuid`	String	—
`name`	String	—
`namespace_id`	String	—
`organization_id`	String	—
`pipeline_id`	String	—
`project_full_path`	String	—
`project_id`	String	—
`project_jobs_running_on_instance_runners_count`	String	—
`project_name`	String	—
`queue_depth`	String	—
`queue_size`	String	—
`root_namespace_id`	String	—
`scoped_user_id`	String	—
`stage`	String	—
`time_in_queue_seconds`	String	—
`user_id`	String	—

APIEntitiesCiJobRequestPort

Property	Type	Description
`name`	String	—
`number`	String	—
`protocol`	String	—

APIEntitiesCiJobRequestResponse

Property	Type	Description
`allow_git_fetch`	String	—
`artifacts`	APIEntitiesCiJobRequestArtifacts	—
`cache`	APIEntitiesCiJobRequestCache	—
`credentials`	APIEntitiesCiJobRequestCredentials	—
`dependencies`	String	—
`features`	String	—
`git_info`	APIEntitiesCiJobRequestGitInfo	—
`hooks`	APIEntitiesCiJobRequestHook	—
`id`	String	—
`image`	APIEntitiesCiJobRequestImage	—
`inputs`	String	—
`job_info`	APIEntitiesCiJobRequestJobInfo	—
`policy_options`	String	—
`run`	String	—
`runner_info`	APIEntitiesCiJobRequestRunnerInfo	—
`secrets`	String	—
`services`	APIEntitiesCiJobRequestService	—
`steps`	APIEntitiesCiJobRequestStep	—
`suspend_options`	String	—
`token`	String	—
`variables`	String	—

APIEntitiesCiJobRequestRunnerInfo

Property	Type	Description
`runner_session_url`	String	—
`timeout`	String	—
`uuid`	String	—

APIEntitiesCiJobRequestService

Property	Type	Description
`alias`	String	—
`command`	String	—
`entrypoint`	String	—
`executor_opts`	String	—
`name`	String	—
`ports`	APIEntitiesCiJobRequestPort	—
`pull_policy`	String	—
`variables`	String	—

APIEntitiesCiJobRequestStep

Property	Type	Description
`allow_failure`	String	—
`name`	String	—
`script`	String	—
`timeout`	String	—
`when`	String	—

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiRuntimeEnvironmentKey

Property	Type	Description
`runtime_environment_key`	String	Example: `42/s_machineid/data`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

Keys

Use this API to retrieve information about SSH keys, manage a user's SSH and GPG keys, and manage SSH certificates for top-level groups and the instance.

Queries about deploy key fingerprints also retrieve information about the projects using that key. If you use a SHA256 fingerprint in an API call, URL-encode the fingerprint.

Managing group SSH certificates requires the Owner role for a top-level group.

Managing instance SSH certificates requires administrator access.

Retrieve user by SSH key fingerprint

`GET /api/v4/keys`

Retrieves user by SSH key fingerprint. Administrators only.

Parameters

Name	Type	Description
`fingerprint` Query, required	String	Fingerprint of an SSH key Example: `ba:81:59:68:d7:6c:cd:02:02:bf:6a:9b:55:4e:af:d1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserWithAdmin
`400`	Bad Request	—

Retrieve user by SSH key ID

`GET /api/v4/keys/{id}`

Retrieves user by SSH key ID. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID of the SSH key Example: `2`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSSHKeyWithUser
`400`	Bad Request	—
`404`	Not Found	—

List all GPG keys

`GET /api/v4/user/gpg_keys`

Lists all GPG keys for the currently authenticated user.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGpgKey
`400`	Bad Request	—

Add a GPG key

`POST /api/v4/user/gpg_keys`

Adds a GPG key for the currently authenticated user.

Request body (`application/json`)

Property	Type	Description
`key` Required	String	Public key value of the GPG key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGpgKey
`400`	Bad Request	—

Retrieve a GPG key

`GET /api/v4/user/gpg_keys/{key_id}`

Retrieves a GPG key for the currently authenticated user.

Parameters

Name	Type	Description
`key_id` Path, required	Integer	ID of the GPG key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGpgKey
`400`	Bad Request	—
`404`	Not Found	—

Delete a GPG key

`DELETE /api/v4/user/gpg_keys/{key_id}`

Deletes a GPG key from your user account.

Parameters

Name	Type	Description
`key_id` Path, required	Integer	ID of the GPG key

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all SSH keys

`GET /api/v4/user/keys`

Lists all SSH keys for the currently authenticated user.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSSHKey
`400`	Bad Request	—

Add an SSH key

`POST /api/v4/user/keys`

Adds an SSH key for the currently authenticated user.

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date-time)	Date when the SSH key expires
`key` Required	String	Public key value of the SSH key
`title` Required	String	Title of the SSH key
`usage_type`	String	Usage scope for the key Allowed values: `auth_and_signing` , `auth` , `signing` Default: `auth_and_signing`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesSSHKey
`400`	Bad Request	—

Retrieve an SSH key

`GET /api/v4/user/keys/{key_id}`

Retrieves a specified SSH key for the currently authenticated user.

Parameters

Name	Type	Description
`key_id` Path, required	Integer	ID of the SSH key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

Delete an SSH key

`DELETE /api/v4/user/keys/{key_id}`

Deletes a specified SSH key from the currently authenticated user.

Parameters

Name	Type	Description
`key_id` Path, required	Integer	ID of the SSH key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

List all GPG keys for a user

`GET /api/v4/users/{id}/gpg_keys`

Lists all GPG keys for a specified user account. This endpoint does not require authentication.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesGpgKey
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a GPG key for a user

`GET /api/v4/users/{id}/gpg_keys/{key_id}`

Retrieves a GPG key for a specified user account. This endpoint does not require authentication.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`key_id` Path, required	Integer	ID of the GPG key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGpgKey
`400`	Bad Request	—
`404`	Not Found	—

Delete a GPG key for a user

`DELETE /api/v4/users/{id}/gpg_keys/{key_id}`

Deletes a GPG key from a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`key_id` Path, required	Integer	ID of the GPG key

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an SSH key for a user

`GET /api/v4/users/{id}/keys/{key_id}`

Retrieves an SSH key for a specified user account. This endpoint does not require authentication.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`key_id` Path, required	Integer	ID of the SSH key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

Delete an SSH key for a user

`DELETE /api/v4/users/{id}/keys/{key_id}`

Deletes an SSH key from a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`key_id` Path, required	Integer	ID of the SSH key

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

List all SSH keys for a user

`GET /api/v4/users/{user_id}/keys`

Lists all SSH keys for a specified user account.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

Add an SSH key for a user

`POST /api/v4/users/{user_id}/keys`

Adds an SSH key for a specified user account. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date-time)	Date when the SSH key expires
`key` Required	String	Public key value of the SSH key
`title` Required	String	Title of the SSH key
`usage_type`	String	Usage scope for the key Allowed values: `auth_and_signing`, `auth`, `signing` Default: `auth_and_signing`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesSSHKey
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesGpgKey

Property	Type	Description
`created_at`	String (date-time)	Example: `2017-09-05T09:17:46.264Z`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `-----BEGIN PGP PUBLIC KEY BLOCK-----`

APIEntitiesIdentity

Property	Type	Description
`extern_uid`	String	—
`provider`	String	—

APIEntitiesSSHKey

Property	Type	Description
`created_at`	String (date-time)	Example: `2015-09-03T07:24:44.627Z`
`expires_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAiPWx6WM4lhHNedGfBpPJNPpZ7yKu+dnn1SJejgt1256k6Yjz\ GGphH2TUxwKzxcKDKKezwkpfnxPkSMkuEspGRt/aZZ9wa++Oi7Qkr8prgHc4soW6NUlfDzpvZK2H5E7eQaSeP3SAwGmQKUFHCdd\ NaP0L+hM7zhFNzjFvpaMgJw0=`
`last_used_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`title`	String	Example: `Sample key 25`
`usage_type`	String	Example: `auth`

APIEntitiesSSHKeyWithUser

Property	Type	Description
`created_at`	String (date-time)	Example: `2015-09-03T07:24:44.627Z`
`expires_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `ssh-rsa AAAAB3NzaC1yc2EAAAABJQAAAIEAiPWx6WM4lhHNedGfBpPJNPpZ7yKu+dnn1SJejgt1256k6Yjz\ GGphH2TUxwKzxcKDKKezwkpfnxPkSMkuEspGRt/aZZ9wa++Oi7Qkr8prgHc4soW6NUlfDzpvZK2H5E7eQaSeP3SAwGmQKUFHCdd\ NaP0L+hM7zhFNzjFvpaMgJw0=`
`last_used_at`	String (date-time)	Example: `2020-09-03T07:24:44.627Z`
`title`	String	Example: `Sample key 25`
`usage_type`	String	Example: `auth`
`user`	APIEntitiesUserPublic	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserPublic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`can_create_group`	Boolean	Example: `true`
`can_create_project`	Boolean	Example: `true`
`color_scheme_id`	Integer	Example: `1`
`commit_email`	String	—
`confirmed_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`created_at`	String	—
`current_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`email`	String	Example: `john@example.com`
`external`	Boolean	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`identities`	APIEntitiesIdentity	—
`is_followed`	String	—

Property	Type	Description
`job_title`	String	—
`last_activity_on`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`last_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`preferred_language`	String	Example: `en`
`private_profile`	Boolean	Example: `false`
`projects_limit`	Integer	Example: `10`
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`theme_id`	Integer	Example: `2`
`twitter`	String	—
`two_factor_enabled`	Boolean	Example: `true`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserWithAdmin

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`can_create_group`	Boolean	Example: `true`
`can_create_project`	Boolean	Example: `true`
`color_scheme_id`	Integer	Example: `1`
`commit_email`	String	—
`confirmed_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`created_at`	String	—
`created_by`	APIEntitiesUserBasic	—
`current_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`email`	String	Example: `john@example.com`
`external`	Boolean	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`identities`	APIEntitiesIdentity	—

Property	Type	Description
`is_admin`	String	—
`is_followed`	String	—
`job_title`	String	—
`last_activity_on`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`last_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`namespace_id`	String	—
`note`	String	—
`organization`	String	—
`preferred_language`	String	Example: `en`
`private_profile`	Boolean	Example: `false`
`projects_limit`	Integer	Example: `10`
`pronouns`	String	—
`provisioned_by_project_id`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`theme_id`	Integer	Example: `2`
`twitter`	String	—
`two_factor_enabled`	Boolean	Example: `true`
`username`	String	Example: `admin`

Property	Type	Description
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

Labels

Use this API to manage labels for both projects and groups. For the difference between the two, see types of labels.

List all group labels

`GET /api/v4/groups/{id}/labels`

Lists all group labels for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`with_counts` Query	Boolean	Include issue and merge request counts Default: `false`
`include_ancestor_groups` Query	Boolean	Include ancestor groups Default: `true`
`include_descendant_groups` Query	Boolean	Include descendant groups. This feature was added in Git Lab 13.6 Default: `false`
`only_group_labels` Query	Boolean	Toggle to include only group labels or also project labels. This feature was added in Git Lab 13.6 Default: `true`
`search` Query	String	Keyword to filter labels by. This feature was added in GitLab 13.6
`archived` Query	Boolean	Filter by archived status. This feature was added in GitLab 18.10
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Create a group label

`POST /api/v4/groups/{id}/labels`

Creates a group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived
`color` Required	String	Color of the label in 6-digit hex notation with a leading `#` sign, for example `#FFAABB`, or one of the CSS color names
`description`	String	Description of the label
`name` Required	String	The name of the label to be created

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Update a group label

`PUT /api/v4/groups/{id}/labels`

Updates an existing group label. At least one parameter is required to update the group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived. At least one of `new_name`, `color`, `description`, `archived` must be provided
`color`	String	New color of the label, given in 6-digit hex notation with a leading `#` sign (for example, `#FFAABB`), or one of the CSS color names. At least one of `new_name`, `color`, `description`, `archived` must be provided
`description`	String	New description of label. At least one of `new_name`, `color`, `description`, `archived` must be provided
`label_id`	Integer	The ID of the label to be updated. Exactly one of `label_id`, `name` must be provided
`name`	String	The name of the label to be updated. Exactly one of `label_id`, `name` must be provided
`new_name`	String	The new name of the label. At least one of `new_name`, `color`, `description`, `archived` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Delete a group label

`DELETE /api/v4/groups/{id}/labels`

Deletes a specified group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`name` Query, required	String	The name of the label to be deleted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a group label

`GET /api/v4/groups/{id}/labels/{name}`

Retrieves a specified group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`name` Path, required	String or integer	The ID or name of a label
`include_ancestor_groups` Query	Boolean	Include ancestor groups Default: `true`
`include_descendant_groups` Query	Boolean	Include descendant groups. This feature was added in Git Lab 13.6 Default: `false`
`only_group_labels` Query	Boolean	Toggle to include only group labels or also project labels. This feature was added in Git Lab 13.6 Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Update a group label

`PUT /api/v4/groups/{id}/labels/{name}`

Updates a specified group label. At least one parameter is required to update the group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`name` Path, required	String	The name or id of the label to be updated

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived. At least one of `new_name`, `color`, `description`, `archived` must be provided
`color`	String	New color of the label, given in 6-digit hex notation with a leading `#` sign (for example, `#FFAABB`), or one of the CSS color names. At least one of `new_name`, `color`, `description`, `archived` must be provided
`description`	String	New description of label. At least one of `new_name`, `color`, `description`, `archived` must be provided
`new_name`	String	The new name of the label. At least one of `new_name`, `color`, `description`, `archived` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Delete a group label

`DELETE /api/v4/groups/{id}/labels/{name}`

Deletes a specified group label.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group
`name` Path, required	String	The name or id of the label to be deleted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

List all project labels

`GET /api/v4/projects/{id}/labels`

Lists all labels for a specified project. By default, this request returns 20 results at a time because the API results are paginated.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`with_counts` Query	Boolean	Include issue and merge request counts Default: `false`
`include_ancestor_groups` Query	Boolean	Include ancestor groups Default: `true`
`search` Query	String	Keyword to filter labels by. This feature was added in GitLab 13.6
`archived` Query	Boolean	Filter by archived status. This feature was added in GitLab 18.10
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Create a project label

`POST /api/v4/projects/{id}/labels`

Creates a label for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived
`color` Required	String	Color of the label in 6-digit hex notation with a leading `#` sign, for example `#FFAABB`, or one of the CSS color names
`description`	String	Description of the label
`name` Required	String	The name of the label to be created
`priority`	Integer	The priority of the label

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Update an existing label (deprecated)

`PUT /api/v4/projects/{id}/labels`

At least one optional parameter is required.

Deprecated in GitLab 12.4. Use PUT /projects/:id/labels/:name instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`color`	String	New color of the label, given in 6-digit hex notation with a leading `#` sign (for example, `#FFAABB`), or one of the CSS color names. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`description`	String	New description of label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`label_id`	Integer	The ID of the label to be updated. Exactly one of `label_id`, `name` must be provided
`name`	String	The name of the label to be updated. Exactly one of `label_id`, `name` must be provided
`new_name`	String	The new name of the label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`priority`	Integer	The priority of the label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Delete an existing label (deprecated)

`DELETE /api/v4/projects/{id}/labels`

Deprecated in GitLab 12.4. Use DELETE /projects/:id/labels/:name instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`label_id` Query	Integer	The ID of the label to be deleted. Exactly one of `label_id`, `name` must be provided
`name` Query	String	The name of the label to be deleted. Exactly one of `label_id`, `name` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Promote a label to a group label (deprecated)

`PUT /api/v4/projects/{id}/labels/promote`

Added in GitLab 12.3 and deprecated in GitLab 12.4. Use PUT /projects/:id/labels/:name/promote instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`name` Required	String	The name of the label to be promoted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a project label

`GET /api/v4/projects/{id}/labels/{name}`

Retrieves a specified label for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`name` Path, required	String or integer	The ID or name of a label
`include_ancestor_groups` Query	Boolean	Include ancestor groups Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Update a project label

`PUT /api/v4/projects/{id}/labels/{name}`

Updates a specified label for a project with a different name or color. At least one parameter is required to update the label.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`name` Path, required	String	The name or id of the label to be updated

Request body (`application/json`)

Property	Type	Description
`archived`	Boolean	Whether the label is archived. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`color`	String	New color of the label, given in 6-digit hex notation with a leading `#` sign (for example, `#FFAABB`), or one of the CSS color names. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`description`	String	New description of label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`new_name`	String	The new name of the label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided
`priority`	Integer	The priority of the label. At least one of `new_name`, `color`, `description`, `priority`, `archived` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Delete a project label

`DELETE /api/v4/projects/{id}/labels/{name}`

Deletes a specified label from a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`name` Path, required	String	The name or id of the label to be deleted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Promote a project label to a group label

`PUT /api/v4/projects/{id}/labels/{name}/promote`

Promotes a specified project label to a group label. The label keeps its ID.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`name` Path, required	String	The name or id of the label to be promoted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesGroupLabel

Property	Type	Description
`archived`	Boolean	Example: `false`
`closed_issues_count`	Integer	Example: `0`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `bug`
`open_issues_count`	Integer	Example: `1`
`open_merge_requests_count`	Integer	Example: `1`
`subscribed`	Boolean	Example: `false`
`text_color`	String	Example: `#FFFFFF`

APIEntitiesProjectLabel

Property	Type	Description
`archived`	Boolean	Example: `false`
`closed_issues_count`	Integer	Example: `0`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`is_project_label`	Boolean	—
`name`	String	Example: `bug`
`open_issues_count`	Integer	Example: `1`
`open_merge_requests_count`	Integer	Example: `1`
`priority`	Integer	Example: `10`
`subscribed`	Boolean	Example: `false`
`text_color`	String	Example: `#FFFFFF`

MCP

Use this API to connect Model Context Protocol (MCP) clients to the GitLab MCP server.

Get MCP response listener

`GET /api/v4/mcp`

Listens for Model Context Protocol responses

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—
`405`	Method not allowed	—

Create MCP request handler

`POST /api/v4/mcp`

Handles Model Context Protocol requests

Request body (`application/json`)

Property	Type	Description
`id`	Integer or string	ID of the JSON-RPC request returned in the response
`jsonrpc` Required	String	JSON-RPC protocol version identifier. Must be `2.0` Allowed values: `2.0` Minimum length: `1`
`method` Required	String	Name of the JSON-RPC method invoked on the MCP server
`params`	Object or array	Parameters for the JSON-RPC method specified in `method`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Method not found	—

ML Model Registry

Use this API to upload and download the files attached to a version of a machine learning model in the model registry. To create models and model versions, use an MLflow client.

Download an ml_model package file

`GET /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{file_name}`

This feature was introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`status` Query	String	Package status Allowed values: `default`, `hidden`
`model_version_id` Path, required	String	Model version id

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse upload model package file

`PUT /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{file_name}`

Introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`model_version_id` Path, required	String	Model version id

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize model package file

`PUT /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{file_name}/authorize`

Introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`model_version_id` Path, required	String	Model version id

Request body (`application/json`)

Property	Type	Description
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download an ml_model package file

`GET /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{path}/{file_name}`

This feature was introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`path` Path, required	String	File directory path
`status` Query	String	Package status Allowed values: `default`, `hidden`
`model_version_id` Path, required	String	Model version id

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse upload model package file

`PUT /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{path}/{file_name}`

Introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`path` Path, required	String	File directory path
`model_version_id` Path, required	String	Model version id

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize model package file

`PUT /api/v4/projects/{id}/packages/ml_models/{model_version_id}/files/{path}/{file_name}/authorize`

Introduced in GitLab 16.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	File name
`path` Path, required	String	File directory path
`model_version_id` Path, required	String	Model version id

Request body (`application/json`)

Property	Type	Description
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

MLOps

Use this API to track machine learning experiments, runs, and metrics, and to manage the model registry. These endpoints implement the MLflow REST API, so you can point an MLflow client at GitLab.

MLflow artifact API

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow-artifacts/artifacts`

MLflow artifacts mapping to GitLab artifacts

Parameters

Name	Type	Description
`path` Query	String	Path to the artifact, model version id, optionally followed by path. E.g. 15/MLmodel
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Get MLflow artifact file

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow-artifacts/artifacts/{model_version}/{file_path}`

Retrieves an MLflow artifact file

Parameters

Name	Type	Description
`model_version` Path, required	String	The model version
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`file_path` Path, required	—	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Create experiment

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/create`

https://www.mlflow.org/docs/2.19.0/rest-api.html#create-experiment

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`artifact_location`	String	Ignored. Accepted for MLflow API compatibility
`name` Required	String	Name of the experiment
`tags`	Array	Tags containing metadata about the experiment

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowNewExperiment
`400`	Bad Request	—
`404`	Not Found	—

Delete an experiment

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/delete`

https://mlflow.org/docs/2.19.0/rest-api.html#delete-experiment

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`experiment_id` Required	String	ID of the experiment

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Fetch experiment by experiment_id

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/get`

https://www.mlflow.org/docs/2.19.0/rest-api.html#get-experiment

Parameters

Name	Type	Description
`experiment_id` Query	String	ID of the experiment, relative to the project Default: ``
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowGetExperiment
`400`	Bad Request	—
`404`	Not Found	—

Fetch experiment by experiment_name

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/get-by-name`

https://www.mlflow.org/docs/2.19.0/rest-api.html#get-experiment-by-name

Parameters

Name	Type	Description
`experiment_name` Query	String	Name of the experiment Default: ``
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowGetExperiment
`400`	Bad Request	—
`404`	Not Found	—

List experiments

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/list`

https://www.mlflow.org/docs/2.19.0/rest-api.html#search-experiments

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowListExperiment
`404`	Not Found	—

Search experiments

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/search`

https://www.mlflow.org/docs/2.19.0/rest-api.html#search-experiments

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`filter`	String	Ignored. Accepted for MLflow API compatibility
`max_results`	Integer	Maximum number of experiments to fetch in a page. The maximum is 1000 Default: `200`
`order_by`	String	Column and direction to sort experiments by, for example `created_at DESC` or `name`. Supported columns are `created_at` and `name` Default: `created_at DESC`
`page_token`	String	Token for pagination

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowListExperiment
`400`	Bad Request	—
`404`	Not Found	—

Sets a tag for an experiment

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/experiments/set-experiment-tag`

https://www.mlflow.org/docs/2.19.0/rest-api.html#set-experiment-tag

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`experiment_id` Required	String	ID of the experiment
`key` Required	String	Name of the tag
`value` Required	String	Value of the tag

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Gets metric history for a run

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/metrics/get-history`

https://www.mlflow.org/docs/2.19.0/rest-api.html#get-metric-history

Parameters

Name	Type	Description
`run_id` Query, required	String	UUID of the run
`metric_key` Query, required	String	Name of the metric
`max_results` Query	Integer	Maximum number of metrics to return Default: `1000`
`page_token` Query	String	Token for pagination
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowGetMetricHistory
`400`	Bad Request	—
`404`	Not Found	—

Creates a Model Version

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/model-versions/create`

MLFlow Model Versions map to GitLab Model Versions. https://mlflow.org/docs/2.19.0/rest-api.html#create-modelversion

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the model version
`name`	String	Name of the registered model to create the version for. Required
`run_id`	String	Run ID of the candidate to be promoted to a model version
`tags`	Array	Additional metadata for a model version

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowModelVersion
`400`	Bad Request	—
`404`	Not Found	—

Fetch model version by name and version

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/model-versions/get`

https://mlflow.org/docs/2.19.0/rest-api.html#get-modelversion

Parameters

Name	Type	Description
`name` Query, required	String	Name of the registered model
`version` Query, required	String	Version of the model
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowModelVersion
`400`	Bad Request	—
`404`	Not Found	—

Fetch the download URI for the model version

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/model-versions/get-download-uri`

Returns version in MLflow format “mlflow-artifacts:<version>” https://mlflow.org/docs/2.19.0/rest-api.html#get-download-uri-for-modelversion-artifacts

Parameters

Name	Type	Description
`name` Query, required	String	Name of the registered model
`version` Query, required	Integer	ID of the model version
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowGetDownload
`400`	Bad Request	—
`404`	Not Found	—

Updates a Model Version

`PATCH /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/model-versions/update`

https://mlflow.org/docs/2.19.0/rest-api.html#update-modelversion

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the model version
`name`	String	Name of the registered model

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowModelVersion
`400`	Bad Request	—
`404`	Not Found	—

Gets a Model Version by alias

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/alias`

https://mlflow.org/docs/2.19.0/rest-api.html#get-model-version-by-alias

Parameters

Name	Type	Description
`name` Query	String	Name of the registered model
`alias` Query	String	Alias of the model version, for example `1.0.0`
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Creates a Registered Model

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/create`

MLFlow Registered Models map to GitLab Models. https://mlflow.org/docs/2.19.0/rest-api.html#create-registeredmodel

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the registered model
`name` Required	String	Name of the registered model
`tags`	Array	Additional metadata for the registered model

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowRegisteredModel
`400`	Bad Request	—
`404`	Not Found	—

Delete a Registered Model by Name

`DELETE /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/delete`

https://mlflow.org/docs/2.19.0/rest-api.html#delete-registeredmodel

Parameters

Name	Type	Description
`name` Query	String	Name of the registered model
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowRegisteredModel
`400`	Bad Request	—
`404`	Not Found	—

Fetch a Registered Model by Name

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/get`

https://www.mlflow.org/docs/2.19.0/rest-api.html#get-registeredmodel

Parameters

Name	Type	Description
`name` Query	String	Name of the registered model Default: ``
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowRegisteredModel
`400`	Bad Request	—
`404`	Not Found	—

Fetch the latest Model Version for the given Registered Model Name

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/get-latest-versions`

https://mlflow.org/docs/2.19.0/rest-api.html#get-latest-modelversions

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`name`	String	Name of the registered model

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowModelVersion
`400`	Bad Request	—
`404`	Not Found	—

Search Registered Models within a project

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/search`

https://mlflow.org/docs/2.19.0/rest-api.html#search-registeredmodels

Parameters

Name	Type	Description
`filter` Query	String	Filter for the model search, in the format `name='value'`. Only filtering by name is supported
`max_results` Query	Integer	Maximum number of models to fetch in a page. The maximum is 1000 Default: `200`
`order_by` Query	String	Column and direction to sort registered models by, for example `name`, `name DESC`, or `last_updated_timestamp DESC`. Supported columns are `name` and `last_updated_timestamp`. Sorting by model metadata is not supported Default: `name ASC`
`page_token` Query	String	Token for pagination
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowRegisteredModel
`400`	Bad Request	—
`404`	Not Found	—

Update a Registered Model by Name

`PATCH /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/registered-models/update`

https://mlflow.org/docs/2.19.0/rest-api.html#update-registeredmodel

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the registered model
`name`	String	Name of the registered model

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowRegisteredModel
`400`	Bad Request	—
`404`	Not Found	—

Creates a Run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/create`

MLFlow Runs map to GitLab Candidates. https://www.mlflow.org/docs/2.19.0/rest-api.html#create-run

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`experiment_id` Required	Integer	Id for the experiment, relative to the project
`run_name`	String	A name for this run
`start_time`	Integer	Unix timestamp in milliseconds of when the run started Default: `0`
`tags`	Array	Tags are stored, but not displayed
`user_id`	String	This will be ignored

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowRun
`400`	Bad Request	—
`404`	Not Found	—

Delete a run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/delete`

https://mlflow.org/docs/2.19.0/rest-api.html#delete-run

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`run_id` Required	String	UUID of the run

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Gets an MLFlow Run, which maps to GitLab Candidates

`GET /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/get`

https://www.mlflow.org/docs/1.28.0/rest-api.html#get-run

Parameters

Name	Type	Description
`run_id` Query, required	String	UUID of the candidate
`run_uuid` Query	String	This parameter is ignored
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMlMlflowRun
`400`	Bad Request	—
`404`	Not Found	—

Logs multiple parameters and metrics

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/log-batch`

https://www.mlflow.org/docs/2.19.0/rest-api.html#log-param

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`metrics`	Array of objects	Array that contains metric information Default: `[]`
`metrics[].key` Required	String	Name for the metric
`metrics[].step`	Integer	Step at which the metric was recorded
`metrics[].timestamp` Required	Integer	Unix timestamp in milliseconds when metric was recorded
`metrics[].value` Required	Number	Value of the metric
`params`	Array of objects	Array that contains parameter information Default: `[]`
`params[].key` Required	String	Name for the metric
`params[].value` Required	String	Value of the metric
`run_id` Required	String	UUID of the run

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Logs a metric to a run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/log-metric`

https://www.mlflow.org/docs/1.28.0/rest-api.html#log-metric

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`key` Required	String	Name for the metric
`run_id` Required	String	UUID of the run
`step`	Integer	Step at which the metric was recorded
`timestamp` Required	Integer	Unix timestamp in milliseconds when metric was recorded
`value` Required	Number	Value of the metric

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Logs a parameter to a run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/log-parameter`

https://www.mlflow.org/docs/2.19.0/rest-api.html#log-param

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`key` Required	String	Name for the parameter
`run_id` Required	String	UUID of the run
`value` Required	String	Value for the parameter

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Searches runs/candidates within a project

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/search`

https://www.mlflow.org/docs/2.19.0/rest-api.html#search-runsexperiment_ids supports only a single experiment ID.Introduced in GitLab 16.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`experiment_ids` Required	Array	IDs of the experiments to get searches from, relative to the project
`max_results`	Integer	Maximum number of runs/ candidates to fetch in a page. Default is 200, maximum in 1000 Default: `200`
`order_by`	String	Order criteria. Can be by a column of the run/candidate (created_at, name) or by a metric ifprefixed by `metrics.` Valid examples: `created_at`, `created_at DESC`, `metrics.my_metric DESC` Sorting by candidate parameter or metadata is not supported Default: `created_at DESC`
`page_token`	String	Token for pagination

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowRun
`400`	Bad Request	—
`404`	Not Found	—

Sets a tag for a run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/set-tag`

https://www.mlflow.org/docs/2.19.0/rest-api.html#set-tag

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`key` Required	String	Name for the tag
`run_id` Required	String	UUID of the run
`value` Required	String	Value for the tag

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Updates a Run

`POST /api/v4/projects/{id}/ml/mlflow/api/2.0/mlflow/runs/update`

MLFlow Runs map to GitLab Candidates. https://www.mlflow.org/docs/2.19.0/rest-api.html#update-run

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`end_time`	Integer	Ending time of the run
`run_id` Required	String	UUID of the candidate
`status`	String	Status of the run. Accepts: [“RUNNING”, “SCHEDULED”, “FINISHED”, “FAILED”, “KILLED”] Allowed values: `RUNNING`, `SCHEDULED`, `FINISHED`, `FAILED`, `KILLED`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMlMlflowUpdateRun
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMlMlflowExperiment

Property	Type	Description
`artifact_location`	String	—
`experiment_id`	String	Example: `1`
`lifecycle_stage`	String	Example: `active`
`name`	String	Example: `my_experiment`
`tags`	Array of APIEntitiesMlMlflowKeyValue	—

APIEntitiesMlMlflowGetDownload

Property	Type	Description
`artifact_uri`	String	Download URI for MLflow artifact

APIEntitiesMlMlflowGetExperiment

Property	Type	Description
`experiment`	APIEntitiesMlMlflowExperiment	—

APIEntitiesMlMlflowGetMetricHistory

Property	Type	Description
`metrics`	Array of APIEntitiesMlMlflowMetric	—
`next_page_token`	String	—

APIEntitiesMlMlflowKeyValue

Property	Type	Description
`key`	String	—
`value`	String	—

APIEntitiesMlMlflowListExperiment

Property	Type	Description
`experiments`	Array of APIEntitiesMlMlflow Experiment	—

APIEntitiesMlMlflowMetric

Property	Type	Description
`key`	String	—
`step`	Integer	—
`timestamp`	Integer	Unix timestamp in milliseconds
`value`	Number	—

APIEntitiesMlMlflowModelVersion

Property	Type	Description
`aliases`	Array of strings	—
`creation_timestamp`	Integer	—
`current_stage`	String	Example: `development`
`description`	String	—
`last_updated_timestamp`	Integer	—
`name`	String	—
`run_id`	String	—
`run_link`	String	—
`source`	String	—
`status`	String	Example: `READY`
`status_message`	String	—
`tags`	Array of APIEntitiesMlMlflowKeyValue	—
`user_id`	String	—
`version`	String	—

APIEntitiesMlMlflowNewExperiment

Property	Type	Description
`experiment_id`	String	Example: `1`

APIEntitiesMlMlflowRegisteredModel

Property	Type	Description
`creation_timestamp`	Integer	—
`description`	String	—
`last_updated_timestamp`	Integer	—
`latest_versions`	Array of APIEntitiesMlMlflowModelVersion	—
`name`	String	—
`tags`	Array of APIEntitiesMlMlflowKeyValue	—
`user_id`	String	—

APIEntitiesMlMlflowRun

Property	Type	Description
`data`	Object	—
`data.metrics`	Array of APIEntitiesMlMlflowMetric	—
`data.params`	Array of APIEntitiesMlMlflowKeyValue	—
`data.tags`	Array of APIEntitiesMlMlflowKeyValue	—
`info`	APIEntitiesMlMlflowRunInfo	—

APIEntitiesMlMlflowRunInfo

Property	Type	Description
`artifact_uri`	String	—
`end_time`	Integer	Unix timestamp in milliseconds
`experiment_id`	String	—
`lifecycle_stage`	String	Example: `active`
`run_id`	String	—
`run_name`	String	—
`run_uuid`	String	—
`start_time`	Integer	Unix timestamp in milliseconds
`status`	String	Example: `FINISHED`
`user_id`	String	—

APIEntitiesMlMlflowUpdateRun

Property	Type	Description
`run_info`	APIEntitiesMlMlflowRunInfo	—

Markdown

Use this API to render Markdown content as HTML.

All requests to this API must be authenticated.

> The availability of this feature is controlled by a feature flag. This feature is available for testing, but not ready for production use.

Render Markdown content

`POST /api/v4/markdown`

Renders Markdown content as HTML.

Request body (`application/json`)

Property	Type	Description
`gfm`	Boolean	Render text using GitLab Flavored Markdown Default: `false`
`project`	String	Project used as context when creating references with GitLab Flavored Markdown
`text` Required	String	Markdown text to render

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMarkdown
`400`	Bad request	—
`401`	Unauthorized	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMarkdown

Property	Type	Description
`html`	String	Example: `<p dir=\"auto\">Hello world!</p>"`

Members

Use this API to manage the members of a group or a project, including inherited memberships, membership approvals, and the billable members of a group.

List all direct members of a group

`GET /api/v4/groups/{id}/members`

Lists all direct members of a specified group viewable by the authenticated user. Does not return inherited members from ancestor groups or invited groups.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`query` Query	String	A query string to search for members
`user_ids` Query	Array of integers	Array of user ids to look up for membership
`skip_users` Query	Array of integers	Array of user ids to be skipped for membership
`show_seat_info` Query	Boolean	Show seat information for members
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Add a member to a group

`POST /api/v4/groups/{id}/members`

Adds a member to a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	A valid access level
`expires_at`	String (date-time)	Date string in the format YEAR-MONTH-DAY
`invite_source`	String	Source that triggered the member creation process Default: `members-api`
`user_id`	Integer or string	The user ID of the new member or multiple IDs separated by commas. Mutually exclusive with `username` . At least one of `user_id` , `username` must be provided
`username`	String	The username of the new member or multiple usernames separated by commas. Mutually exclusive with `user_id` . At least one of `user_id` , `username` must be provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

List all members of a group

`GET /api/v4/groups/{id}/members/all`

Lists all members of a specified group viewable by the authenticated user. Also returns inherited members from ancestor groups or invited groups. If a user is a member of this group and one or more ancestor groups, only returns the highest `access_level`. Members from an invited group are returned if the invited group is public, the requester is a member of an invited group, or the requester is a member of the shared group or project.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`query` Query	String	A query string to search for members
`user_ids` Query	Array of integers	Array of user ids to look up for membership
`show_seat_info` Query	Boolean	Show seat information for members
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a group member

`GET /api/v4/groups/{id}/members/all/{user_id}`

Retrieves a specified member of a group. Returns direct members and members inherited or invited through ancestor groups.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`user_id` Path, required	Integer	The user ID of the member

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a direct group member

`GET /api/v4/groups/{id}/members/{user_id}`

Retrieves a specified member of a group. Returns only direct members and not inherited members through ancestor groups.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`user_id` Path, required	Integer	The user ID of the member

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Update a group member

`PUT /api/v4/groups/{id}/members/{user_id}`

Updates a specified member of a group.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`user_id` Path, required	Integer	The user ID of the new member

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	A valid access level
`expires_at`	String (date-time)	Date string in the format YEAR-MONTH-DAY

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Remove a member from a group

`DELETE /api/v4/groups/{id}/members/{user_id}`

Removes a specified user from a group. The user must be a direct member.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`user_id` Path, required	Integer	The user ID of the member
`skip_subresources` Query	Boolean	If `true`, the member retains any direct memberships in subgroups or projects Default: `false`
`unassign_issuables` Query	Boolean	If `true`, unassigns the member from any issues or merge requests in the group Default: `false`

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

List all direct members of a project

`GET /api/v4/projects/{id}/members`

Lists all direct members of a specified project viewable by the authenticated user. Does not return inherited members from ancestor groups or invited groups.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`query` Query	String	A query string to search for members
`user_ids` Query	Array of integers	Array of user ids to look up for membership
`skip_users` Query	Array of integers	Array of user ids to be skipped for membership
`show_seat_info` Query	Boolean	Show seat information for members
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Add a member to a project

`POST /api/v4/projects/{id}/members`

Adds a member to a specified project.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	A valid access level
`expires_at`	String (date-time)	Date string in the format YEAR-MONTH-DAY
`invite_source`	String	Source that triggered the member creation process Default: `members-api`
`user_id`	Integer or string	The user ID of the new member or multiple IDs separated by commas. Mutually exclusive with `username` . At least one of `user_id` , `username` must be provided
`username`	String	The username of the new member or multiple usernames separated by commas. Mutually exclusive with `user_id` . At least one of `user_id` , `username` must be provided

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

List all members of a project

`GET /api/v4/projects/{id}/members/all`

Lists all members of a specified project viewable by the authenticated user. Also returns inherited members from ancestor groups or invited groups. If a user is a member of this project and one or more ancestor groups, only returns the highest `access_level`. Members from an invited group are returned if the invited group is public, the requester is a member of an invited group, or the requester is a member of the shared group or project.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`query` Query	String	A query string to search for members
`user_ids` Query	Array of integers	Array of user ids to look up for membership
`show_seat_info` Query	Boolean	Show seat information for members
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a project member

`GET /api/v4/projects/{id}/members/all/{user_id}`

Retrieves a specified member of a project. Returns direct members and members inherited or invited through ancestor groups.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`user_id` Path, required	Integer	The user ID of the member

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a direct project member

`GET /api/v4/projects/{id}/members/{user_id}`

Retrieves a specified member of a project. Returns only direct members and not inherited members through ancestor groups.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`user_id` Path, required	Integer	The user ID of the member

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Update a project member

`PUT /api/v4/projects/{id}/members/{user_id}`

Updates a specified member of a project.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`user_id` Path, required	Integer	The user ID of the new member

Request body (`application/json`)

Property	Type	Description
`access_level` Required	Integer	A valid access level
`expires_at`	String (date-time)	Date string in the format YEAR-MONTH-DAY

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMember
`400`	Bad Request	—
`404`	Not Found	—

Remove a member from a project

`DELETE /api/v4/projects/{id}/members/{user_id}`

Removes a specified user from a project. The user must be a direct member.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`user_id` Path, required	Integer	The user ID of the member
`skip_subresources` Query	Boolean	If `true`, the member retains any direct memberships in subgroups or projects Default: `false`
`unassign_issuables` Query	Boolean	If `true`, unassigns the member from any issues or merge requests in the project Default: `false`

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesMember

Property	Type	Description
`access_level`	String	—
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`created_at`	String	—
`created_by`	APIEntitiesUserBasic	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`expires_at`	String	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`two_factor_enabled`	Boolean	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Merge request approvals

Use this API to manage merge request approvals and the group and project approval settings that govern them. All endpoints require authentication.

Endpoints to approve, unapprove, reset approvals, and retrieve approval state are available on all tiers, including Free. All other endpoints require Premium or Ultimate, and each one shows its tier.

> The `merge_request_iid` and `id` path parameters must each be a single value. Passing multiple space-separated values (for example, `451 454 458`) is not supported and returns `400 Bad Request`. To act on multiple merge requests, make one request per merge request IID.

Retrieve approval state for a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/approvals`

Retrieves the approval state for a specified merge request. In the response, `approved_by` contains information about all approvers of the merge request, regardless of whether those approvals satisfy any approval rule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestApprovals
`400`	Bad Request	—
`404`	Not found	—

Approve merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/approve`

Approves a specified merge request. The currently authenticated user must be an eligible approver. The `sha` parameter ensures you are approving the current version of the merge request. If defined, the value must match the merge request's HEAD commit SHA. A mismatch returns a `409 Conflict` response.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Request body (`application/json`)

Property	Type	Description
`publish_review`	Boolean	If `true`, submits pending review comments
`sha`	String	SHA at `HEAD` of the source branch of the merge request

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequestApprovals
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Reset approvals for a merge request

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}/reset_approvals`

Resets all approvals for a specified merge request. Available only to bot users with a valid project or group token. Human users receive a `401 Unauthorized` response.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Unapprove a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/unapprove`

Unapproves a merge request. Removes the approval for the currently authenticated user from a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequestApprovals
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesApprovals

Property	Type	Description
`approved_at`	String (date-time)	Example: `2025-01-01T10:00:00Z`
`user`	APIEntitiesUserBasic	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesMergeRequestApprovals

Property	Type	Description
`approved`	Boolean	—
`approved_by`	APIEntitiesApprovals	—
`user_can_approve`	Boolean	—
`user_has_approved`	Boolean	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Merge requests

Use this API to manage merge requests in a project, and to list the merge requests across a group or an entire instance. You can:

- Automate any part of the code review process.
- Connect code changes to external tools.
- Send merge request information to non-GitLab systems in your preferred format.
- Update, approve, merge, or block merge requests based on data from external systems.
- Add commits from previous merge requests to a merge request for more context.

All API calls to non-public information require authentication.

List all group merge requests

`GET /api/v4/groups/{id}/merge_requests`

Lists all merge requests for a specified group and any subgroups.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`author_id` Query	Integer	Returns merge requests created by the given user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Returns merge requests created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Returns merge requests created by the given `username`. Mutually exclusive with `assignee_id`
`reviewer_username` Query	String	Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `reviewer_id`
`labels` Query	Array of strings	Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels.

Name	Type	Description
		`Any` lists all merge requests with at least one label. Predefined names are case-insensitive
`milestone` Query	String	Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`my_reaction_emoji` Query	String	Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`reviewer_id` Query	Integer or string	Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `reviewer_username`
`state` Query	String	Filter merge requests by state Allowed values: `opened`, `closed`, `locked`, `merged`, `all` Default: `all`
`order_by` Query	String	Returns merge requests ordered by `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at` or `merged_at` fields. Introduced in GitLab 14.8 Allowed values: `created_at`,

Name	Type	Description
		`label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at`, `merged_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`with_merge_status_recheck` Query	Boolean	If `true`, this projection requests (but does not guarantee) an asynchronous recalculation of the `merge_status` field. Enable the `restrict_merge_status_recheck` feature flag to ignore this attribute when requested by users without the Developer, Maintainer, or Owner role Default: `false`
`created_after` Query	String (date-time)	Return merge requests created on or after the specified time Example: `2019-03-15T08:00:00Z`
`created_before` Query	String (date-time)	Return merge requests created on or before the specified time Example: `2019-03-15T08:00:00Z`
`updated_after` Query	String (date-time)	Return merge requests updated on or after the specified time

Name	Type	Description
		Example: `2019-03-15T08:00:00Z`
`updated_before` Query	String (date-time)	Return merge requests updated on or before the specified time Example: `2019-03-15T08:00:00Z`
`merged_after` Query	String (date-time)	Return merge requests merged on or after the specified time Example: `2019-03-15T08:00:00Z`
`merged_before` Query	String (date-time)	Return merge requests merged on or before the specified time Example: `2019-03-15T08:00:00Z`
`view` Query	String	If `simple`, returns the `iid`, URL, title, description, and basic state of merge request Allowed values: `simple`
`scope` Query	String	Return merge requests for the given scope. `reviews_for_me` returns merge requests where the current user is assigned as a reviewer Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `reviews_for_me`, `all`
`source_branch` Query	String	Return merge requests with the given source branch
`source_project_id` Query	Integer	Return merge requests with the given source project ID
`target_branch` Query	String	Return merge requests with the given target branch

Name	Type	Description
`search` Query	String	Search merge requests against their `title` and `description`. Combine with the `in` attribute
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description` Example: `title,description`
`wip` Query	String	Filter merge requests by their `wip` status. `yes` returns only draft merge requests, `no` returns non-draft merge requests. Deprecated in GitLab 19.0. Use `draft` instead. Mutually exclusive with `draft` Allowed values: `yes`, `no`
`draft` Query	Boolean	If `true`, returns only draft merge requests. If `false`, returns only non-draft merge requests. Mutually exclusive with `wip`
`not` Query	Object	Return merge requests that do not match the parameters supplied. Accepts: `labels`, `milestone`, `author_id`, `author_username`, `assignee_id`, `assignee_username`, `reviewer_id`, `reviewer_username`, `my_reaction_emoji`
`not[author_id]` Query	Integer	`<Negated>` Returns merge requests created by the given user `id`. Combine with `scope=all` or

Name	Type	Description
		`scope=assigned_to_me`. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer or string	`<Negated>` Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[assignee_id]`
`not[reviewer_username]` Query	String	`<Negated>` Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `not[reviewer_id]`
`not[labels]` Query	Array of strings	`<Negated>` Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one label. Predefined names are case-insensitive

Name	Type	Description
`not[milestone]` Query	String	`<Negated>` Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`not[my_reaction_emoji]` Query	String	`<Negated>` Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`not[reviewer_id]` Query	Integer	`<Negated>` Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `not[reviewer_username]`
`deployed_before` Query	String (date-time)	Return merge requests deployed before the specified time Example: `2019-03-15T08:00:00Z`
`deployed_after` Query	String (date-time)	Return merge requests deployed after the specified time Example: `2019-03-15T08:00:00Z`
`environment` Query	String	Return merge requests deployed to the given environment Example: `production`

Name	Type	Description
`merge_user_id` Query	Integer	Returns merge requests which have been merged by the user with the given user `id`. Mutually exclusive with `merge_user_username`
`merge_user_username` Query	String	Returns merge requests which have been merged by the user with the given `username`. Mutually exclusive with `merge_user_id`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`non_archived` Query	Boolean	If `true`, returns merge requests from non-archived projects only Default: `true`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

List all merge requests

`GET /api/v4/merge_requests`

Lists all merge requests accessible to the authenticated user. By default, returns only merge requests created by the current user. Use `scope=all` to get all merge requests.

Parameters

Name	Type	Description
`author_id` Query	Integer	Returns merge requests created by the given user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Returns merge requests created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Returns merge requests created by the given `username`. Mutually exclusive with `assignee_id`
`reviewer_username` Query	String	Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `reviewer_id`
`labels` Query	Array of strings	Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one label.

Name	Type	Description
		Predefined names are case-insensitive
`milestone` Query	String	Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`my_reaction_emoji` Query	String	Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`reviewer_id` Query	Integer or string	Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `reviewer_username`
`state` Query	String	Filter merge requests by state Allowed values: `opened`, `closed`, `locked`, `merged`, `all` Default: `all`
`order_by` Query	String	Returns merge requests ordered by `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at` or `merged_at` fields. Introduced in GitLab 14.8 Allowed values: `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`,

Name	Type	Description
		`title`, `updated_at`, `merged_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`with_merge_status_recheck` Query	Boolean	If `true`, this projection requests (but does not guarantee) an asynchronous recalculation of the `merge_status` field. Enable the `restrict_merge_status_recheck` feature flag to ignore this attribute when requested by users without the Developer, Maintainer, or Owner role Default: `false`
`created_after` Query	String (date-time)	Return merge requests created on or after the specified time Example: `2019-03-15T08:00:00Z`
`created_before` Query	String (date-time)	Return merge requests created on or before the specified time Example: `2019-03-15T08:00:00Z`
`updated_after` Query	String (date-time)	Return merge requests updated on or after the specified time

Name	Type	Description
		Example: `2019-03-15T08:00:00Z`
`updated_before` Query	String (date-time)	Return merge requests updated on or before the specified time Example: `2019-03-15T08:00:00Z`
`merged_after` Query	String (date-time)	Return merge requests merged on or after the specified time Example: `2019-03-15T08:00:00Z`
`merged_before` Query	String (date-time)	Return merge requests merged on or before the specified time Example: `2019-03-15T08:00:00Z`
`view` Query	String	If `simple`, returns the `iid`, URL, title, description, and basic state of merge request Allowed values: `simple`
`scope` Query	String	Return merge requests for the given scope. `reviews_for_me` returns merge requests where the current user is assigned as a reviewer Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `reviews_for_me`, `all` Default: `created_by_me`
`source_branch` Query	String	Return merge requests with the given source branch
`source_project_id` Query	Integer	Return merge requests with the given source project ID
	String	

Name	Type	Description
`target_branch` Query		Return merge requests with the given target branch
`search` Query	String	Search merge requests against their `title` and `description`. Combine with the `in` attribute
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description` Example: `title,description`
`wip` Query	String	Filter merge requests by their `wip` status. `yes` returns only draft merge requests, `no` returns non-draft merge requests. Deprecated in GitLab 19.0. Use `draft` instead. Mutually exclusive with `draft` Allowed values: `yes`, `no`
`draft` Query	Boolean	If `true`, returns only draft merge requests. If `false`, returns only non-draft merge requests. Mutually exclusive with `wip`
`not` Query	Object	Return merge requests that do not match the parameters supplied. Accepts: `labels`, `milestone`, `author_id`, `author_username`, `assignee_id`, `assignee_username`, `reviewer_id`, `reviewer_username`, `my_reaction_emoji`
	Integer	`<Negated>` Returns merge requests created by the given

Name	Type	Description
`not[author_id]` Query		user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer or string	`<Negated>` Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[assignee_id]`
`not[reviewer_username]` Query	String	`<Negated>` Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `not[reviewer_id]`
`not[labels]` Query	Array of strings	`<Negated>` Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one

Name	Type	Description
		label. Predefined names are case-insensitive
`not[milestone]` Query	String	`<Negated>` Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`not[my_reaction_emoji]` Query	String	`<Negated>` Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`not[reviewer_id]` Query	Integer	`<Negated>` Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `not[reviewer_username]`
`deployed_before` Query	String (date-time)	Return merge requests deployed before the specified time Example: `2019-03-15T08:00:00Z`
`deployed_after` Query	String (date-time)	Return merge requests deployed after the specified time Example: `2019-03-15T08:00:00Z`
	String	Return merge requests deployed to the given

Name	Type	Description
`environment` Query		environment Example: `production`
`merge_user_id` Query	Integer	Returns merge requests which have been merged by the user with the given user `id`. Mutually exclusive with `merge_user_username`
`merge_user_username` Query	String	Returns merge requests which have been merged by the user with the given `username`. Mutually exclusive with `merge_user_id`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`non_archived` Query	Boolean	If `true`, returns merge requests from non-archived projects only Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`422`	Unprocessable entity	—

List all project merge requests

`GET /api/v4/projects/{id}/merge_requests`

Lists all project merge requests.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`author_id` Query	Integer	Returns merge requests created by the given user `id`. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Returns merge requests created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Returns merge requests created by the given `username`. Mutually exclusive with `assignee_id`
`reviewer_username` Query	String	Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `reviewer_id`
`labels` Query	Array of strings	Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels.

Name	Type	Description
		`Any` lists all merge requests with at least one label. Predefined names are case-insensitive
`milestone` Query	String	Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`my_reaction_emoji` Query	String	Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`reviewer_id` Query	Integer or string	Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `reviewer_username`
`state` Query	String	Filter merge requests by state Allowed values: `opened`, `closed`, `locked`, `merged`, `all` Default: `all`
`order_by` Query	String	Returns merge requests ordered by `created_at`, `label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at` or `merged_at` fields. Introduced in GitLab 14.8 Allowed values: `created_at`,

Name	Type	Description
		`label_priority`, `milestone_due`, `popularity`, `priority`, `title`, `updated_at`, `merged_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`with_merge_status_recheck` Query	Boolean	If `true`, this projection requests (but does not guarantee) an asynchronous recalculation of the `merge_status` field. Enable the `restrict_merge_status_recheck` feature flag to ignore this attribute when requested by users without the Developer, Maintainer, or Owner role Default: `false`
`created_after` Query	String (date-time)	Return merge requests created on or after the specified time Example: `2019-03-15T08:00:00Z`
`created_before` Query	String (date-time)	Return merge requests created on or before the specified time Example: `2019-03-15T08:00:00Z`
`updated_after` Query	String (date-time)	Return merge requests updated on or after the specified time

Name	Type	Description
		Example: `2019-03-15T08:00:00Z`
`updated_before` Query	String (date-time)	Return merge requests updated on or before the specified time Example: `2019-03-15T08:00:00Z`
`merged_after` Query	String (date-time)	Return merge requests merged on or after the specified time Example: `2019-03-15T08:00:00Z`
`merged_before` Query	String (date-time)	Return merge requests merged on or before the specified time Example: `2019-03-15T08:00:00Z`
`view` Query	String	If `simple`, returns the `iid`, URL, title, description, and basic state of merge request Allowed values: `simple`
`scope` Query	String	Return merge requests for the given scope. `reviews_for_me` returns merge requests where the current user is assigned as a reviewer Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `reviews_for_me`, `all`
`source_branch` Query	String	Return merge requests with the given source branch
`source_project_id` Query	Integer	Return merge requests with the given source project ID
`target_branch` Query	String	Return merge requests with the given target branch

Name	Type	Description
`search` Query	String	Search merge requests against their `title` and `description`. Combine with the `in` attribute
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description` Example: `title,description`
`wip` Query	String	Filter merge requests by their `wip` status. `yes` returns only draft merge requests, `no` returns non-draft merge requests. Deprecated in GitLab 19.0. Use `draft` instead. Mutually exclusive with `draft` Allowed values: `yes`, `no`
`draft` Query	Boolean	If `true`, returns only draft merge requests. If `false`, returns only non-draft merge requests. Mutually exclusive with `wip`
`not` Query	Object	Return merge requests that do not match the parameters supplied. Accepts: `labels`, `milestone`, `author_id`, `author_username`, `assignee_id`, `assignee_username`, `reviewer_id`, `reviewer_username`, `my_reaction_emoji`
`not[author_id]` Query	Integer	`<Negated>` Returns merge requests created by the given user `id`. Combine with `scope=all` or

Name	Type	Description
		`scope=assigned_to_me`. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer or string	`<Negated>` Returns merge requests assigned to the given user `id`. `None` returns unassigned merge requests. `Any` returns merge requests with an assignee. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	`<Negated>` Returns merge requests created by the given `username`. Mutually exclusive with `not[assignee_id]`
`not[reviewer_username]` Query	String	`<Negated>` Returns merge requests which have the user as a reviewer with the given `username`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Introduced in GitLab 13.8. Mutually exclusive with `not[reviewer_id]`
`not[labels]` Query	Array of strings	`<Negated>` Returns merge requests matching a comma-separated list of labels. `None` lists all merge requests with no labels. `Any` lists all merge requests with at least one label. Predefined names are case-insensitive

Name	Type	Description
`not[milestone]` Query	String	`<Negated>` Returns merge requests for a specific milestone. `None` returns merge requests with no milestone. `Any` returns merge requests that have an assigned milestone
`not[my_reaction_emoji]` Query	String	`<Negated>` Returns merge requests reacted by the authenticated user by the given `emoji`. `None` returns issues not given a reaction. `Any` returns issues given at least one reaction
`not[reviewer_id]` Query	Integer	`<Negated>` Returns merge requests which have the user as a reviewer with the given user `id`. `None` returns merge requests with no reviewers. `Any` returns merge requests with any reviewer. Mutually exclusive with `not[reviewer_username]`
`deployed_before` Query	String (date-time)	Return merge requests deployed before the specified time Example: `2019-03-15T08:00:00Z`
`deployed_after` Query	String (date-time)	Return merge requests deployed after the specified time Example: `2019-03-15T08:00:00Z`
`environment` Query	String	Return merge requests deployed to the given environment Example: `production`

Name	Type	Description
`merge_user_id` Query	Integer	Returns merge requests which have been merged by the user with the given user `id`. Mutually exclusive with `merge_user_username`
`merge_user_username` Query	String	Returns merge requests which have been merged by the user with the given `username`. Mutually exclusive with `merge_user_id`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`iids` Query	Array of integers	Return merge requests matching the provided internal IDs

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create a merge request

`POST /api/v4/projects/{id}/merge_requests`

Creates a merge request for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`add_labels`	Array of strings	Comma-separated label names to add to the merge request. If a label does not already exist, this creates a new project label and assigns it to the merge request
`allow_collaboration`	Boolean	If `true`, allows commits from members who can merge to the target branch
`allow_maintainer_to_push`	Boolean	If `true`, allows commits from members who can merge to the target branch. Deprecated. Use `allow_collaboration` instead
`assignee_id`	Integer	ID of the user to assign to the merge request. Set to `0` or leave empty to unassign all assignees
`assignee_ids`	Array of integers	IDs of the users to assign to the merge request, as a comma-separated list. Set to `0` or leave empty to unassign all assignees
`description`	String	Description of the merge request. Limited to 1,048,576 characters
`labels`	Array of strings	Comma-separated label names for the merge request. If a label does not already exist, this creates a new project label and assigns it to the merge request. Set to an empty string to unassign all labels
`merge_after`	String	Date after which the merge request can be merged. Introduced in GitLab 17.8

Property	Type	Description
`milestone`	String	The title of a project or ancestor-group milestone to assign the merge request to. Mutually exclusive with `milestone_id` Maximum length: `255`
`milestone_id`	Integer	Global ID of a milestone to assign to the merge request. Set to `0` or leave empty to unassign the milestone. Mutually exclusive with `milestone`
`remove_labels`	Array of strings	Comma-separated label names to remove from the merge request
`remove_source_branch`	Boolean	If `true`, the source branch is removed when the merge request merges
`reviewer_ids`	Array of integers	IDs of the users to set as reviewers of the merge request, as a comma-separated list. If set to `0` or left empty, no reviewers are set
`source_branch` Required	String	Source branch
`squash`	Boolean	If `true`, squashes all commits into a single commit on merge. When omitted, defaults to the project's squash option setting, which can override this value at merge time
`target_branch` Required	String	Target branch
`target_project_id`	Integer	

Property	Type	Description
		ID of the target project for the merge request. If omitted, defaults to the source project
`title` Required	String	Title of the merge request

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequest
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`409`	Conflict	—
`422`	Unprocessable entity	—

Retrieve a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}`

Retrieves a merge request for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`render_html` Query	Boolean	If `true`, the response includes rendered HTML for title and description
`include_diverged_commits_count` Query	Boolean	If `true`, the response includes the commits behind the target branch
`include_rebase_in_progress` Query	Boolean	If `true`, the response includes whether a rebase operation is in progress

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequest
`400`	Bad Request	—
`404`	Not found	—

Update a merge request

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}`

Updates a merge request for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`add_labels`	Array of strings	Comma-separated label names to add to the merge request. If a label does not already exist, this creates a new project label and assigns it to the merge request. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`allow_collaboration`	Boolean	If `true`, allows commits from members who can merge to the target branch. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`allow_maintainer_to_push`	Boolean	If `true`, allows commits from members who can merge to the target branch. Deprecated. Use `allow_collaboration` instead. At least one of `assignee_id`, `assignee_ids`,

Property	Type	Description
		`reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`assignee_id`	Integer	ID of the user to assign to the merge request. Set to `0` or leave empty to unassign all assignees. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`assignee_ids`	Array of integers	IDs of the users to assign to the merge request, as a comma-separated list. Set to `0` or leave empty to unassign all assignees. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_`

Property	Type	Description
		`maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`description`	String	Description of the merge request. Limited to 1,048,576 characters. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`discussion_locked`	Boolean	If `true`, locks the merge request's discussion so only project members can add, edit, or resolve comments. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`labels`	Array of strings	

Property	Type	Description
		Comma-separated label names for the merge request. If a label does not already exist, this creates a new project label and assigns it to the merge request. Set to an empty string to unassign all labels. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`merge_after`	String	Date after which the merge request can be merged. Introduced in GitLab 17.8. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`milestone`	String	The title of a project or ancestor-group milestone to assign the merge request to. Mutually exclusive with

Property	Type	Description
		`milestone_id`. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided Maximum length: `255`
`milestone_id`	Integer	Global ID of a milestone to assign to the merge request. Set to `0` or leave empty to unassign the milestone. Mutually exclusive with `milestone`. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`remove_labels`	Array of strings	Comma-separated label names to remove from the merge request. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_`

Property	Type	Description
		`id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`remove_source_branch`	Boolean	If `true`, the source branch is removed when the merge request merges. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`reviewer_ids`	Array of integers	IDs of the users to set as reviewers of the merge request, as a comma-separated list. If set to `0` or left empty, no reviewers are set. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`,

Property	Type	Description
		`discussion_locked`, `merge_after` must be provided
`squash`	Boolean	If `true`, squashes all commits into a single commit on merge. When omitted, defaults to the project's squash option setting, which can override this value at merge time. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`state_event`	String	Event to change the state of the merge request. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided Allowed values: `close`, `reopen`
`target_branch`	String	Target branch. At least one of `assignee_id`, `assignee_ids`,

Property	Type	Description
		`reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided
`title`	String	Title of the merge request. At least one of `assignee_id`, `assignee_ids`, `reviewer_ids`, `description`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `remove_source_branch`, `allow_collaboration`, `allow_maintainer_to_push`, `squash`, `target_branch`, `title`, `state_event`, `discussion_locked`, `merge_after` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequest
`400`	Bad request	—
`404`	Not found	—
`409`	Conflict	—
`422`	Unprocessable entity	—

Delete a merge request

`DELETE /api/v4/projects/{id}/merge_requests/{merge_request_iid}`

Deletes a specified merge request for a project. Administrators and project Owners only.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`412`	Precondition failed	—

Add spent time for a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/add_spent_time`

Adds spent time for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of the merge_ request

Request body (`application/json`)

Property	Type	Description
`duration` Required	String	The duration in human format

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Cancel auto merge

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/cancel_auto_merge`

Cancels an automatic merge for a merge request that has been set to merge automatically. Use this endpoint instead of `cancel_merge_when_pipeline_succeeds`, which returns a status hash rather than the merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—

Cancel merge when pipeline succeeds (deprecated)

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/cancel_merge_when_pipeline_succeeds`

Cancels an automatic merge for a merge request that has been set to merge when the pipeline succeeds. This endpoint does not return the merge request. Deprecated in GitLab 19.5. Use POST /projects/:id/merge_requests/:merge_request_iid/cancel_auto_merge instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`201`	The status field of the response body is success when an auto-merge was canceled, and error when the merge request had none set	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`405`	Method not allowed	—

Retrieve merge request changes

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/changes`

Retrieves changes for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`unidiff` Query	Boolean	If `true`, presents diffs in the unified diff format Default: `false`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestChanges
`400`	Bad Request	—
`404`	Not found	—

List all issues that close on merge

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/closes_issues`

Lists all issues that close on merge.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMRNote
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve merge request commits

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/commits`

Retrieves commits for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCommit
`400`	Bad Request	—
`404`	Not found	—

List all context commits for a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/context_commits`

Lists all context commits for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCommitWithLink
`400`	Bad Request	—
`404`	Not found	—

Create context commits for a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/context_commits`

Creates context commits for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`commits` Required	Array of strings	Context commits' SHAs

Responses

Code	Description	Schema
`201`	Created	Array of APIEntitiesCommit
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete context commits from a merge request

`DELETE /api/v4/projects/{id}/merge_requests/{merge_request_iid}/context_commits`

Deletes specified context commits from a merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`commits` Query, required	Array of strings	Context commits’ SHAs
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

List all merge request diffs

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/diffs`

Lists all merge request diffs.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`unidiff` Query	Boolean	If `true`, presents diffs in the unified diff format Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDiff
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Merge a merge request

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}/merge`

Merges a merge request. Accepts and merges changes submitted with the merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`auto_merge`	Boolean	If `true` , the merge request merges when checks pass
`merge_commit_message`	String	Custom merge commit message
`merge_when_pipeline_succeeds`	Boolean	If `true` , merges the merge request automatically once the pipeline succeeds. Deprecated in GitLab 17.11. Use `auto_merge` instead
`sha`	String	SHA that must match the HEAD of the source branch, otherwise the merge fails. Use to ensure that only reviewed commits are merged. Required if the require a commit SHA on the merge requests API setting is enabled for the group or instance
`should_remove_source_branch`	Boolean	If `true` , removes the source branch
`skip_merge_train`	Boolean	If `true` , avoids restarting the merge train when merging immediately
`squash`	Boolean	If `true` , squashes all commits into a single commit on merge
`squash_commit_message`	String	Custom squash commit message

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequest
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`405`	Method not allowed	—
`409`	Conflict	—
`422`	Unprocessable entity	—

Merge to default merge ref path

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/merge_ref`

Merges the changes between the merge request source and target branches into the `refs/merge-requests/:iid/merge` ref, of the target project repository, if possible. This ref has the state the target branch would have if a regular merge action was taken.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`404`	Not Found	—

Retrieve merge request participants

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/participants`

Retrieves participants for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`404`	Not found	—

List all merge request pipelines

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/pipelines`

Lists all merge request pipelines.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiPipelineBasic
`400`	Bad Request	—
`404`	Not found	—

Create a merge request pipeline

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/pipelines`

Creates a merge request pipeline. Pipelines created with this operation must configure `.gitlab-ci.yml` with `only: [merge_requests]` to create jobs.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`async`	Boolean	If `true` , creates the merge request pipeline asynchronously outside the API request, returning an empty response with a `202` status code. Pipeline creation can still fail after a `202` response Default: `false`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad request	—
`404`	Not found	—
`405`	Method not allowed	—

Retrieve merge request raw diffs

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/raw_diffs`

Retrieves the raw diffs of the files changed in a merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Rebase a merge request

`PUT /api/v4/projects/{id}/merge_requests/{merge_request_iid}/rebase`

Rebases a merge request. Automatically rebases the `source_branch` of the merge request against its `target_branch`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Request body (`application/json`)

Property	Type	Description
`skip_ci`	Boolean	If `true`, skips creating a CI pipeline

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—

List all issues related to the merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/related_issues`

Lists all issues related to the merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Reset spent time for a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/reset_spent_time`

Resets the total spent time for a specified merge request to `0` seconds.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of the merge_ request

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Reset the estimated time for a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/reset_time_estimate`

Resets the estimated time for a specified merge request to `0` seconds.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of the merge_ request

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve merge request reviewers

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/reviewers`

Retrieves reviewers for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestReviewer
`400`	Bad Request	—
`404`	Not found	—

Set the estimated time for a merge request

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/time_estimate`

Sets an estimated time of work for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of the merge_request

Request body (`application/json`)

Property	Type	Description
`duration` Required	String	The duration in human format Example: `3h30m`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssuableTimeStats
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve time tracking stats for a merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/time_stats`

Retrieves time tracking stats for a specified merge request, including time estimate and time spent in seconds and human-readable format (for example, `1h 30m`).

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of the merge_request

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssuableTimeStats
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve merge request diff versions

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/versions`

Retrieves merge request diff versions.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeRequestDiff
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge request diff version

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/versions/{version_id}`

Retrieves a merge request diff version.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	Internal ID of the merge request
`version_id` Path, required	Integer	ID of the merge request diff version
`unidiff` Query	Boolean	If `true`, presents diffs in the unified diff format Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestDiffFull
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiPipeline

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`
`tag`	Boolean	Example: `false`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCommitWithLink

Property	Type	Description
`author`	APIEntitiesUserPath	—
`author_email`	String	Example: `john@example.com`
`author_gravatar_url`	String	Example: `https://www.gravatar.com/avatar/abc`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`commit_path`	String	Example: `/project/-/commit/abc`
`commit_url`	String	Example: `https://gitlab.example.com/project/-/commit/abc`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`description_html`	String	Example: `<p>description</p>`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`next_commit_id`	String	

Property	Type	Description
		Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`pipeline_status_path`	String	Example: `/project/pipelines/1`
`prev_commit_id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`short_id`	String	Example: `2695effb`
`signature_html`	String	Example: `<p>signature</p>`
`title`	String	Example: `Initial commit`
`title_html`	String	Example: `<p>title</p>`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiff

Property	Type	Description
`a_mode`	String	Example: `100755`
`b_mode`	String	Example: `100644`
`collapsed`	Boolean	—
`deleted_file`	Boolean	—
`diff`	String	Example: `@@ -71,6 +71,8 @@\n...`
`generated_file`	Boolean	—
`new_file`	Boolean	—
`new_path`	String	Example: `doc/update/5.4-to-6.0.md`
`old_path`	String	Example: `doc/update/5.4-to-6.0.md`
`renamed_file`	Boolean	—
`too_large`	Boolean	—

APIEntitiesDiffRefs

Property	Type	Description
`base_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`head_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`start_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesMergeRequest

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`changes_count`	String	Example: `1`
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`diff_refs`	APIEntitiesDiffRefs	—
`discussion_locked`	Boolean	—
`diverged_commits_count`	Integer	Example: `0`
`downvotes`	Integer	—
`draft`	Boolean	—
`first_contribution`	Boolean	—
`first_deployed_to_production_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`force_remove_source_branch`	Boolean	—

Property	Type	Description
`has_conflicts`	Boolean	—
`head_pipeline`	APIEntitiesCiPipeline	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`
`labels`	Array of strings	—
`latest_build_finished_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`latest_build_started_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_error`	String	Example: `Merge error`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`pipeline`	APIEntitiesCiPipelineBasic	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`

Property	Type	Description
`rebase_in_progress`	Boolean	—
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`
`subscribed`	Boolean	—
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user`	Object	—

Property	Type	Description
`user.can_merge`	Boolean	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMergeRequestChanges

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`changes`	Array of APIEntitiesDiff	—
`changes_count`	String	Example: `1`
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`diff_refs`	APIEntitiesDiffRefs	—
`discussion_locked`	Boolean	—
`diverged_commits_count`	Integer	Example: `0`
`downvotes`	Integer	—
`draft`	Boolean	—
`first_contribution`	Boolean	—
`first_deployed_to_production_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`

Property	Type	Description
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`head_pipeline`	APIEntitiesCiPipeline	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`
`labels`	Array of strings	—
`latest_build_finished_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`latest_build_started_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_error`	String	Example: `Merge error`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`overflow`	Boolean	—
`pipeline`	APIEntitiesCiPipelineBasic	—
`prepared_at`	String (date-time)	

Property	Type	Description
		Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`rebase_in_progress`	Boolean	—
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`
`subscribed`	Boolean	—
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	

Property	Type	Description
		Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user`	Object	—
`user.can_merge`	Boolean	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMergeRequestDiff

Property	Type	Description
`base_commit_sha`	String	Example: `1234abcd`
`created_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`head_commit_sha`	String	Example: `1234abcd`
`id`	Integer (int64)	Example: `1`
`merge_request_id`	Integer (int64)	Example: `1`
`patch_id_sha`	String	Example: `1234abcd`
`real_size`	String	Example: `1`
`start_commit_sha`	String	Example: `1234abcd`
`state`	String	Example: `collected`

APIEntitiesMergeRequestDiffFull

Property	Type	Description
`base_commit_sha`	String	Example: `1234abcd`
`commits`	Array of APIEntitiesCommit	—
`created_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`diffs`	Array of APIEntitiesDiff	—
`head_commit_sha`	String	Example: `1234abcd`
`id`	Integer (int64)	Example: `1`
`merge_request_id`	Integer (int64)	Example: `1`
`patch_id_sha`	String	Example: `1234abcd`
`real_size`	String	Example: `1`
`start_commit_sha`	String	Example: `1234abcd`
`state`	String	Example: `collected`

APIEntitiesMergeRequestReviewer

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`state`	String	Example: `unreviewed`
`user`	APIEntitiesUserBasic	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesMRNote

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`note`	String	Example: `LGTM!`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserPath

Property	Type	Description
`availability`	String	—
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`path`	String	—
`public_email`	String	Example: `john@example.com`
`show_status`	String	—
`state`	String	Example: `active`
`status_tooltip_html`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

DetailedStatusEntity

Property	Type	Description
`action`	Object	—
`action.button_title`	String	Example: `Cancel this job`
`action.confirmation_message`	String	Example: `Are you sure?`
`action.icon`	String	Example: `cancel`
`action.method`	String	Example: `post`
`action.path`	String	Example: `/namespace1/project1/-/jobs/2/cancel`
`action.title`	String	Example: `Cancel`
`details_path`	String	Example: `/test-group/test-project/-/pipelines/287`
`favicon`	String	Example: `/assets/ci_favicons/favicon_status_success.png`
`group`	String	Example: `success`
`has_details`	Boolean	Example: `true`
`icon`	String	Example: `status_success`
`illustration`	Object	Example: `{"content":"This job depends on upstream jobs that need to succeed in order for this job to be triggered","image":"illustrations/empty-state/empty-job-not-triggered-md.svg","size":"","title":"This job has not been triggered yet"}`
`label`	String	Example: `passed`
`text`	String	Example: `passed`
`tooltip`	String	Example: `passed`

Merge trains

Use this API to retrieve the merge trains of a project. These endpoints require the Developer, Maintainer, or Owner role, and support offset-based pagination.

Each object in a response represents one merge request in a train, not an entire train. Requesting merge trains without a target branch can return entries from more than one train in the project.

Responses do not include an explicit queue position, and return merge requests as a flat list rather than grouped by train:

- For an exact queue position, sort by `id` in ascending order, or use the GraphQL API `MergeTrainCar.index` field.
- To group merge requests by train, use the `target_branch` attribute, or query the GraphQL API `Project.mergeTrains` and `MergeTrain.cars` fields.

Read every merge train of a project

`GET /api/v4/projects/{id}/merge_trains`

Returns the cars of all the trains of the project at once. Cars of different target branches are mixed together, so read one train instead when the queue of a single branch is what matters.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`scope` Query	String	Limit the response to cars still queued (`active`) or to cars that have left the queue (`complete`) Allowed values: `active` , `complete` Example: `active`
`sort` Query	String	Order the cars by their position, oldest or newest first Allowed values: `asc` , `desc` Default: `desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeTrainsCar
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Read the place of a merge request in its merge train

`GET /api/v4/projects/{id}/merge_trains/merge_requests/{merge_request_iid}`

Returns the car that represents the merge request in the train of its target branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	IID of the merge request Example: `1`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeTrainsCar
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Queue a merge request on a merge train

`POST /api/v4/projects/{id}/merge_trains/merge_requests/{merge_request_iid}`

Puts the merge request at the back of the train of its target branch, or has it join the queue once its merge checks pass.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	IID of the merge request Example: `1`

Request body (`application/json`)

Property	Type	Description
`auto_merge`	Boolean	Queue the merge request instead of merging it right away
`sha`	String	Head of the source branch as the caller last saw it. The request is rejected when it moved
`squash`	Boolean	Squash the commits of the source branch into one when the merge request merges

Responses

Code	Description	Schema
`201`	Created	Array of APIEntitiesMergeTrainsCar
`202`	Accepted	Array of APIEntitiesMergeTrainsCar
`400`	Cannot be queued	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—

Read the merge train of a target branch

`GET /api/v4/projects/{id}/merge_trains/{target_branch}`

Returns the cars queued for one target branch, in queue order.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`target_branch` Path, required	String	Target branch whose train to read Example: `main`
`scope` Query	String	Limit the response to cars still queued (`active`) or to cars that have left the queue (`complete`) Allowed values: `active` , `complete` Example: `active`
`sort` Query	String	Order the cars by their position, oldest or newest first Allowed values: `asc` , `desc` Default: `desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMergeTrainsCar
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesMergeRequestSimple

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`project_id`	Integer (int64)	Example: `4`
`state`	String	Example: `closed`
`title`	String	Example: `Test MR 1580978354`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`web_url`	String	Example: `http://local.gitlab.test:8181/root/merge-train-race-condition/-/merge_requests/59`

APIEntitiesMergeTrainsCar

Property	Type	Description
`created_at`	String (date-time)	Example: `2026-03-05T09:15:00.000Z`
`duration`	Integer	Seconds between queueing and merging Example: `480`
`id`	Integer	Example: `7`
`merge_request`	APIEntitiesMergeRequestSimple	—
`merged_at`	String (date-time)	Example: `2026-03-05T09:23:00.000Z`
`pipeline`	APIEntitiesCiPipelineBasic	—
`status`	String	Where the car stands in its queue Allowed values: `idle`, `merged`, `stale`, `fresh`, `merging`, `skip_merged` Example: `merged`
`target_branch`	String	Example: `main`
`updated_at`	String (date-time)	Example: `2026-03-05T09:23:00.000Z`
`user`	APIEntitiesUserBasic	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Metadata

Use this API to retrieve metadata about a GitLab instance, including its version, its revision, whether it is Enterprise Edition, and the configuration of the GitLab agent server for Kubernetes (KAS).

Retrieve metadata information for this GitLab instance

`GET /api/v4/metadata`

Retrieves metadata information for the GitLab instance.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMetadata
`401`	Unauthorized	—

Retrieves version information for the GitLab instance

`GET /api/v4/version`

This feature was introduced in GitLab 8.13 and deprecated in 15.5. We recommend you instead use the Metadata API.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMetadata
`401`	Unauthorized	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMetadata

Property	Type	Description
`enterprise`	Boolean	—
`kas`	Object	—
`kas.enabled`	Boolean	—
`kas.externalK8sProxyUrl`	String	Example: `https://gitlab.example.com:8150/k8s-proxy`
`kas.externalUrl`	String	Example: `grpc://gitlab.example.com:8150`
`kas.version`	String	Example: `15.0.0`
`revision`	String	Example: `c401a659d0c`
`version`	String	Example: `15.2-pre`

Metrics

Use this API to retrieve the definitions of the metrics that GitLab collects for Service Ping.

Download metric definitions

`GET /api/v4/usage_data/metric_definitions`

Downloads all metric definitions as a single YAML file.

Parameters

Name	Type	Description
`include_paths` Query	Boolean	If `true`, includes file paths in the metric definitions Example: `true`

Responses

Code	Description	Schema
`200`	OK	String (`application/yaml`)
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Migrations

Use this API to mark GitLab database migrations as successful, so that a deployment can proceed without re-running them.

These endpoints are available for GitLab Self-Managed and require administrator access to the instance.

List all pending migrations

`GET /api/v4/admin/migrations/pending`

Lists all pending migrations for the instance.

Parameters

Name	Type	Description
`database` Query	String	Database name to query Allowed values: `main` , `ci` , `sec` Default: `main`

Responses

Code	Description	Schema
`200`	200 OK	—
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Update status of a migration

`POST /api/v4/admin/migrations/{timestamp}/mark`

Updates the status of a migration to indicate a successful execution. This prevent them from being executed by the `db:migrate` tasks. Use this API to skip failing migrations after you determine they are safe to skip.

Parameters

Name	Type	Description
`timestamp` Path, required	Integer	Version timestamp of the migration

Request body (`application/json`)

Property	Type	Description
`database`	String	Database name for which the migration is skipped Allowed values: `main` , `ci` , `sec` Default: `main`

Responses

Code	Description	Schema
`201`	201 Created	—
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—
`422`	You can mark only pending migrations	—

Milestones

Use this API to manage milestones for both projects and groups.

List all group milestones

`GET /api/v4/groups/{id}/milestones`

Lists all group milestones for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`state` Query	String	Return “active”, “closed”, or “all” milestones Allowed values: `active`, `closed`, `all` Default: `all`
`iids` Query	Array of integers	Return only the milestones having the given `iid`. Ignored if `include_ancestors` is `true`
`title` Query	String	The title of the milestones
`search` Query	String	The search criteria for the title or description of the milestone
`search_title` Query	String	The search criteria for the title of the milestone
`include_parent_milestones` Query	Boolean	If `true`, includes milestones from all parent groups. Deprecated. Use `include_ancestors` instead. Mutually exclusive with `include_ancestors`
`include_ancestors` Query	Boolean	Include milestones from all parent groups. Mutually exclusive with `include_parent_milestones`
`updated_before` Query	String (date-time)	Return milestones updated before the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`updated_after` Query	String (date-time)	Return milestones updated after the specified datetime.

Name	Type	Description
		Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`containing_date` Query	String (date-time)	Return milestones where start_date <= containing_date <= due_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`start_date` Query	String (date-time)	Return milestones where due_date >= start_date. Requires end_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`end_date` Query	String (date-time)	Return milestones where start_date <= end_date. Requires start_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`sort` Query	String	Return milestones sorted in the given order Allowed values: `due_date_asc`, `due_date_desc`, `start_date_asc`, `start_date_desc`, `name_asc`, `name_desc`, `created_asc`, `created_desc`, `updated_asc`, `updated_desc` Default: `due_date_asc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`include_descendants` Query	Boolean	If `true`, includes milestones for the group and its descendants

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Create a group milestone

`POST /api/v4/groups/{id}/milestones`

Creates a milestone for the specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the milestone
`due_date`	String	Due date of the milestone, in ISO 8601 format (`YYYY-MM-DD`)
`start_date`	String	Start date of the milestone, in ISO 8601 format (`YYYY-MM-DD`)
`title` Required	String	The title of the milestone

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a group milestone

`GET /api/v4/groups/{id}/milestones/{milestone_id}`

Retrieves a specified group milestone.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`milestone_id` Path, required	Integer	ID of the group milestone

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Update a group milestone

`PUT /api/v4/groups/{id}/milestones/{milestone_id}`

Updates the specified group milestone.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`milestone_id` Path, required	Integer	The milestone ID number

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the milestone. At least one of `title` , `description` , `start_date` , `due_date` , `state_event` must be provided
`due_date`	String	Due date of the milestone, in ISO 8601 format (`YYYY-MM-DD`). At least one of `title` , `description` , `start_date` , `due_date` , `state_event` must be provided
`start_date`	String	Start date of the milestone, in ISO 8601 format (`YYYY-MM-DD`). At least one of `title` , `description` , `start_date` , `due_date` , `state_event` must be provided
`state_event`	String	State event of the milestone. At least one of `title` , `description` , `start_date` , `due_date` , `state_event` must be provided Allowed values: `close` , `activate`
`title`	String	The title of the milestone. At least one of `title` , `description` , `start_date` , `due_date` , `state_event` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Delete a group milestone

`DELETE /api/v4/groups/{id}/milestones/{milestone_id}`

Deletes the specified group milestone.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`milestone_id` Path, required	Integer	ID of the group milestone

Responses

Code	Description	Schema
`204`	204 No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all issues assigned to a single milestone

`GET /api/v4/groups/{id}/milestones/{milestone_id}/issues`

Lists all issues assigned to a specified group milestone.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`milestone_id` Path, required	Integer	ID of the group milestone
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssueBasic
`400`	Bad Request	—
`404`	Not Found	—

List all merge requests assigned to a single milestone

`GET /api/v4/groups/{id}/milestones/{milestone_id}/merge_requests`

Lists all merge requests assigned to a specified group milestone.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`milestone_id` Path, required	Integer	ID of the group milestone
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`404`	Not Found	—

List all project milestones

`GET /api/v4/projects/{id}/milestones`

Lists all milestones for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`state` Query	String	Return “active”, “closed”, or “all” milestones Allowed values: `active`, `closed`, `all` Default: `all`
`iids` Query	Array of integers	Return only the milestones having the given `iid`. Ignored if `include_ancestors` is `true`
`title` Query	String	The title of the milestones
`search` Query	String	The search criteria for the title or description of the milestone
`search_title` Query	String	The search criteria for the title of the milestone
`include_parent_milestones` Query	Boolean	If `true`, includes milestones from all parent groups. Deprecated. Use `include_ancestors` instead. Mutually exclusive with `include_ancestors`
`include_ancestors` Query	Boolean	Include milestones from all parent groups. Mutually exclusive with `include_parent_milestones`
`updated_before` Query	String (date-time)	Return milestones updated before the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`updated_after` Query	String (date-time)	Return milestones updated after the specified datetime.

Name	Type	Description
		Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`containing_date` Query	String (date-time)	Return milestones where start_date <= containing_date <= due_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`start_date` Query	String (date-time)	Return milestones where due_date >= start_date. Requires end_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`end_date` Query	String (date-time)	Return milestones where start_date <= end_date. Requires start_date. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`sort` Query	String	Return milestones sorted in the given order Allowed values: `due_date_asc`, `due_date_desc`, `start_date_asc`, `start_date_desc`, `name_asc`, `name_desc`, `created_asc`, `created_desc`, `updated_asc`, `updated_desc` Default: `due_date_asc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Create a project milestone

`POST /api/v4/projects/{id}/milestones`

Creates a project milestone.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the milestone
`due_date`	String	Due date of the milestone, in ISO 8601 format (`YYYY-MM-DD`)
`start_date`	String	Start date of the milestone, in ISO 8601 format (`YYYY-MM-DD`)
`title` Required	String	The title of the milestone

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a project milestone

`GET /api/v4/projects/{id}/milestones/{milestone_id}`

Retrieves a specified project milestone.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	ID of the project milestone

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Update a project milestone

`PUT /api/v4/projects/{id}/milestones/{milestone_id}`

Updates a specified project milestone.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	The milestone ID number

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the milestone. At least one of `title`, `description`, `start_date`, `due_date`, `state_event` must be provided
`due_date`	String	Due date of the milestone, in ISO 8601 format (`YYYY-MM-DD`). At least one of `title`, `description`, `start_date`, `due_date`, `state_event` must be provided
`start_date`	String	Start date of the milestone, in ISO 8601 format (`YYYY-MM-DD`). At least one of `title`, `description`, `start_date`, `due_date`, `state_event` must be provided
`state_event`	String	State event of the milestone. At least one of `title`, `description`, `start_date`, `due_date`, `state_event` must be provided Allowed values: `close`, `activate`
`title`	String	The title of the milestone. At least one of `title`, `description`, `start_date`, `due_date`, `state_event` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMilestone
`400`	Bad Request	—
`404`	Not Found	—

Delete a project milestone

`DELETE /api/v4/projects/{id}/milestones/{milestone_id}`

Deletes a specified project milestone. Only for users with the Planner, Reporter, Developer, Maintainer, or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	ID of the project milestone

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all issues for a project milestone

`GET /api/v4/projects/{id}/milestones/{milestone_id}/issues`

Lists all issues assigned to a specified project milestone.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	ID of the project milestone
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssueBasic
`400`	Bad Request	—
`404`	Not Found	—

List all merge requests for a project milestone

`GET /api/v4/projects/{id}/milestones/{milestone_id}/merge_requests`

Lists all merge requests assigned to a specified project milestone.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	ID of the project milestone
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`404`	Not Found	—

Promote a project milestone to a group milestone

`POST /api/v4/projects/{id}/milestones/{milestone_id}/promote`

Promotes a specified project milestone to a group milestone. Only for users with the Planner, Reporter, Developer, Maintainer, or Owner role for the group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`milestone_id` Path, required	Integer	ID of the project milestone

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesIssueBasic

Property	Type	Description
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`project_id`	Integer (int64)	Example: `4`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—

Property	Type	Description
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Mobile push subscriptions

Operations related to mobile push subscriptions.

Register a mobile device for push notifications

`POST /api/v4/user/push_subscriptions`

Idempotently registers the calling user’s device token for APNs push notifications. Re-registering an existing token refreshes its attributes; a token previously owned by another user is reassigned to the calling user.

Request body (`application/json`)

Property	Type	Description
`apns_environment`	String	The APNs environment the token was issued for Allowed values: `production`, `sandbox` Default: `production`
`app_version`	String	The installed application version
`bundle_id`	String	The application bundle identifier
`device_name`	String	A human-readable device name
`device_token` Required	String	The hexadecimal APNs device token
`locale`	String	The device locale
`payload_mode`	String	Whether notifications carry the full content or only identifiers Allowed values: `full`, `id_only`
`platform`	String	The device platform Allowed values: `ios`, `macos`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMobilePushSubscription
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Unregister a mobile device from push notifications

`DELETE /api/v4/user/push_subscriptions`

Removes the calling user’s push subscriptions for the given device token.

Parameters

Name	Type	Description
`device_token` Query, required	String	The hexadecimal APNs device token

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMobilePushSubscription

Property	Type	Description
`created_at`	String (date-time)	Example: `2026-07-30T12:00:00.000Z`
`id`	Integer (int64)	Example: `1`

Namespaces

Use this API to interact with namespaces, a special resource category used to organize users and groups. For more information, see namespaces.

This API uses pagination to filter results.

List all namespaces

`GET /api/v4/namespaces`

Lists all namespaces available to the current user. If the user is an administrator, this endpoint returns all namespaces in the instance.

Parameters

Name	Type	Description
`search` Query	String	Return only namespaces that contain the specified value in their name or path
`owned_only` Query	Boolean	If `true`, returns only namespaces owned by the current user
`top_level_only` Query	Boolean	If `true`, returns only top-level namespaces Default: `false`
`full_path_search` Query	Boolean	If `true`, the `search` parameter is matched against the full path of the namespaces Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesNamespace
`400`	Bad Request	—
`401`	Unauthorized	—

Retrieve namespace details

`GET /api/v4/namespaces/{id}`

Retrieves a specified namespace.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the namespace

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNamespace
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Verify namespace availability

`GET /api/v4/namespaces/{id}/exists`

Verifies that a namespace is available for use.

Parameters

Name	Type	Description
`id` Path, required	String	Path of the namespace
`parent_id` Query	Integer	ID of the parent namespace. If omitted, only top-level namespaces are considered

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNamespace Existence
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesNamespace

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`members_count_with_descendants`	Integer	Example: `5`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`projects_count`	Integer	Example: `123`
`root_repository_size`	Integer	Example: `123`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesNamespaceExistence

Property	Type	Description
`exists`	Boolean	—
`suggests`	Array of strings	Example: `["my-group1"]`

Notes

Use this API to manage the comments and system records attached to GitLab content. You can:

- Create and modify comments on issues, merge requests, epics, snippets, commits, and wiki pages.
- Retrieve system-generated notes about object changes.
- Control visibility with confidential and internal flags.

Some system-generated notes are tracked as separate resource events instead, including label, state, milestone, weight, and iteration events.

By default, `GET` requests return 20 results at a time, because the results are paginated.

List all issue notes

`GET /api/v4/projects/{id}/issues/{noteable_id}/notes`

Lists all notes for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`activity_filter` Query	String	Filter notes by activity type Allowed values: `all_notes`, `only_comments`, `only_activity` Default: `all_notes`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Create an issue note

`POST /api/v4/projects/{id}/issues/{noteable_id}/notes`

Creates a note for an issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true`, marks the note as internal. Deprecated. Use `internal` instead
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z`. Requires administrator or project/group owner rights
`internal`	Boolean	Internal note flag Default: `false`
`merge_request_diff_head_sha`	String	SHA of the head commit, which ensures the merge request wasn't updated after the API request was sent. Required for the `/merge` quick action

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an issue note

`GET /api/v4/projects/{id}/issues/{noteable_id}/notes/{note_id}`

Retrieves a specified note for an issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`note_id` Path, required	Integer	ID of the note
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update an issue note

`PUT /api/v4/projects/{id}/issues/{noteable_id}/notes/{note_id}`

Updates an existing note for an issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true`, the note must already be confidential. If `false`, it must not be. Confidentiality cannot be changed after a note is created. Deprecated

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete an issue note

`DELETE /api/v4/projects/{id}/issues/{noteable_id}/notes/{note_id}`

Deletes a specified note from an issue.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all merge request notes

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/notes`

Lists all notes for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`activity_filter` Query	String	Filter notes by activity type Allowed values: `all_notes`, `only_comments`, `only_activity` Default: `all_notes`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Create a merge request note

`POST /api/v4/projects/{id}/merge_requests/{noteable_id}/notes`

Creates a note for a merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true` , marks the note as internal. Deprecated. Use `internal` instead
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights
`internal`	Boolean	Internal note flag Default: `false`
`merge_request_diff_head_sha`	String	SHA of the head commit, which ensures the merge request wasn't updated after the API request was sent. Required for the `/merge` quick action

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge request note

`GET /api/v4/projects/{id}/merge_requests/{noteable_id}/notes/{note_id}`

Retrieves a specified note for a merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`note_id` Path, required	Integer	ID of the note
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a merge request note

`PUT /api/v4/projects/{id}/merge_requests/{noteable_id}/notes/{note_id}`

Updates an existing note for a merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true` , the note must already be confidential. If `false` , it must not be. Confidentiality cannot be changed after a note is created. Deprecated

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a merge request note

`DELETE /api/v4/projects/{id}/merge_requests/{noteable_id}/notes/{note_id}`

Deletes a specified note from a merge request.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all snippet notes

`GET /api/v4/projects/{id}/snippets/{noteable_id}/notes`

Lists all notes for a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`activity_filter` Query	String	Filter notes by activity type Allowed values: `all_notes`, `only_comments`, `only_activity` Default: `all_notes`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Create a snippet note

`POST /api/v4/projects/{id}/snippets/{noteable_id}/notes`

Creates a note for a snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true` , marks the note as internal. Deprecated. Use `internal` instead
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights
`internal`	Boolean	Internal note flag Default: `false`
`merge_request_diff_head_sha`	String	SHA of the head commit, which ensures the merge request wasn't updated after the API request was sent. Required for the `/merge` quick action

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a snippet note

`GET /api/v4/projects/{id}/snippets/{noteable_id}/notes/{note_id}`

Retrieves a specified note for a snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`note_id` Path, required	Integer	ID of the note
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a snippet note

`PUT /api/v4/projects/{id}/snippets/{noteable_id}/notes/{note_id}`

Updates an existing note for a snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true`, the note must already be confidential. If `false`, it must not be. Confidentiality cannot be changed after a note is created. Deprecated

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a snippet note

`DELETE /api/v4/projects/{id}/snippets/{noteable_id}/notes/{note_id}`

Deletes a specified note from a snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

List all wiki page meta notes

`GET /api/v4/projects/{id}/wiki_pages/{noteable_id}/notes`

Lists all notes for a specified wiki page meta.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`activity_filter` Query	String	Filter notes by activity type Allowed values: `all_notes`, `only_comments`, `only_activity` Default: `all_notes`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Create a wiki page meta note

`POST /api/v4/projects/{id}/wiki_pages/{noteable_id}/notes`

Creates a note for a wiki page meta.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Request body (`application/json`)

Property	Type	Description
`body` Required	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true` , marks the note as internal. Deprecated. Use `internal` instead
`created_at`	String	Date and time the note was created, in ISO 8601 format such as `2016-03-11T03:45:40Z` . Requires administrator or project/group owner rights
`internal`	Boolean	Internal note flag Default: `false`
`merge_request_diff_head_sha`	String	SHA of the head commit, which ensures the merge request wasn't updated after the API request was sent. Required for the `/merge` quick action

Responses

Code	Description	Schema
`201`	Created	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a wiki page meta note

`GET /api/v4/projects/{id}/wiki_pages/{noteable_id}/notes/{note_id}`

Retrieves a specified note for a wiki page meta.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`note_id` Path, required	Integer	ID of the note
`noteable_id` Path, required	Integer	ID of the resource the note is attached to

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Update a wiki page meta note

`PUT /api/v4/projects/{id}/wiki_pages/{noteable_id}/notes/{note_id}`

Updates an existing note for a wiki page meta.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Request body (`application/json`)

Property	Type	Description
`body`	String	Content of the note. Limited to 1,000,000 characters
`confidential`	Boolean	If `true` , the note must already be confidential. If `false` , it must not be. Confidentiality cannot be changed after a note is created. Deprecated

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Delete a wiki page meta note

`DELETE /api/v4/projects/{id}/wiki_pages/{noteable_id}/notes/{note_id}`

Deletes a specified note from a wiki page meta.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`noteable_id` Path, required	Integer	ID of the resource the note is attached to
`note_id` Path, required	Integer	ID of the note

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNote
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesNote

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`body`	String	Example: `Note body.`
`commands_changes`	Object	—
`commit_id`	String	Example: `7b09ce7e6f80347baf0316c8c94cdba9a0a7e91d`
`confidential`	Boolean	Example: `false`
`created_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`internal`	Boolean	Example: `false`
`noteable_id`	Integer (int64)	Example: `1`
`noteable_iid`	Integer	Example: `1`
`noteable_type`	String	Example: `Issue`
`position`	Object	—
`project_id`	Integer (int64)	Example: `1`
`resolvable`	Boolean	Example: `false`
`resolved`	Boolean	Example: `false`
`resolved_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`
`resolved_by`	APIEntitiesUserBasic	—
`suggestions`	APIEntitiesSuggestion	—
`system`	Boolean	Example: `false`
`type`	String	Example: `DiscussionNote`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2022-01-31T15:10:44.988Z`

APIEntitiesSuggestion

Property	Type	Description
`appliable`	Boolean	Example: `true`
`applied`	Boolean	Example: `false`
`from_content`	String	Example: `Original content`
`from_line`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`to_content`	String	Example: `New content`
`to_line`	Integer	Example: `1`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Notification settings

Use this API to retrieve and update notification settings globally for your account, or for an individual group or project, which overrides the global setting.

Retrieve notification settings for a group

`GET /api/v4/groups/{id}/notification_settings`

Retrieves the notification level for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNotificationSetting
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Update notification settings for a group

`PUT /api/v4/groups/{id}/notification_settings`

Updates the notification settings for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`change_reviewer_merge_request`	Boolean	Enable/disable this notification
`close_issue`	Boolean	Enable/disable this notification
`close_merge_request`	Boolean	Enable/disable this notification
`failed_pipeline`	Boolean	Enable/disable this notification
`fixed_pipeline`	Boolean	Enable/disable this notification
`issue_due`	Boolean	Enable/disable this notification
`level`	String	The group notification level
`merge_merge_request`	Boolean	Enable/disable this notification
`merge_when_pipeline_succeeds`	Boolean	Enable/disable this notification
`moved_project`	Boolean	Enable/disable this notification
`new_issue`	Boolean	Enable/disable this notification
`new_merge_request`	Boolean	Enable/disable this notification
`new_note`	Boolean	Enable/disable this notification
`new_release`	Boolean	Enable/disable this notification
`push_to_merge_request`	Boolean	Enable/disable this notification
`reassign_issue`	Boolean	Enable/disable this notification
`reassign_merge_request`	Boolean	Enable/disable this notification
`reopen_issue`	Boolean	Enable/disable this notification
`reopen_merge_request`	Boolean	Enable/disable this notification
`success_pipeline`	Boolean	Enable/disable this notification

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNotificationSetting
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve global notification settings

`GET /api/v4/notification_settings`

Retrieves the global notification level and email address.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGlobalNotificationSetting
`401`	Unauthorized	—

Update global notification settings

`PUT /api/v4/notification_settings`

Updates notification settings and email address.

Request body (`application/json`)

Property	Type	Description
`change_reviewer_merge_request`	Boolean	Enable/disable this notification
`close_issue`	Boolean	Enable/disable this notification
`close_merge_request`	Boolean	Enable/disable this notification
`failed_pipeline`	Boolean	Enable/disable this notification
`fixed_pipeline`	Boolean	Enable/disable this notification
`issue_due`	Boolean	Enable/disable this notification
`level`	String	Global notification level
`merge_merge_request`	Boolean	Enable/disable this notification
`merge_when_pipeline_succeeds`	Boolean	Enable/disable this notification
`moved_project`	Boolean	Enable/disable this notification
`new_issue`	Boolean	Enable/disable this notification
`new_merge_request`	Boolean	Enable/disable this notification
`new_note`	Boolean	Enable/disable this notification
`new_release`	Boolean	Enable/disable this notification
`notification_email`	String	Email address where notifications are sent
`push_to_merge_request`	Boolean	Enable/disable this notification
`reassign_issue`	Boolean	Enable/disable this notification
`reassign_merge_request`	Boolean	Enable/disable this notification
`reopen_issue`	Boolean	Enable/disable this notification
`reopen_merge_request`	Boolean	Enable/disable this notification
`success_pipeline`	Boolean	Enable/disable this notification

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGlobalNotificationSetting
`400`	Bad request	—
`401`	Unauthorized	—

Retrieve notification settings for a project

`GET /api/v4/projects/{id}/notification_settings`

Retrieves the notification level for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNotificationSetting
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Update notification settings for a project

`PUT /api/v4/projects/{id}/notification_settings`

Updates the notification settings for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`change_reviewer_merge_request`	Boolean	Enable/disable this notification
`close_issue`	Boolean	Enable/disable this notification
`close_merge_request`	Boolean	Enable/disable this notification
`failed_pipeline`	Boolean	Enable/disable this notification
`fixed_pipeline`	Boolean	Enable/disable this notification
`issue_due`	Boolean	Enable/disable this notification
`level`	String	The project notification level
`merge_merge_request`	Boolean	Enable/disable this notification
`merge_when_pipeline_succeeds`	Boolean	Enable/disable this notification
`moved_project`	Boolean	Enable/disable this notification
`new_issue`	Boolean	Enable/disable this notification
`new_merge_request`	Boolean	Enable/disable this notification
`new_note`	Boolean	Enable/disable this notification
`new_release`	Boolean	Enable/disable this notification
`push_to_merge_request`	Boolean	Enable/disable this notification
`reassign_issue`	Boolean	Enable/disable this notification
`reassign_merge_request`	Boolean	Enable/disable this notification
`reopen_issue`	Boolean	Enable/disable this notification
`reopen_merge_request`	Boolean	Enable/disable this notification
`success_pipeline`	Boolean	Enable/disable this notification

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNotificationSetting
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesGlobalNotificationSetting

Property	Type	Description
`events`	String	—
`level`	String	—
`notification_email`	String	—

APIEntitiesNotificationSetting

Property	Type	Description
`events`	String	—
`level`	String	—

Offline transfers

Use this API to start and track offline transfer exports and imports, which migrate data between GitLab instances that cannot reach each other over a network.

List all offline transfer exports

`GET /api/v4/offline_exports`

Lists all offline transfer exports. For more information, see https://docs.gitlab.com/user/group/import/offline_transfer_migrations/

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`sort` Query	String	Return offline transfer exports sorted in created by `asc` or `desc` order Allowed values: `asc`, `desc` Default: `desc`
`status` Query	String	Return offline transfer exports with specified status Allowed values: `created`, `started`, `finished`, `failed`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—

Start a new offline transfer export

`POST /api/v4/offline_exports`

Initiates a new offline transfer export. For more information, see https://docs.gitlab.com/user/group/import/offline_transfer_migrations/

Request body (`application/json`)

Property	Type	Description
`aws_s3_configuration`	Object	AWS S3 object storage configuration. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`aws_s3_configuration.aws_access_key_id` Required	String	AWS S3 access key ID
`aws_s3_configuration.aws_secret_access_key` Required	String	AWS S3 secret access key
`aws_s3_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `false`
`aws_s3_configuration.region` Required	String	AWS S3 object storage region
`bucket` Required	String	Name of the object storage bucket where export data is stored
`entities` Required	Array of objects	List of entities to export
`entities[].full_path` Required	String	Relative path of the entity to export Example: `'source/full/path' not 'https://example.com/source/full/path'`
`gcs_adc_configuration`	Object	Google Cloud Storage configuration using Application Default Credentials. Available on GitLab Self-Managed or Git

Property	Type	Description
		Lab Dedicated, and must be enabled by an administrator. Only administrators can use these credentials, and the bucket name must start with “gitlab-offline-transfer-”. For more information, see https://docs.gitlab.com/administration/settings/import_and_export_settings/#allow-application-default-credentials-for-offline-transfer. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_adc_configuration.google_project` Required	String	Google Cloud project ID
`gcs_configuration`	Object	Google Cloud Storage configuration using a service account JSON key. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_configuration.google_json_key_string` Required	String	Google Cloud service account JSON key contents
`gcs_configuration.google_project` Required	String	Google Cloud project ID
`gcs_hmac_configuration`	Object	

Property	Type	Description
		Google Cloud Storage configuration using S3-interoperability HMAC keys. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_hmac_configuration.google_storage_access_key_id` Required	String	GCS HMAC access key ID
`gcs_hmac_configuration.google_storage_secret_access_key` Required	String	GCS HMAC secret
`gcs_hmac_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `true`
`gcs_hmac_configuration.region` Required	String	GCS bucket region
`s3_compatible_configuration`	Object	MinIO or other S3-compatible object storage configuration. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`s3_compatible_configuration.aws_access_key_id` Required	String	S3-compatible access key ID
	String	

Property	Type	Description
`s3_compatible_configuration.aws_secret_access_key` Required		S3-compatible secret access key
`s3_compatible_configuration.endpoint` Required	String	Object storage location endpoint
`s3_compatible_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `true`
`s3_compatible_configuration.region` Required	String	S3-compatible object storage region

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Get offline transfer export details

`GET /api/v4/offline_exports/{id}`

Retrieves details of an offline transfer export. For more information, see https://docs.gitlab.com/user/group/import/offline_transfer_migrations/

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of user’s offline transfer export

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Start a new offline transfer import

`POST /api/v4/offline_imports`

Initiates a new offline transfer import from object storage. For more information, see https://docs.gitlab.com/user/group/import/offline_transfer_migrations/

Request body (`application/json`)

Property	Type	Description
`aws_s3_configuration`	Object	AWS S3 object storage configuration. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`aws_s3_configuration.aws_access_key_id` Required	String	AWS S3 access key ID
`aws_s3_configuration.aws_secret_access_key` Required	String	AWS S3 secret access key
`aws_s3_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `false`
`aws_s3_configuration.region` Required	String	AWS S3 object storage region
`bucket` Required	String	Name of the object storage bucket where export data is stored
`entities`	Array of objects	List of entities to import. Exactly one of `entities`, `import_all` must be provided
`entities[].destination_namespace` Required	String	Full path of the destination namespace
`entities[].destination_slug`	String	Destination slug for the imported entity
`entities[].source_full_path` Required	String	Full path of the entity on the source instance

Property	Type	Description
`entities[].source_type` Required	String	Type of the entity to import Allowed values: `group_entity`, `project_entity` Minimum length: `1`
`export_prefix` Required	String	Prefix of the export within the bucket
`gcs_adc_configuration`	Object	Google Cloud Storage configuration using Application Default Credentials. Available on GitLab Self-Managed or Git Lab Dedicated, and must be enabled by an administrator. Only administrators can use these credentials, and the bucket name must start with “gitlab-offline-transfer-”. For more information, see https://docs.gitlab.com/administration/settings/import_and_export_settings/#allow-application-default-credentials-for-offline-transfer. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_adc_configuration.google_project` Required	String	Google Cloud project ID
`gcs_configuration`	Object	Google Cloud Storage configuration using a service account JSON key. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_`

Property	Type	Description
		`hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_configuration.google_json_key_string` Required	String	Google Cloud service account JSON key contents
`gcs_configuration.google_project` Required	String	Google Cloud project ID
`gcs_hmac_configuration`	Object	Google Cloud Storage configuration using S3-interoperability HMAC keys. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`gcs_hmac_configuration.google_storage_access_key_id` Required	String	GCS HMAC access key ID
`gcs_hmac_configuration.google_storage_secret_access_key` Required	String	GCS HMAC secret
`gcs_hmac_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `true`
`gcs_hmac_configuration.region` Required	String	GCS bucket region
`import_all`	Object	Import every top-level group in the export, recreating the source instance structure.

Property	Type	Description
		Exactly one of `entities`, `import_all` must be provided
`import_all.destination_namespace` Required	String	Full path of the destination namespace the source structure is recreated under. Use an empty string to recreate it at the instance root Example: `destination-group/subgroup`
`s3_compatible_configuration`	Object	MinIO or other S3-compatible object storage configuration. Exactly one of `aws_s3_configuration`, `s3_compatible_configuration`, `gcs_configuration`, `gcs_hmac_configuration`, `gcs_adc_configuration` must be provided
`s3_compatible_configuration.aws_access_key_id` Required	String	S3-compatible access key ID
`s3_compatible_configuration.aws_secret_access_key` Required	String	S3-compatible secret access key
`s3_compatible_configuration.endpoint` Required	String	Object storage location endpoint
`s3_compatible_configuration.path_style`	Boolean	Use path-style URLs instead of virtual-hosted-style URLs Default: `true`
`s3_compatible_configuration.region` Required	String	S3-compatible object storage region

Responses

Code	Description	Schema
`201`	Created	APIEntitiesBulkImport
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—
`429`	Too many requests	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBulkImport

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`has_failures`	Boolean	Example: `false`
`id`	Integer (int64)	Example: `1`
`source_type`	String	Example: `gitlab`
`source_url`	String	Example: `https://source.gitlab.com/`
`status`	String	Allowed values: `created`, `started`, `finished`, `timeout`, `failed` Example: `finished`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

Organizations

Use this API to interact with GitLab organizations. For more information, see organizations.

Cancel maintenance on an organization (maintenance_initialization -> active)

`POST /api/v4/internal/org_mover/cancel_maintenance`

Aborts maintenance initialization before the organization reaches the maintenance state, returning it to active.

Request body (`application/json`)

Property	Type	Description
`organization_id` Required	Integer	The ID of the organization

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Confirm maintenance on an organization (maintenance_initialization -> maintenance)

`POST /api/v4/internal/org_mover/confirm_maintenance`

Completes the maintenance transition once the organization is drained.

Request body (`application/json`)

Property	Type	Description
`organization_id` Required	Integer	The ID of the organization

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`409`	Conflict	—
`422`	Unprocessable entity	—

Exit maintenance on an organization (maintenance -> active)

`POST /api/v4/internal/org_mover/exit_maintenance`

Returns an organization from maintenance to active.

Request body (`application/json`)

Property	Type	Description
`organization_id` Required	Integer	The ID of the organization

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Report cutover readiness for an organization

`GET /api/v4/internal/org_mover/maintenance_readiness`

Reports whether an organization is drained and safe to enter maintenance, and what is blocking it when not.

Parameters

Name	Type	Description
`organization_id` Query, required	Integer	The ID of the organization

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Report an organization’s current lifecycle state

`GET /api/v4/internal/org_mover/maintenance_state`

Returns the organization’s current maintenance lifecycle state.

Parameters

Name	Type	Description
`organization_id` Query, required	Integer	The ID of the organization

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Start maintenance on an organization (active -> maintenance_initialization)

`POST /api/v4/internal/org_mover/start_maintenance`

Begins the maintenance transition for an organization.

Request body (`application/json`)

Property	Type	Description
`maintenance_reason` Required	String	The reason for entering maintenance Allowed values: `migration`, `isolation`, `incident`, `billing`, `legal` Minimum length: `1`
`organization_id` Required	Integer	The ID of the organization

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—
`422`	Unprocessable entity	—

Create an organization

`POST /api/v4/organizations`

Creates an organization. This feature was introduced in GitLab 17.5. This feature is behind the `allow_organization_creation` feature flag. In GitLab 18.3, the feature flag changed to `organization_switching`. In GitLab 19.4, the feature flag changed to `org_stage_experimental`.

Request body (`multipart/form-data`)

Property	Type	Description
`avatar`	String (binary)	The avatar image for the organization
`description`	String	The description of the organization
`name` Required	String	The name of the organization
`path` Required	String	The path of the organization
`visibility`	String	The visibility level of the organization Allowed values: `private`, `public`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesOrganizationsOrganization
`400`	Bad Request	—

Soft-delete an organization

`DELETE /api/v4/organizations/{id}`

This feature was introduced in GitLab 19.2.

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the organization

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesOrganizationsOrganization

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/uploads/-/system/organizations/organization_detail/avatar/1/avatar.png`
`created_at`	String (date-time)	Example: `2022-02-24T20:22:30.097Z`
`description`	String	Example: `My description`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `GitLab`
`path`	String	Example: `gitlab`
`updated_at`	String (date-time)	Example: `2022-02-24T20:22:30.097Z`
`uuid`	String	Example: `0192f8c2-1a2b-7cde-89ab-0123456789ab`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://example.com/o/gitlab/-/overview`

Packages

Use this API to manage the packages stored in a project or a group.

Download the maven package file for a group

`GET /api/v4/groups/{id}/-/packages/maven/{path}/{file_name}`

This feature was introduced in GitLab 11.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group
`file_name` Path, required	String	Package file name Example: `mypkg-1.0-SNAPSHOT.jar`
`path` Path, required	String	Package path Example: `foo/bar/mypkg/1.0-SNAPSHOT`

Responses

Code	Description	Schema
`200`	OK	—
`302`	Found	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all packages for a group

`GET /api/v4/groups/{id}/packages`

Lists all packages for a specified group. When accessed without authentication, only packages of public projects are returned. By default, packages with `default`, `deprecated`, and `error` status are returned. Use the `status` parameter to view other packages.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`exclude_subgroups` Query	Boolean	If `true`, excludes packages from subgroup projects Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `name`, `version`, `type`, `project_path` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`package_type` Query	String	Filter packages by type Allowed values: `maven`, `npm`, `conan`, `nuget`, `pypi`, `composer`, `generic`, `golang`, `debian`, `rubygems`, `helm`, `terraform_module`, `rpm`, `ml_model`, `cargo`
`package_name` Query	String	Filter packages by name, using a fuzzy search
`package_version` Query	String	Filter packages by version. When used together with `include_versionless`,

Name	Type	Description
		versionless packages are not returned
`include_versionless` Query	Boolean	If `true`, includes versionless packages in the response
`status` Query	String	Filter packages by status Allowed values: `default`, `hidden`, `processing`, `error`, `pending_destruction`, `deprecated`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPackage
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Group Not Found	—

Download the maven package file for the instance

`GET /api/v4/packages/maven/{path}/{file_name}`

This feature was introduced in GitLab 11.6

Parameters

Name	Type	Description
`file_name` Path, required	String	Package file name Example: `mypkg-1.0-SNAPSHOT.jar`
`path` Path, required	String	Package path Example: `foo/bar/mypkg/1.0-SNAPSHOT`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all packages for a project

`GET /api/v4/projects/{id}/packages`

Lists all packages for a specified project. All package types are included in results. Unauthenticated requests return only packages of public projects. By default, packages with `default`, `deprecated`, and `error` status are returned. Use the `status` parameter to view other packages.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `name`, `version`, `type` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`
`package_type` Query	String	Filter packages by type Allowed values: `maven`, `npm`, `conan`, `nuget`, `pypi`, `composer`, `generic`, `golang`, `debian`, `rubygems`, `helm`, `terraform_module`, `rpm`, `ml_model`, `cargo`
`package_name` Query	String	Filter packages by name, using a fuzzy search
`package_version` Query	String	Filter packages by version. When used together with `include_versionless`, versionless packages are not returned
`include_versionless` Query	Boolean	If `true`, includes versionless packages in the response

Name	Type	Description
`status` Query	String	Filter packages by status Allowed values: `default`, `hidden`, `processing`, `error`, `pending_destruction`, `deprecated`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPackage
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Project Not Found	—

Download package file

`GET /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{file_name}`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy` , `direct`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload package file

`PUT /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{file_name}`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to publish (generated by Multipart middleware)
`select`	String	Response format selector. If set to “package_file”, returns the created package file object in the response Allowed values: `package_file`
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize generic package file

`PUT /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{file_name}/authorize`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`

Request body (`application/json`)

Property	Type	Description
`status`	String	Package status Allowed values: `default`, `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download package file

`GET /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{path}/{file_name}`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`
`path` Path, required	String	File directory path
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy` , `direct`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload package file

`PUT /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{path}/{file_name}`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`
`path` Path, required	String	File directory path

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to publish (generated by Multipart middleware)
`select`	String	Response format selector. If set to “package_file”, returns the created package file object in the response Allowed values: `package_file`
`status`	String	Package status Allowed values: `default` , `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize generic package file

`PUT /api/v4/projects/{id}/packages/generic/{package_name}/{package_version}/{path}/{file_name}/authorize`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Pattern: `^[A-Za-z0-9._\-+]+$`
`file_name` Path, required	String	Package file name Pattern: `^(?!~)(?!@)[A-Za-z0-9._\-+~@]+(?<!~)(?<!@)$`
`package_version` Path, required	String	Package version Pattern: `^(?!(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.\.)[\w+.\-]+$`
`path` Path, required	—	—

Request body (`application/json`)

Property	Type	Description
`status`	String	Package status Allowed values: `default`, `hidden`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List

`GET /api/v4/projects/{id}/packages/go/{module_name}/@v/list`

Get all tagged versions for a given Go module.See `go help goproxy`, GET $GOPROXY/<module>/@v/list. This feature was introduced in GitLab 13.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The project ID or full path of a project
`module_name` Path, required	String	The name of the Go module

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Version metadata

`GET /api/v4/projects/{id}/packages/go/{module_name}/@v/{module_version}.info`

Get all tagged versions for a given Go module.See `go help goproxy`, GET $GOPROXY/<module>/@v/<version>.info. This feature was introduced in GitLab 13.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The project ID or full path of a project
`module_name` Path, required	String	The name of the Go module
`module_version` Path, required	String	The version of the Go module

Responses

Code	Description	Schema
`200`	OK	APIEntitiesGoModuleVersion
`400`	Bad Request	—
`404`	Not Found	—

Download module file

`GET /api/v4/projects/{id}/packages/go/{module_name}/@v/{module_version}.mod`

Get the module file of a given module version.See `go help goproxy`, GET $GOPROXY/<module>/@v/<version>.mod. This feature was introduced in GitLab 13.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The project ID or full path of a project
`module_name` Path, required	String	The name of the Go module
`module_version` Path, required	String	The version of the Go module

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Download module source

`GET /api/v4/projects/{id}/packages/go/{module_name}/@v/{module_version}.zip`

Get a zip of the source of the given module version.See `go help goproxy`, GET $GOPROXY/<module>/@v/<version>.zip. This feature was introduced in GitLab 13.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The project ID or full path of a project
`module_name` Path, required	String	The name of the Go module
`module_version` Path, required	String	The version of the Go module

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Download the maven package file for a project

`GET /api/v4/projects/{id}/packages/maven/{path}/{file_name}`

This feature was introduced in GitLab 11.3

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	Package file name Example: `mypkg-1.0-SNAPSHOT.jar`
`path` Path, required	String	Package path Example: `foo/bar/mypkg/1.0-SNAPSHOT`

Responses

Code	Description	Schema
`200`	OK	—
`302`	Found	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload the maven package file

`PUT /api/v4/projects/{id}/packages/maven/{path}/{file_name}`

This feature was introduced in GitLab 11.3

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`path` Path, required	String	Package path Example: `foo/bar/mypkg/1.0-SNAPSHOT`
`file_name` Path, required	String	Package file name Pattern: `^[A-Za-z0-9._\-+]+$` Example: `mypkg-1.0-SNAPSHOT.pom`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Workhorse authorize the maven package file upload

`PUT /api/v4/projects/{id}/packages/maven/{path}/{file_name}/authorize`

This feature was introduced in GitLab 11.3

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`path` Path, required	String	Package path Example: `foo/bar/mypkg/1.0-SNAPSHOT`
`file_name` Path, required	String	Package file name Pattern: `^[A-Za-z0-9._\-+]+$` Example: `mypkg-1.0-SNAPSHOT.pom`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a project package

`GET /api/v4/projects/{id}/packages/{package_id}`

Retrieves a specified project package. Only packages with status `default` or `deprecated` are returned.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_id` Path, required	Integer	ID of the package

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackage
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a project package

`DELETE /api/v4/projects/{id}/packages/{package_id}`

Deletes a specified project package.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_id` Path, required	Integer	ID of the package

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all package files

`GET /api/v4/projects/{id}/packages/{package_id}/package_files`

Lists all package files for a specified package.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_id` Path, required	Integer	ID of the package
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `created_at`, `file_name` Default: `id`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPackageFile
`400`	Bad Request	—
`404`	Not Found	—

Delete a package file

`DELETE /api/v4/projects/{id}/packages/{package_id}/package_files/{package_file_id}`

Deletes a specified package file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_id` Path, required	Integer	ID of the package
`package_file_id` Path, required	Integer	ID of the package file

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Download a package file

`GET /api/v4/projects/{id}/packages/{package_id}/package_files/{package_file_id}/download`

This feature was introduced in GitLab 18.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_id` Path, required	Integer	ID of the package
`package_file_id` Path, required	Integer	ID of the package file
`download_mode` Query	String	Requested download transfer mode (`proxy` or `direct`). Only honored when allowed by the object storage configuration Allowed values: `proxy` , `direct`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all package pipelines

`GET /api/v4/projects/{id}/packages/{package_id}/pipelines`

Lists all pipelines for a specified package. The results are sorted by `id` in descending order. The results are paginated and return up to 20 records per page. This feature was introduced in GitLab 16.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items per page Default: `20` Maximum: `20` Minimum: `1` Example: `20`
`package_id` Path, required	Integer	ID of the package
`cursor` Query	String	Cursor for obtaining the next set of records

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagePipeline
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesGoModuleVersion

Property	Type	Description
`Time`	String	Example: `1617822312 -0600`
`Version`	String	Example: `v1.0.0`

APIEntitiesPackage

Property	Type	Description
`_links`	Object	—
`_links.delete_api_path`	String	—
`_links.web_path`	String	—
`conan_package_name`	String	—
`created_at`	String (date-time)	Example: `2022-09-16T12:47:31.949Z`
`creator_id`	Integer (int64)	ID of the user who created the package Example: `1`
`id`	Integer (int64)	Example: `1`
`last_downloaded_at`	String (date-time)	Example: `2022-09-19T11:32:35.169Z`
`name`	String	Example: `@foo/bar`
`package_type`	String	Example: `npm`
`pipeline`	APIEntitiesPackagePipeline	—
`pipelines`	APIEntitiesPackagePipeline	—
`project_id`	Integer (int64)	Example: `2`
`project_path`	String	Example: `gitlab/foo/bar`
`status`	String	Example: `default`
`tags`	String	—
`version`	String	Example: `1.0.3`
`versions`	APIEntitiesPackageVersion	—

APIEntitiesPackageFile

Property	Type	Description
`created_at`	String (date-time)	Example: `2018-11-07T15:25:52.199Z`
`file_md5`	String	Example: `58e6a45a629910c6ff99145a688971ac`
`file_name`	String	Example: `my-app-1.5-20181107.152550-1.jar`
`file_sha1`	String	Example: `ebd193463d3915d7e22219f52740056dfd26cbfe`
`file_sha256`	String	Example: `a903393463d3915d7e22219f52740056dfd26cbfeff321b`
`id`	Integer (int64)	Example: `225`
`package_id`	Integer (int64)	Example: `4`
`pipelines`	APIEntitiesPackagePipeline	—
`size`	Integer	Example: `2421`

APIEntitiesPackagePipeline

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesPackageVersion

Property	Type	Description
`created_at`	String	—
`id`	String	—
`pipeline`	APIEntitiesPackagePipeline	—
`tags`	String	—
`version`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Packages: Cargo

Use this API to interact with the Cargo package registry, including its sparse index, registry configuration, and crate downloads. Generally these endpoints are used by Cargo itself and are not meant for manual consumption. For more information, see the package registry.

Get the sparse index for a Cargo crate (1-character name)

`GET /api/v4/projects/{id}/packages/cargo/1/{package_name}`

Returns newline-delimited JSON, one line per published version, most recently published first. Limited to the 500 most recently published versions.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the Cargo package

Responses

Code	Description	Schema
`200`	OK	String (`text/plain`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get the sparse index for a Cargo crate (2-character name)

`GET /api/v4/projects/{id}/packages/cargo/2/{package_name}`

Returns newline-delimited JSON, one line per published version, most recently published first. Limited to the 500 most recently published versions.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the Cargo package

Responses

Code	Description	Schema
`200`	OK	String (`text/plain`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get the sparse index for a Cargo crate (3-character name)

`GET /api/v4/projects/{id}/packages/cargo/3/{first_char}/{package_name}`

Returns newline-delimited JSON, one line per published version, most recently published first. Limited to the 500 most recently published versions.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`first_char` Path, required	String	First character of the Cargo package name
`package_name` Path, required	String	Name of the Cargo package

Responses

Code	Description	Schema
`200`	OK	String (`text/plain`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Publish a Cargo crate

`PUT /api/v4/projects/{id}/packages/cargo/api/v1/crates/new`

Receives the cargo publish body and extracts the crate asynchronously

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The cargo publish body, uploaded by workhorse

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get config.json

`GET /api/v4/projects/{id}/packages/cargo/config.json`

This will be used by cargo for further requests

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a Cargo crate

`GET /api/v4/projects/{id}/packages/cargo/{package_name}/{package_version}/download`

This endpoint serves the .crate file for a given package name and version

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the Cargo package
`package_version` Path, required	String	Version of the Cargo package

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get the sparse index for a Cargo crate (4+ character name)

`GET /api/v4/projects/{id}/packages/cargo/{prefix_1}/{prefix_2}/{package_name}`

Returns newline-delimited JSON, one line per published version, most recently published first. Limited to the 500 most recently published versions.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`prefix_1` Path, required	String	First two characters of the Cargo package name
`prefix_2` Path, required	String	Next two characters of the Cargo package name
`package_name` Path, required	String	Name of the Cargo package

Responses

Code	Description	Schema
`200`	OK	String (`text/plain`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Packages: Composer

Use this API to interact with the Composer package manager client.

> This API is used by the Composer package manager client and is generally not meant for manual consumption.

These endpoints do not adhere to the standard API authentication methods. See the Composer package registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

List all packages for a group

`GET /api/v4/group/{id}/-/packages/composer/p/{sha}`

Lists all repository packages for a specified group. Composer V2 is recommended over V1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a group
`sha` Path, required	String	Shasum of current json Example: `673594f85a55fe3c0eb45df7bd2fa9d95a1601ab`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Composer v2 packages p2 endpoint for a group’s package versions metadata

`GET /api/v4/group/{id}/-/packages/composer/p2/{package_name}`

This feature was introduced in GitLab 13.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a group
`package_name` Path, required	String	The Composer package name Example: `my-composer-package`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve repository URL templates

`GET /api/v4/group/{id}/-/packages/composer/packages`

Retrieves the repository URL templates for requesting individual packages for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a group

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Composer packages endpoint for a group’s package versions metadata

`GET /api/v4/group/{id}/-/packages/composer/{package_name}`

This feature was introduced in GitLab 12.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a group
`package_name` Path, required	String	The Composer package name Example: `my-composer-package`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Create a package

`POST /api/v4/projects/{id}/packages/composer`

Creates a Composer package from a specified Git tag or branch for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a project

Request body (`application/json`)

Property	Type	Description
`branch`	String	The name of the branch. Exactly one of `tag` , `branch` must be provided Example: `release`
`tag`	String	The name of the tag. Exactly one of `tag` , `branch` must be provided Example: `v1.0.0`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Composer package endpoint to download a package archive

`GET /api/v4/projects/{id}/packages/composer/archives/{package_name}`

This feature was introduced in GitLab 13.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of a project
`sha` Query, required	String	Shasum of current json Example: `673594f85a55fe3c0eb45df7bd2fa9d95a1601ab`
`package_name` Path, required	String	The Composer package name Example: `my-composer-package`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Packages: Conan

> The Conan registry is not FIPS compliant and is disabled when FIPS mode is enabled. These endpoints all return `404 Not Found`.

Use this API to interact with the Conan package manager for v1 and v2 repositories. The v1 endpoints work for both projects and instances, and the availability of the v2 endpoints is controlled by a feature flag. Generally these endpoints are used by the Conan package manager client and are not meant for manual consumption.

> These endpoints do not adhere to the standard API authentication methods. See each route for details on how credentials are expected to be passed. Undocumented authentication methods might be removed in the future.

Search for a Conan package

`GET /api/v4/packages/conan/v1/conans/search`

Searches the instance for a specified Conan package.

Parameters

Name	Type	Description
`q` Query, required	String	Search query. Use `*` as a wildcard Example: `Hello*`
`ignorecase` Query	Boolean	If `true`, ignores case when searching Example: `false`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a recipe snapshot

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}`

Retrieves a snapshot of the files for a specified Conan recipe. The snapshot is a list of filenames with their associated MD5 hash.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeSnapshot
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a recipe and package

`DELETE /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}`

Deletes a specified Conan recipe and associated package files from the package registry.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a recipe manifest

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/digest`

Retrieves a manifest that includes a list of files and associated download URLs for a specified recipe.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe download URLs

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/download_urls`

Lists all files and associated download URLs for a specified recipe in the package registry. Returns the same payload as the recipe manifest endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package snapshot

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}`

Retrieves a snapshot of the files for a specified Conan package and reference. The snapshot is a list of filenames with their associated MD5 hash.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageSnapshot
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package manifest

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/digest`

Retrieves a manifest that includes a list of files and associated download URLs for a specified package.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all package download URLs

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/download_urls`

Lists all files and associated download URLs for a specified package in the package registry. Returns the same payload as the package manifest endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all package upload URLs

`POST /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/upload_urls`

Lists all the upload URLs for a specified collection of package files. The request must include a JSON object with the name and size of the individual files.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanUploadUrls
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve package references metadata

`GET /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/search`

Retrieves the metadata for all package references of a specified package. This feature was introduced in GitLab 18.0.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe upload URLs

`POST /api/v4/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/upload_urls`

Lists all the upload URLs for a specified collection of recipe files. The request must include a JSON object with the name and size of the individual files.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanUploadUrls
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a recipe file

`GET /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}`

Retrieves a specified recipe file from the package registry. You must use the download URL returned from the recipe download URLs endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a recipe file

`PUT /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}`

Uploads a specified recipe file to the package registry. You must use the upload URL returned from the recipe upload URLs endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan recipe file

`PUT /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}/authorize`

Authorizes the Conan recipe file.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package file

`GET /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}`

Retrieves a specified package file from the package registry. You must use the download URL returned from the package download URLs endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a package file

`PUT /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}`

Uploads a specified package file to the package registry. You must use the upload URL returned from the package upload URLs endpoint.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan package file

`PUT /api/v4/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}/authorize`

Authorizes the Conan package file.

Parameters

Name	Type	Description
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Verify availability of a Conan repository

`GET /api/v4/packages/conan/v1/ping`

Verifies availability of a Conan repository.

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not Found	—

Retrieve an authentication token

`GET /api/v4/packages/conan/v1/users/authenticate`

Retrieves an authentication token. Creates a JSON Web Token (JWT) for use as a Bearer header in other requests to the package registry.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`404`	Not Found	—

Verify authentication credentials

`GET /api/v4/packages/conan/v1/users/check_credentials`

Verifies authentication credentials for a Conan package registry.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`404`	Not Found	—

Search for a Conan package

`GET /api/v4/projects/{id}/packages/conan/v1/conans/search`

Searches the instance for a specified Conan package.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`q` Query, required	String	Search query. Use `*` as a wildcard Example: `Hello*`
`ignorecase` Query	Boolean	If `true`, ignores case when searching Example: `false`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a recipe snapshot

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}`

Retrieves a snapshot of the files for a specified Conan recipe. The snapshot is a list of filenames with their associated MD5 hash.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeSnapshot
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a recipe and package

`DELETE /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}`

Deletes a specified Conan recipe and associated package files from the package registry.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a recipe manifest

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/digest`

Retrieves a manifest that includes a list of files and associated download URLs for a specified recipe.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe download URLs

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/download_urls`

Lists all files and associated download URLs for a specified recipe in the package registry. Returns the same payload as the recipe manifest endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package snapshot

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}`

Retrieves a snapshot of the files for a specified Conan package and reference. The snapshot is a list of filenames with their associated MD5 hash.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageSnapshot
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package manifest

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/digest`

Retrieves a manifest that includes a list of files and associated download URLs for a specified package.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all package download URLs

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/download_urls`

Lists all files and associated download URLs for a specified package in the package registry. Returns the same payload as the package manifest endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageManifest
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

List all package upload URLs

`POST /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/packages/{conan_package_reference}/upload_urls`

Lists all the upload URLs for a specified collection of package files. The request must include a JSON object with the name and size of the individual files.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanUploadUrls
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve package references metadata

`GET /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/search`

Retrieves the metadata for all package references of a specified package. This feature was introduced in GitLab 18.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe upload URLs

`POST /api/v4/projects/{id}/packages/conan/v1/conans/{package_name}/{package_version}/{package_username}/{package_channel}/upload_urls`

Lists all the upload URLs for a specified collection of recipe files. The request must include a JSON object with the name and size of the individual files.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanUploadUrls
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a recipe file

`GET /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}`

Retrieves a specified recipe file from the package registry. You must use the download URL returned from the recipe download URLs endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a recipe file

`PUT /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}`

Uploads a specified recipe file to the package registry. You must use the upload URL returned from the recipe upload URLs endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan recipe file

`PUT /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/export/{file_name}/authorize`

Authorizes the Conan recipe file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package file

`GET /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}`

Retrieves a specified package file from the package registry. You must use the download URL returned from the package download URLs endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.`

Name	Type	Description
		`tgz` , `conaninfo.txt` , `conan_package.tgz` Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a package file

`PUT /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}`

Uploads a specified package file to the package registry. You must use the upload URL returned from the package upload URLs endpoint.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.`

Name	Type	Description
		`tgz` , `conaninfo.txt` , `conan_package.tgz` Minimum length: `1` Example: `conaninfo.txt`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan package file

`PUT /api/v4/projects/{id}/packages/conan/v1/files/{package_name}/{package_version}/{package_username}/{package_channel}/{recipe_revision}/package/{conan_package_reference}/{package_revision}/{file_name}/authorize`

Authorizes the Conan package file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe. GitLab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `103f6067a947f366ef91fc1b7da351c588d1827f`
`package_revision` Path, required	String	Revision of the package. Git Lab does not yet support Conan revisions, so the default value of `0` is always used Example: `0`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.`

Name	Type	Description
		`tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Verify availability of a Conan repository

`GET /api/v4/projects/{id}/packages/conan/v1/ping`

Verifies availability of a Conan repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an authentication token

`GET /api/v4/projects/{id}/packages/conan/v1/users/authenticate`

Retrieves an authentication token. Creates a JSON Web Token (JWT) for use as a Bearer header in other requests to the package registry.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Verify authentication credentials

`GET /api/v4/projects/{id}/packages/conan/v1/users/check_credentials`

Verifies authentication credentials for a Conan package registry.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Search for a Conan package

`GET /api/v4/projects/{id}/packages/conan/v2/conans/search`

Searches the instance for a specified Conan package.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`q` Query, required	String	Search query. Use `*` as a wildcard Example: `Hello*`
`ignorecase` Query	Boolean	If `true`, ignores case when searching Example: `false`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve latest recipe revision

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/latest`

Retrieves the revision hash and creation date of the latest package recipe. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRevision
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe revisions

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions`

Lists all revisions for a package recipe. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRecipeRevisions
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Delete recipe revision

`DELETE /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}`

Deletes a specified recipe revision from the registry. If the recipe revision is the only one, the package is deleted as well. This feature was introduced in GitLab 18.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all recipe files

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/files`

Lists all recipe files from the package registry. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanFilesList
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a recipe file

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/files/{file_name}`

Retrieves a specified recipe file from the package registry. This feature was introduced in GitLab 17.8.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a recipe file

`PUT /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/files/{file_name}`

Uploads a specified recipe file to the package registry. This feature was introduced in GitLab 17.10.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan recipe file

`PUT /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/files/{file_name}/authorize`

Authorizes the Conan recipe file. This feature was introduced in GitLab 17.10.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz` Minimum length: `1` Example: `conanfile.py`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve latest package revision

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/latest`

Retrieves the revision hash and creation date of the latest package revision for a specified recipe revision and package reference. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanRevision
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all package revisions

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions`

Lists all package revisions for a specified recipe revision and package reference. This feature was introduced in GitLab 18.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanPackageRevisions
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a package revision

`DELETE /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions/{package_revision}`

Deletes a specified package revision from the registry. If the package reference has only one package revision, the package reference is deleted as well. This feature was introduced in GitLab 18.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`
`package_revision` Path, required	String	Revision of the package Example: `3bdd2d8c8e76c876ebd1ac0469a4e72c`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all package files

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions/{package_revision}/files`

Lists all package files. This feature was introduced in GitLab 18.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`
`package_revision` Path, required	String	Revision of the package Example: `3bdd2d8c8e76c876ebd1ac0469a4e72c`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesConanFilesList
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a package file

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions/{package_revision}/files/{file_name}`

Retrieves a specified package file from the package registry. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`
`package_revision` Path, required	String	Revision of the package Example: `3bdd2d8c8e76c876ebd1ac0469a4e72c`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a package file

`PUT /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions/{package_revision}/files/{file_name}`

Uploads a specified package file to the package registry. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`
`package_revision` Path, required	String	Revision of the package Example: `3bdd2d8c8e76c876ebd1ac0469a4e72c`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Workhorse authorize the Conan package file

`PUT /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/packages/{conan_package_reference}/revisions/{package_revision}/files/{file_name}/authorize`

Authorizes the Conan package file. This feature was introduced in GitLab 17.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`
`conan_package_reference` Path, required	String	Reference hash of the Conan package. Conan generates this value Example: `5ab84d6acfe1f23c4fae0ab88f26e3a396351ac9`
`package_revision` Path, required	String	Revision of the package Example: `3bdd2d8c8e76c876ebd1ac0469a4e72c`
`file_name` Path, required	String	Name of the package file Allowed values: `conanfile.py`, `conanmanifest.txt`, `conan_sources.tgz`, `conan_export.tgz`, `conaninfo.txt`, `conan_package.tgz`

Name	Type	Description
		Minimum length: `1` Example: `conaninfo.txt`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve package references metadata by recipe revision

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/revisions/{recipe_revision}/search`

Retrieves the metadata for all package references associated with a specified recipe revision. This feature was introduced in GitLab 18.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`
`recipe_revision` Path, required	String	Revision of the recipe Example: `df28fd816be3a119de5ce4d374436b25`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve package references metadata

`GET /api/v4/projects/{id}/packages/conan/v2/conans/{package_name}/{package_version}/{package_username}/{package_channel}/search`

Retrieves the metadata for all package references of a specified package. This feature was introduced in GitLab 18.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package Example: `my-package`
`package_version` Path, required	String	Version of the package Example: `1.0`
`package_username` Path, required	String	Conan username of the package. This is the `+`-separated full path of the project Example: `my-group+my-project`
`package_channel` Path, required	String	Channel of the package Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve an authentication token

`GET /api/v4/projects/{id}/packages/conan/v2/users/authenticate`

Retrieves an authentication token. Creates a JSON Web Token (JWT) for use as a Bearer header in other requests to the package registry.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Verify authentication credentials

`GET /api/v4/projects/{id}/packages/conan/v2/users/check_credentials`

Verifies authentication credentials for a Conan package registry.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesPackagesConanFilesList

Property	Type	Description
`files`	Object	Example: `{"conan_sources.tgz":{},"conanfile.py":{},"conanmanifest.txt":{}}`

APIEntitiesPackagesConanPackageManifest

Property	Type	Description
`package_urls`	Object	Example: `{"conan_package.tgz":"https://gitlab.example.com/api/v4/packages/conan/v1/files/my-package/1.0/my-group+my-project/stable/packages/103f6067a947f366ef91fc1b7da351c588d1827f/0/conan_package.tgz"}`

APIEntitiesPackagesConanPackageRevisions

Property	Type	Description
`packageReference`	String	The Conan package reference Example: `packageTest/1.2.3@gitlab-org+conan/stable#1883c9f810f2d6e5b59d5285c7141970:133a1f2158ff2cf69739f316ec21143785be54c7`
`revisions`	Array of APIEntitiesPackagesConanRevision	List of package revisions

APIEntitiesPackagesConanPackageSnapshot

Property	Type	Description
`package_snapshot`	Object	Example: `{"conan_package.tgz":"749b29bdf72587081ca03ec033ee59dc"}`

APIEntitiesPackagesConanRecipeManifest

Property	Type	Description
`recipe_urls`	Object	Example: `{"conan_sources.tgz":"https://gitlab.example.com/api/v4/packages/conan/v1/files/my-package/1.0/my-group+my-project/stable/0/export/conan_sources.tgz"}`

APIEntitiesPackagesConanRecipeRevisions

Property	Type	Description
`reference`	String	The Conan package reference Example: `packageTest/1.2.3@gitlab-org+conan/stable`
`revisions`	Array of APIEntitiesPackagesConanRevision	List of recipe revisions

APIEntitiesPackagesConanRecipeSnapshot

Property	Type	Description
`recipe_snapshot`	Object	Example: `{"conan_sources.tgz":"eadf19b33f4c3c7e113faabf26e76277"}`

APIEntitiesPackagesConanRevision

Property	Type	Description
`revision`	String	The revision hash of the Conan recipe or package Example: `75151329520e7685dcf5da49ded2fec0`
`time`	String	The UTC timestamp when the revision was created Example: `2024-12-17T09:16:40.334Z`

APIEntitiesPackagesConanUploadUrls

Property	Type	Description
`upload_urls`	Object	Example: `{"conan_package.tgz":"https://gitlab.example.com/api/v4/packages/conan/v1/files/my-package/1.0/my-group+my-project/stable/0/package/103f6067a947f366ef91fc1b7da351c588d1827f/0/conan_package.tgz"}`

Packages: Debian

> These endpoints are under development and are not ready for production use due to limited functionality. The package endpoints are used by Debian package clients such as dput and apt-get, and are generally not meant for manual consumption.

Use this API to manage Debian distributions for groups and projects, and to publish and retrieve Debian packages. The distribution endpoints are behind a feature flag that is disabled by default. To use them, enable the group or project Debian API.

> These endpoints do not adhere to the standard API authentication methods. For the supported headers and token types, see the Debian registry documentation. Undocumented authentication methods might be removed in the future.

List all Debian group distributions

`GET /api/v4/groups/{id}/-/debian_distributions`

Lists all Debian distributions for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`codename` Query	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`suite` Query	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`origin` Query	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`label` Query	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`version` Query	String	Version of the Debian distribution Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u`

Name	Type	Description
		`017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `12`
`description` Query	String	Description of the Debian distribution Example: `My description`
`valid_time_duration_seconds` Query	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`
`components` Query	Array of strings	List of components of the Debian distribution Example: `["main"]`
`architectures` Query	Array of strings	List of architectures of the Debian distribution Example: `["amd64"]`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a Debian group distribution

`POST /api/v4/groups/{id}/-/debian_distributions`

Creates a Debian group distribution for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`architectures`	Array of strings	List of architectures of the Debian distribution
`codename` Required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`components`	Array of strings	List of components of the Debian distribution
`description`	String	Description of the Debian distribution Example: `My description`
`label`	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`origin`	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`suite`	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`valid_time_duration_seconds`	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a Debian group distribution

`GET /api/v4/groups/{id}/-/debian_distributions/{codename}`

Retrieves a specified Debian group distribution for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Update a Debian group distribution

`PUT /api/v4/groups/{id}/-/debian_distributions/{codename}`

Updates a specified Debian group distribution for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Request body (`application/json`)

Property	Type	Description
`architectures`	Array of strings	List of architectures of the Debian distribution
`components`	Array of strings	List of components of the Debian distribution
`description`	String	Description of the Debian distribution Example: `My description`
`label`	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`origin`	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`suite`	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`valid_time_duration_seconds`	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a Debian group distribution

`DELETE /api/v4/groups/{id}/-/debian_distributions/{codename}`

Deletes a specified Debian group distribution for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`suite` Query	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`origin` Query	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`label` Query	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`version` Query	String	Version of the Debian distribution Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `12`
`description` Query	String	Description of the Debian distribution Example: `My description`

Name	Type	Description
`valid_time_duration_seconds` Query	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`
`components` Query	Array of strings	List of components of the Debian distribution Example: `["main"]`
`architectures` Query	Array of strings	List of architectures of the Debian distribution Example: `["amd64"]`

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a Debian group distribution key

`GET /api/v4/groups/{id}/-/debian_distributions/{codename}/key.asc`

Retrieves a specified Debian group distribution key for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The signed Release file

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/InRelease`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The unsigned Release file

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/Release`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The Release file signature

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/Release.gpg`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The binary files index

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/binary-{architecture}/Packages`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The binary files index by hash

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/binary-{architecture}/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The installer (udeb) binary files index

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/debian-installer/binary-{architecture}/Packages`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The installer (udeb) binary files index by hash

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/debian-installer/binary-{architecture}/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The source files index

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/source/Sources`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The source files index by hash

`GET /api/v4/groups/{id}/-/packages/debian/dists/{distribution}/{component}/source/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download Debian package

`GET /api/v4/groups/{id}/-/packages/debian/pool/{distribution}/{project_id}/{letter}/{package_name}/{package_version}/{file_name}`

This feature was introduced in GitLab 14.2

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`project_id` Path, required	Integer	ID of the project
`distribution` Path, required	String	Codename or suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `my-distro`
`letter` Path, required	String	Debian classification of the package, either first-letter or lib-first-letter Example: `a`
`package_name` Path, required	String	Name of the source package Pattern: `^[a-z0-9][\-+.a-z0-9]+$` Example: `my-pkg`
`package_version` Path, required	String	Version of the source package Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `1.0.0`
`file_name` Path, required	String	Name of the package file Example: `example_1.0.0~alpha2_amd64.deb`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all Debian project distributions

`GET /api/v4/projects/{id}/debian_distributions`

Lists all Debian distributions for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`codename` Query	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`suite` Query	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`origin` Query	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`label` Query	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`version` Query	String	Version of the Debian distribution Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u`

Name	Type	Description
		`017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `12`
`description` Query	String	Description of the Debian distribution Example: `My description`
`valid_time_duration_seconds` Query	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`
`components` Query	Array of strings	List of components of the Debian distribution Example: `["main"]`
`architectures` Query	Array of strings	List of architectures of the Debian distribution Example: `["amd64"]`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a Debian project distribution

`POST /api/v4/projects/{id}/debian_distributions`

Creates a Debian project distribution for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`architectures`	Array of strings	List of architectures of the Debian distribution
`codename` Required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`components`	Array of strings	List of components of the Debian distribution
`description`	String	Description of the Debian distribution Example: `My description`
`label`	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`origin`	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`suite`	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`valid_time_duration_seconds`	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a Debian project distribution

`GET /api/v4/projects/{id}/debian_distributions/{codename}`

Retrieves a specified Debian project distribution for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Update a Debian project distribution

`PUT /api/v4/projects/{id}/debian_distributions/{codename}`

Updates a specified Debian project distribution for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Request body (`application/json`)

Property	Type	Description
`architectures`	Array of strings	List of architectures of the Debian distribution
`components`	Array of strings	List of components of the Debian distribution
`description`	String	Description of the Debian distribution Example: `My description`
`label`	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`origin`	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`suite`	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`valid_time_duration_seconds`	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Delete a Debian project distribution

`DELETE /api/v4/projects/{id}/debian_distributions/{codename}`

Deletes a specified Debian project distribution for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`
`suite` Query	String	Suite of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `unstable`
`origin` Query	String	Origin of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `Grep`
`label` Query	String	Label of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `grep.be`
`version` Query	String	Version of the Debian distribution Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `12`
`description` Query	String	Description of the Debian distribution Example: `My description`

Name	Type	Description
`valid_time_duration_seconds` Query	Integer	Duration, in seconds, before the Release file is considered expired by the client Example: `604800`
`components` Query	Array of strings	List of components of the Debian distribution Example: `["main"]`
`architectures` Query	Array of strings	List of architectures of the Debian distribution Example: `["amd64"]`

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a Debian project distribution key

`GET /api/v4/projects/{id}/debian_distributions/{codename}/key.asc`

Retrieves a specified Debian project distribution key for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`codename` Path, required	String	Codename of the Debian distribution Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `sid`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPackagesDebianDistribution
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The signed Release file

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/InRelease`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The unsigned Release file

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/Release`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The Release file signature

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/Release.gpg`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The binary files index

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/binary-{architecture}/Packages`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The binary files index by hash

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/binary-{architecture}/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The installer (udeb) binary files index

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/debian-installer/binary-{architecture}/Packages`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The installer (udeb) binary files index by hash

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/debian-installer/binary-{architecture}/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`architecture` Path, required	String	The Debian Architecture Example: `binary-amd64`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The source files index

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/source/Sources`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The source files index by hash

`GET /api/v4/projects/{id}/packages/debian/dists/{distribution}/{component}/source/by-hash/SHA256/{file_sha256}`

This feature was introduced in GitLab 15.4

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Example: `my-distro`
`component` Path, required	String	The Debian Component Example: `main`
`file_sha256` Path, required	String	The SHA256 checksum of the index file

Responses

Code	Description	Schema
`200`	OK	—
`202`	Accepted	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a Debian package file

`GET /api/v4/projects/{id}/packages/debian/pool/{distribution}/{letter}/{package_name}/{package_version}/{file_name}`

Downloads a specified Debian package file for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`distribution` Path, required	String	The Debian Codename or Suite Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)` Example: `my-distro`
`letter` Path, required	String	The Debian Classification (first-letter or lib-first-letter) Example: `a`
`package_name` Path, required	String	The Debian Source Package Name Pattern: `^[a-z0-9][\-+.a-z0-9]+$` Example: `my-pkg`
`package_version` Path, required	String	The Debian Source Package Version Pattern: `(?:^(?:(?:([0-9]{1,9}):)?([0-9][+.0-9A-Za-z~\u017F\u212A]*)(-[+.0-9A-Za-z~\u017F\u212A]+){0,14}(?<!-))$)` Example: `1.0.0`
`file_name` Path, required	String	The Debian File Name Example: `example_1.0.0~alpha2_amd64.deb`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a Debian package file

`PUT /api/v4/projects/{id}/packages/debian/{file_name}`

Uploads a Debian package file for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`file_name` Path, required	String	Name of the package file Pattern: `^(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.(deb|udeb|ddeb)$` Example: `example_1.0.0~alpha2_amd64.deb`

Request body (`multipart/form-data`)

Property	Type	Description
`component` Required	String	The Debian Component Pattern: `^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$`
`distribution`	String	The Debian Codename or Suite Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)`
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Authorize Debian package upload

`PUT /api/v4/projects/{id}/packages/debian/{file_name}/authorize`

This feature was introduced in GitLab 13.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`file_name` Path, required	String	Name of the package file Pattern: `^(?:[\uD800-\uDBFF][\uDC00-\uDFFF]|[^\n\uD800-\uDFFF])*\.(deb|udeb|ddeb)$` Example: `example_1.0.0~alpha2_amd64.deb`

Request body (`application/json`)

Property	Type	Description
`component` Required	String	The Debian Component Pattern: `^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$`
`distribution`	String	The Debian Codename or Suite Pattern: `(?:^(?:[0-9A-Za-z\u017F\u212A][\x2D.0-9A-Za-z\u017F\u212A]*)$)`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesPackagesDebianDistribution

Property	Type	Description
`architectures`	Array of strings	Example: `["amd64"]`
`codename`	String	Example: `sid`
`components`	Array of strings	Example: `["main"]`
`description`	String	Example: `My description`
`id`	Integer (int64)	Example: `1`
`label`	String	Example: `grep.be`
`origin`	String	Example: `Grep`
`suite`	String	Example: `unstable`
`valid_time_duration_seconds`	Integer	Example: `604800`
`version`	String	Example: `12`

Packages: Helm

Use this API to interact with Helm package clients.

> This API is used by the Helm-related package clients such as Helm and `helm-push`, and is generally not meant for manual consumption.

These endpoints do not adhere to the standard API authentication methods. See the Helm registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

Upload a chart

`POST /api/v4/projects/{id}/packages/helm/api/{channel}/charts`

Uploads a specified chart for a project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The ID or full path of a project
`channel` Path, required	String	Helm channel Pattern: `^([a-zA-Z0-9](\.|-|_)?){1,255}(?<!\.|-|_)$` Example: `stable`

Request body (`multipart/form-data`)

Property	Type	Description
`chart` Required	String (binary)	The chart file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Authorize a chart upload from workhorse

`POST /api/v4/projects/{id}/packages/helm/api/{channel}/charts/authorize`

This feature was introduced in GitLab 14.0

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The ID or full path of a project
`channel` Path, required	String	Helm channel Pattern: `^([a-zA-Z0-9](\.|-|_)?){1,255}(?<!\.|-|_)$` Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a chart

`GET /api/v4/projects/{id}/packages/helm/{channel}/charts/{file_name}.tgz`

Downloads a specified chart for a project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The ID or full path of a project
`channel` Path, required	String	Helm channel Pattern: `^([a-zA-Z0-9](\.|-|_)?){1,255}(?<!\.|-|_)$` Example: `stable`
`file_name` Path, required	String	Helm package file name Example: `mychart`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a chart index

`GET /api/v4/projects/{id}/packages/helm/{channel}/index.yaml`

Downloads a specified chart index for a project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The ID or full path of a project
`channel` Path, required	String	Helm channel Pattern: `^([a-zA-Z0-9](\.|-|_)?){1,255}(?<!\.|-|_)$` Example: `stable`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Packages: NPM

Use this API to interact with the npm package manager client.

> This API is used by the npm package manager client and is not meant for manual consumption.

These endpoints do not adhere to the standard API authentication methods. See the npm package registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

NPM registry bulk advisory endpoint

`POST /api/v4/groups/{id}/-/packages/npm/-/npm/v1/security/advisories/bulk`

This feature was introduced in GitLab 15.6

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry quick audit endpoint

`POST /api/v4/groups/{id}/-/packages/npm/-/npm/v1/security/audits/quick`

This feature was introduced in GitLab 15.6

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get all tags for a given NPM package

`GET /api/v4/groups/{id}/-/packages/npm/-/package/{package_name}/dist-tags`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group
`package_name` Path, required	String	Package name

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNpmPackageTag
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create or Update the given tag for the given NPM package and version

`PUT /api/v4/groups/{id}/-/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Deletes the given tag

`DELETE /api/v4/groups/{id}/-/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry metadata endpoint

`GET /api/v4/groups/{id}/-/packages/npm/{package_name}`

This feature was introduced in GitLab 11.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group
`package_name` Path, required	String	Package name Example: `mypackage`

Responses

Code	Description	Schema
`200`	Ok	APIEntitiesNpmPackage
`302`	Found (redirect)	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry bulk advisory endpoint

`POST /api/v4/packages/npm/-/npm/v1/security/advisories/bulk`

This feature was introduced in GitLab 15.6

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry quick audit endpoint

`POST /api/v4/packages/npm/-/npm/v1/security/audits/quick`

This feature was introduced in GitLab 15.6

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get all tags for a given NPM package

`GET /api/v4/packages/npm/-/package/{package_name}/dist-tags`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`package_name` Path, required	String	Package name

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNpmPackageTag
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create or Update the given tag for the given NPM package and version

`PUT /api/v4/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Deletes the given tag

`DELETE /api/v4/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry metadata endpoint

`GET /api/v4/packages/npm/{package_name}`

This feature was introduced in GitLab 11.8

Parameters

Name	Type	Description
`package_name` Path, required	String	Package name Example: `mypackage`

Responses

Code	Description	Schema
`200`	Ok	APIEntitiesNpmPackage
`302`	Found (redirect)	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry bulk advisory endpoint

`POST /api/v4/projects/{id}/packages/npm/-/npm/v1/security/advisories/bulk`

This feature was introduced in GitLab 15.6

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry quick audit endpoint

`POST /api/v4/projects/{id}/packages/npm/-/npm/v1/security/audits/quick`

This feature was introduced in GitLab 15.6

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	Ok	—
`307`	Temporary Redirect	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Get all tags for a given NPM package

`GET /api/v4/projects/{id}/packages/npm/-/package/{package_name}/dist-tags`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Package name

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNpmPackageTag
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create or Update the given tag for the given NPM package and version

`PUT /api/v4/projects/{id}/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Deletes the given tag

`DELETE /api/v4/projects/{id}/packages/npm/-/package/{package_name}/dist-tags/{tag}`

This feature was introduced in GitLab 12.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Package name
`tag` Path, required	String	Package dist-tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

NPM registry metadata endpoint

`GET /api/v4/projects/{id}/packages/npm/{package_name}`

This feature was introduced in GitLab 11.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Package name Example: `mypackage`

Responses

Code	Description	Schema
`200`	Ok	APIEntitiesNpmPackage
`302`	Found (redirect)	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create or deprecate an NPM package

`PUT /api/v4/projects/{id}/packages/npm/{package_name}`

Creates or deprecates an NPM package for a specified project. Deprecate support was added in GitLab 16.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download the NPM tarball

`GET /api/v4/projects/{id}/packages/npm/{package_name}/-/{file_name}`

This feature was introduced in GitLab 11.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	Name of the package
`file_name` Path, required	String	Package file name

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesNpmPackage

Property	Type	Description
`dist-tags`	Object	Example: `{"latest":"1.0.1"}`
`name`	String	Example: `my_package`
`versions`	Object	Example: `{"1.0.0":{"dist":{"shasum":"12345","tarball":"https://..."},"name":"my_package","version":"1.0.0"}}`

APIEntitiesNpmPackageTag

Property	Type	Description
`dist_tags`	Object	Example: `{"latest":"1.0.1"}`

Packages: NuGet

Use this API to interact with the NuGet package manager client.

> This API is used by the NuGet package manager client and is generally not meant for manual consumption.

These endpoints do not adhere to the standard API authentication methods. See the NuGet package registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

The NuGet V3 Feed Service Index

`GET /api/v4/groups/{id}/-/packages/nuget/index`

This feature was introduced in GitLab 12.6

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetServiceIndex
`400`	Bad Request	—
`404`	Not Found	—

The NuGet Metadata Service - Package name level

`GET /api/v4/groups/{id}/-/packages/nuget/metadata/{package_name}/index`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path
`package_name` Path, required	String	The NuGet package name Example: `MyNuGetPkg`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetPackagesMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Metadata Service - Package name and version level

`GET /api/v4/groups/{id}/-/packages/nuget/metadata/{package_name}/{package_version}`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path
`package_name` Path, required	String	The NuGet package name Example: `MyNuGetPkg`
`package_version` Path, required	String	The NuGet package version Pattern: `[^\/]+` Example: `1.0.0`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetPackageMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Search Service

`GET /api/v4/groups/{id}/-/packages/nuget/query`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path
`q` Query	String	Search term Example: `MyNuGet`
`skip` Query	Integer	Number of results to skip Default: `0` Example: `1`
`take` Query	Integer	Number of results to return Default: `20` Example: `1`
`prerelease` Query	Boolean	If `true`, includes prerelease versions in the results Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetSearchResults
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Symbol File Download Endpoint

`GET /api/v4/groups/{id}/-/packages/nuget/symbolfiles/{file_name}/{signature}/{same_file_name}`

This feature was introduced in GitLab 16.7

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path
`file_name` Path, required	String	The symbol file name Pattern: `[^\/]+` Example: `mynugetpkg.pdb`
`signature` Path, required	String	The symbol file signature Pattern: `[^\/]+` Example: `k813f89485474661234z7109cve5709eFFFFFFFF`
`same_file_name` Path, required	String	The symbol file name. Must match the ‘file_name’ parameter

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet V2 Feed Service Index

`GET /api/v4/groups/{id}/-/packages/nuget/v2`

This feature was introduced in GitLab 16.2

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

The NuGet V2 Feed Package $metadata endpoint

`GET /api/v4/groups/{id}/-/packages/nuget/v2/$metadata`

This feature was introduced in GitLab 16.3

Parameters

Name	Type	Description
`id` Path, required	Integer or string	The group ID or full group path

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Upload a NuGet v3 package file for a project

`PUT /api/v4/projects/{id}/packages/nuget`

Uploads a NuGet v3 package file for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`package` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Package Authorize endpoint

`PUT /api/v4/projects/{id}/packages/nuget/authorize`

This feature was introduced in GitLab 14.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Content Service - index request

`GET /api/v4/projects/{id}/packages/nuget/download/{package_name}/index`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Example: `mynugetpkg.1.3.0.17.nupkg`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetPackagesVersions
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Content Service - content request

`GET /api/v4/projects/{id}/packages/nuget/download/{package_name}/{package_version}/{package_filename}`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Example: `mynugetpkg.1.3.0.17.nupkg`
`package_version` Path, required	String	The NuGet package version Pattern: `[^\/]+` Example: `1.3.0.17`
`package_filename` Path, required	String	The NuGet package filename Pattern: `[^\/]+` Example: `mynugetpkg.1.3.0.17.nupkg`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet V3 Feed Service Index

`GET /api/v4/projects/{id}/packages/nuget/index`

This feature was introduced in GitLab 12.6

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetServiceIndex
`400`	Bad Request	—
`404`	Not Found	—

The NuGet Metadata Service - Package name level

`GET /api/v4/projects/{id}/packages/nuget/metadata/{package_name}/index`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Example: `MyNuGetPkg`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetPackagesMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Metadata Service - Package name and version level

`GET /api/v4/projects/{id}/packages/nuget/metadata/{package_name}/{package_version}`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Example: `MyNuGetPkg`
`package_version` Path, required	String	The NuGet package version Pattern: `[^\/]+` Example: `1.0.0`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetPackageMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Search Service

`GET /api/v4/projects/{id}/packages/nuget/query`

This feature was introduced in GitLab 12.8

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`q` Query	String	Search term Example: `MyNuGet`
`skip` Query	Integer	Number of results to skip Default: `0` Example: `1`
`take` Query	Integer	Number of results to return Default: `20` Example: `1`
`prerelease` Query	Boolean	If `true`, includes prerelease versions in the results Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesNugetSearchResults
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Symbol File Download Endpoint

`GET /api/v4/projects/{id}/packages/nuget/symbolfiles/{file_name}/{signature}/{same_file_name}`

This feature was introduced in GitLab 16.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	The symbol file name Pattern: `[^\/]+` Example: `mynugetpkg.pdb`
`signature` Path, required	String	The symbol file signature Pattern: `[^\/]+` Example: `k813f89485474661234z7109cve5709eFFFFFFFF`
`same_file_name` Path, required	String	The symbol file name. Must match the ‘file_name’ parameter

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a NuGet symbol package file

`PUT /api/v4/projects/{id}/packages/nuget/symbolpackage`

Uploads a specified NuGet symbol package file (`.snupkg`) for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`package` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Symbol Package Authorize endpoint

`PUT /api/v4/projects/{id}/packages/nuget/symbolpackage/authorize`

This feature was introduced in GitLab 14.1

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet V2 Feed Service Index

`GET /api/v4/projects/{id}/packages/nuget/v2`

This feature was introduced in GitLab 16.2

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Upload a NuGet v2 package file for a project

`PUT /api/v4/projects/{id}/packages/nuget/v2`

Uploads a NuGet v2 package file for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`package` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet V2 Feed Package $metadata endpoint

`GET /api/v4/projects/{id}/packages/nuget/v2/$metadata`

This feature was introduced in GitLab 16.3

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

The NuGet V2 Feed Package Authorize endpoint

`PUT /api/v4/projects/{id}/packages/nuget/v2/authorize`

This feature was introduced in GitLab 16.2

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet Package Delete endpoint

`DELETE /api/v4/projects/{id}/packages/nuget/{package_name}/{package_version}`

This feature was introduced in GitLab 16.5

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Pattern: `^[\-+._a-zA-Z0-9]+$` Example: `mynugetpkg`
`package_version` Path, required	String	The NuGet package version Pattern: `^(?:(0|[1-9]\d*))\.(?:(0|[1-9]\d*))(\.(?:(0|[1-9]\d*)))?(\.\d*)?(?:(?:-((?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0)(?:\.(?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0))*))?(?:\+([0-9a-zA-Z\-]+(?:\.[0-9a-zA-Z\-]+)*))?)$` Example: `1.0.1`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The NuGet V2 Feed Find Packages by ID endpoint

`GET /api/v4/projects/{project_id}/packages/nuget/v2/FindPackagesById()`

This feature was introduced in GitLab 16.4

Parameters

Name	Type	Description
`project_id` Path, required	String or integer	The ID or URL-encoded path of the project
`id` Query, required	String	The NuGet package name Pattern: `^[\-+._a-zA-Z0-9]+$` Example: `mynugetpkg`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

The NuGet V2 Feed Enumerate Packages endpoint

`GET /api/v4/projects/{project_id}/packages/nuget/v2/Packages()`

This feature was introduced in GitLab 16.4

Parameters

Name	Type	Description
`project_id` Path, required	String or integer	The ID or URL-encoded path of the project
`$filter` Query, required	String	The NuGet package name Pattern: `^[\-+._a-zA-Z0-9]+$` Example: `mynugetpkg`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

The NuGet V2 Feed Single Package Metadata endpoint

`GET /api/v4/projects/{project_id}/packages/nuget/v2/Packages(Id='{package_name}',Version='{package_version}')`

This feature was introduced in GitLab 16.4

Parameters

Name	Type	Description
`project_id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_name` Path, required	String	The NuGet package name Pattern: `^[\-+._a-zA-Z0-9]+$` Example: `mynugetpkg`
`package_version` Path, required	String	The NuGet package version Pattern: `^(?:(0|[1-9]\d*))\.(?:(0|[1-9]\d*))(\.(?:(0|[1-9]\d*)))?(\.\d*)?(?:(?:-((?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0)(?:\.(?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0))*))?(?:\+([0-9a-zA-Z\-]+(?:\.[0-9a-zA-Z\-]+)*))?)$` Example: `1.3.0.17`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesNugetDependency

Property	Type	Description
`@id`	String	Example: `http://gitlab.com/Sandbox.App/1.0.0.json#dependency`
`@type`	String	Example: `PackageDependency`
`id`	String	Example: `Dependency`
`range`	String	Example: `2.0.0`

APIEntitiesNugetDependencyGroup

Property	Type	Description
`@id`	String	Example: `http://gitlab.com/Sandbox.App/1.0.0.json#dependencygroup`
`@type`	String	Example: `PackageDependencyGroup`
`dependencies`	Array of APIEntitiesNugetDependency	—
`targetFramework`	String	Example: `fwk test`

APIEntitiesNugetPackageMetadata

Property	Type	Description
`@id`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/metadata/MyNuGetPkg/1.3.0.17.json`
`catalogEntry`	APIEntitiesNugetPackageMetadataCatalogEntry	—
`packageContent`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/download/MyNuGetPkg/1.3.0.17/helloworld.1.3.0.17.nupkg`

APIEntitiesNugetPackageMetadataCatalogEntry

Property	Type	Description
`@id`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/metadata/MyNuGetPkg/1.3.0.17.json`
`authors`	String	Example: `Authors`
`dependencyGroups`	Array of APIEntitiesNugetDependencyGroup	—
`description`	String	Example: `Description`
`iconUrl`	String	Example: `http://sandbox.com/icon`
`id`	String	Example: `MyNuGetPkg`
`licenseUrl`	String	Example: `http://sandbox.com/license`
`packageContent`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/download/MyNuGetPkg/1.3.0.17/helloworld.1.3.0.17.nupkg`
`projectUrl`	String	Example: `http://sandbox.com/project`
`published`	String	Example: `2023-05-08T17:23:25Z`
`summary`	String	Example: `Description`
`tags`	String	Example: `tag#1 tag#2`
`version`	String	Example: `1.3.0.17`

APIEntitiesNugetPackagesMetadata

Property	Type	Description
`count`	Integer	Example: `1`
`items`	Array of APIEntitiesNugetPackagesMetadataItem	—

APIEntitiesNugetPackagesMetadataItem

Property	Type	Description
`@id`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/metadata/MyNuGetPkg/1.3.0.17.json`
`count`	Integer	Example: `1`
`items`	Array of APIEntitiesNugetPackageMetadata	—
`lower`	String	Example: `1.3.0.17`
`upper`	String	Example: `1.3.0.17`

APIEntitiesNugetPackagesVersions

Property	Type	Description
`versions`	Array of strings	Example: `["1.3.0.17"]`

APIEntitiesNugetSearchResult

Property	Type	Description
`@type`	String	Example: `Package`
`authors`	String	Example: `Authors`
`description`	String	Example: `Description`
`iconUrl`	String	Example: `http://sandbox.com/icon`
`id`	String	Example: `MyNuGetPkg`
`licenseUrl`	String	Example: `http://sandbox.com/license`
`projectUrl`	String	Example: `http://sandbox.com/project`
`summary`	String	Example: `Description`
`tags`	String	Example: `tag#1 tag#2`
`title`	String	Example: `MyNuGetPkg`
`totalDownloads`	Integer	Example: `1`
`verified`	Boolean	—
`version`	String	Example: `1.3.0.17`
`versions`	APIEntitiesNugetSearchResultVersion	—

APIEntitiesNugetSearchResults

Property	Type	Description
`data`	Array of APIEntitiesNugetSearchResult	—
`totalHits`	Integer	Example: `1`

APIEntitiesNugetSearchResultVersion

Property	Type	Description
`@id`	String	Example: `https://gitlab.example.com/api/v4/projects/1/packages/nuget/metadata/MyNuGetPkg/1.3.0.17.json`
`downloads`	Integer	Example: `1`
`version`	String	Example: `1.3.0.17`

APIEntitiesNugetServiceIndex

Property	Type	Description
`resources`	Array of objects	Example: `[{"@id":"https://gitlab.com/api/v4/projects/1/packages/nuget/query","@type":"SearchQueryService","comment":"Filter and search for packages by keyword."}]`
`version`	String	Example: `1.3.0.17`

Packages: PyPi

Use this API to interact with the PyPI package manager client.

> This API is used by the PyPI package manager client and is generally not meant for manual consumption.

These endpoints do not adhere to the standard API authentication methods. See the PyPI package registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

> Twine 3.4.2 or greater is recommended when FIPS mode is enabled.

Download a package file from a group

`GET /api/v4/groups/{id}/-/packages/pypi/files/{sha256}/{file_identifier}`

This feature was introduced in GitLab 13.12

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group
`file_identifier` Path, required	String	The PyPi package file identifier Example: `my.pypi.package-0.0.1.tar.gz`
`sha256` Path, required	String	The PyPi package sha256 check sum Example: `5y57017232013c8ac80647f4ca153k3726f6cba62d055cd747844ed95b3c65ff`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all packages for a group

`GET /api/v4/groups/{id}/-/packages/pypi/simple`

Lists all packages for a specified group in an HTML file.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The PyPi Simple Group Package Endpoint

`GET /api/v4/groups/{id}/-/packages/pypi/simple/{package_name}`

This feature was introduced in GitLab 12.10

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the group
`package_name` Path, required	String	The PyPi package name Example: `my.pypi.package`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a package

`POST /api/v4/projects/{id}/packages/pypi`

Uploads a PyPI package for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`author_email`	String	Email address for the package author Example: `cschultz@example.com, snoopy@peanuts.com`
`content` Required	String (binary)	The package file to be published (generated by Multipart middleware)
`description`	String	Description of the package
`description_content_type`	String	Type of content for the package description Example: `text/markdown; charset=UTF-8; variant=GFM`
`keywords`	String	Keywords listed for the package Example: `dog,puppy,voting,election`
`md5_digest`	String	MD5 checksum of the package Example: `900150983cd24fb0d6963f7d28e17f72`
`metadata_version`	String	Metadata version of the package Example: `2.3`
`name` Required	String	Name of the package Example: `my.pypi.package`
`requires_python`	String	PyPI version required for the package Example: `>=3.7`
`sha256_digest`	String	SHA256 checksum of the package Pattern: `(?:^[0-9A-Fa-f]{64}$)` Example: `ba7816bf8f01cfea414140de5da`

Property	Type	Description
		`e2223b00361a396177a9cb410ff61f20015ad`
`summary`	String	Short summary of the package Example: `A module for collecting votes from beagles.`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Authorize the PyPi package upload from workhorse

`POST /api/v4/projects/{id}/packages/pypi/authorize`

This feature was introduced in GitLab 12.10

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The PyPi package download endpoint

`GET /api/v4/projects/{id}/packages/pypi/files/{sha256}/{file_identifier}`

This feature was introduced in GitLab 12.10

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`file_identifier` Path, required	String	The PyPi package file identifier Example: `my.pypi.package-0.0.1.tar.gz`
`sha256` Path, required	String	The PyPi package sha256 check sum Example: `5y57017232013c8ac80647f4ca153k3726f6cba62d055cd747844ed95b3c65ff`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a forwarded (proxied) PyPI package file

`GET /api/v4/projects/{id}/packages/pypi/forward/{package_name}/{upstream_path}`

Enforces the Dependency Firewall then redirects to the upstream artifact.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	The normalized PyPI package name Example: `my.pypi.package`
`upstream_path` Path, required	String	The upstream artifact host and path, as emitted by the Simple index rewrite

Responses

Code	Description	Schema
`302`	Found	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all packages for a project

`GET /api/v4/projects/{id}/packages/pypi/simple`

Lists all packages for a specified project in an HTML file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

The PyPi Simple Project Package Endpoint

`GET /api/v4/projects/{id}/packages/pypi/simple/{package_name}`

This feature was introduced in GitLab 12.10

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_name` Path, required	String	The PyPi package name Example: `my.pypi.package`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Packages: RPM

Use this API to upload RPM packages to the package registry.

Upload a RPM package

`POST /api/v4/projects/{id}/packages/rpm`

This feature was introduced in GitLab 15.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Authorize package upload from workhorse

`POST /api/v4/projects/{id}/packages/rpm/authorize`

This feature was introduced in GitLab 15.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download repository metadata files

`GET /api/v4/projects/{id}/packages/rpm/repodata/{file_name}`

This feature was introduced in GitLab 15.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`file_name` Path, required	String	Repository metadata file name

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download RPM package files

`GET /api/v4/projects/{id}/packages/rpm/{package_file_id}/{file_name}`

This feature was introduced in GitLab 15.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`package_file_id` Path, required	Integer	RPM package file id
`file_name` Path, required	String	RPM package file name

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Packages: RubyGem

Use this API to interact with the Ruby gems and Bundler package manager clients.

> This API is used by the Ruby gems and Bundler package manager clients and is generally not meant for manual consumption. This API is under development and is not ready for production use due to limited functionality.

These endpoints do not adhere to the standard API authentication methods. See the Ruby gems registry documentation for details on which headers and token types are supported. Undocumented authentication methods might be removed in the future.

Retrieve dependencies

`GET /api/v4/projects/{id}/packages/rubygems/api/v1/dependencies`

Retrieves a list of dependencies for specified gems. The response is a marshalled array of hashes for all versions of the requested gems. Because the response is marshalled, you can store it in a file.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project
`gems` Query	Array of strings	Comma-separated list of gems to fetch dependencies for

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Upload a gem

`POST /api/v4/projects/{id}/packages/rubygems/api/v1/gems`

Uploads a gem for a specified project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Authorize a gem upload from workhorse

`POST /api/v4/projects/{id}/packages/rubygems/api/v1/gems/authorize`

This feature was introduced in GitLab 13.9

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a gem file

`GET /api/v4/projects/{id}/packages/rubygems/gems/{file_name}`

Downloads a specified gem file for a project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project
`file_name` Path, required	String	Name of the `.gem` file

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download a gemspec file

`GET /api/v4/projects/{id}/packages/rubygems/quick/Marshal.4.8/{file_name}`

Downloads a gemspec file in Marshal format for a specified gem version.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project
`file_name` Path, required	String	Name of the gemspec file, in the format `<gem_name>-<version>.gemspec.rz`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Download the spec index file

`GET /api/v4/projects/{id}/packages/rubygems/{file_name}`

Downloads a RubyGems spec index file (specs.4.8.gz, latest_specs.4.8.gz, or prerelease_specs.4.8.gz) for a project.

Parameters

Name	Type	Description
`id` Path, required	Integer or string	ID or URL-encoded path of the project
`file_name` Path, required	String	Name of the spec file Allowed values: `specs.4.8.gz` , `latest_specs.4.8.gz` , `prerelease_specs.4.8.gz` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Pipeline schedules

Use this API to interact with pipeline schedules.

List all pipeline schedules

`GET /api/v4/projects/{id}/pipeline_schedules`

Lists all pipeline schedules for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`scope` Query	String	Scope of pipeline schedules Allowed values: `active`, `inactive` Example: `active`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiPipeline Schedule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create a pipeline schedule

`POST /api/v4/projects/{id}/pipeline_schedules`

Creates a pipeline schedule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`

Request body (`application/json`)

Property	Type	Description
`active`	Boolean	If `true`, activates the pipeline schedule. If `false`, the pipeline schedule is initially deactivated Default: `true` Example: `true`
`cron` Required	String	Cron schedule, for example `0 1 * * *` Example: `* * * * *`
`cron_timezone`	String	Time zone supported by `ActiveSupport::TimeZone`, for example `Pacific Time (US & Canada)`, or `TZInfo::Timezone`, for example `America/Los_Angeles` Default: `UTC` Example: `Asia/Tokyo`
`description` Required	String	Description of the pipeline schedule Example: `Test schedule pipeline`
`inputs`	Array of objects	Array of inputs to pass to the pipeline schedule. Each input has a `name` and a `value`, which can be a string, array, number, or boolean. To delete an existing input, include the `name` field and set `destroy` to `true`
`inputs[].name` Required	String	Name of the input Example: `deploy_strategy`
`inputs[].value` Required	String or array or number or boolean	Value of the input

Property	Type	Description
`ref` Required	String	Branch or tag name that triggers the pipeline. Accepts either short refs (`main`) or full refs (`refs/heads/main` or `refs/tags/main`). Short refs are automatically expanded to full refs, unless the value could match either a branch or a tag Example: `develop`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipelineScheduleDetails
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a pipeline schedule

`GET /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}`

Retrieves a pipeline schedule for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiPipelineScheduleDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Update a pipeline schedule

`PUT /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}`

Updates a pipeline schedule for a project. After the update is done, it is rescheduled automatically.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Request body (`application/json`)

Property	Type	Description
`active`	Boolean	If `true`, activates the pipeline schedule. If `false`, the pipeline schedule is initially deactivated Example: `true`
`cron`	String	Cron schedule, for example `0 1 * * *` Example: `* * * * *`
`cron_timezone`	String	Time zone supported by `ActiveSupport::TimeZone`, for example `Pacific Time (US & Canada)`, or `TZInfo::Timezone`, for example `America/Los_Angeles` Example: `Asia/Tokyo`
`description`	String	Description of the pipeline schedule Example: `Test schedule pipeline`
`inputs`	Array of objects	Array of inputs to pass to the pipeline schedule. Each input has a `name` and a `value`, which can be a string, array, number, or boolean. To delete an existing input, include the `name` field and set `destroy` to `true`
`inputs[].destroy`	Boolean	If `true`, deletes the input
`inputs[].name` Required	String	Name of the input Example: `deploy_strategy`
`inputs[].value` Required	String or array or number or boolean	Value of the input
`ref`	String	

Property	Type	Description
		Branch or tag name that triggers the pipeline. Accepts either short refs (`main`) or full refs (`refs/heads/main` or `refs/tags/main`). Short refs are automatically expanded to full refs, unless the value could match either a branch or a tag Example: `develop`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiPipelineScheduleDetails
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a pipeline schedule

`DELETE /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}`

Deletes a pipeline schedule for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`412`	Precondition Failed	—

List all pipelines triggered by a pipeline schedule

`GET /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/pipelines`

Lists all pipelines triggered by a pipeline schedule in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`
`scope` Query	String	Return pipelines in the specified scope Allowed values: `running`, `pending`, `finished`, `branches`, `tags` Example: `pending`
`status` Query	String	Return pipelines with the specified status Allowed values: `created`, `waiting_for_resource`, `preparing`, `waiting_for_callback`, `pending`, `running`, `success`, `failed`, `canceling`, `canceled`, `skipped`, `manual`, `scheduled` Example: `pending`
`updated_before` Query	String (date-time)	Return pipelines updated on or before the specified time Example: `2015-12-24T15:51:21.880Z`
	String (date-time)	Return pipelines updated on or after the specified time

Name	Type	Description
`updated_after` Query		Example: `2015-12-24T15:51:21.880Z`
`created_before` Query	String (date-time)	Return pipelines created before the specified time Example: `2015-12-24T15:51:21.880Z`
`created_after` Query	String (date-time)	Return pipelines created after the specified time Example: `2015-12-24T15:51:21.880Z`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc` Example: `desc`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiPipelineBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Run a pipeline schedule

`POST /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/play`

Runs a pipeline schedule immediately. The next scheduled run of this pipeline is not affected.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create or update ownership of a pipeline schedule

`POST /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/take_ownership`

Creates or updates the owner of a pipeline schedule for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipelineScheduleDetails
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Create a variable for a pipeline schedule

`POST /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/variables`

Creates a variable for a pipeline schedule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`

Request body (`application/json`)

Property	Type	Description
`key` Required	String	Key of the pipeline schedule variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_` Example: `NEW_VARIABLE`
`value` Required	String	Value of the variable Example: `new value`
`variable_type`	String	Type of the variable. Defaults to `env_var` Allowed values: `env_var`, `file`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiVariable
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a variable for a pipeline schedule

`GET /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/variables/{key}`

Retrieves a specified variable for a pipeline schedule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`
`key` Path, required	String	Key of the pipeline schedule variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_` Example: `NEW_VARIABLE`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Update a variable for a pipeline schedule

`PUT /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/variables/{key}`

Updates a variable for a pipeline schedule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`
`key` Path, required	String	Key of the pipeline schedule variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_` Example: `NEW_VARIABLE`

Request body (`application/json`)

Property	Type	Description
`value`	String	Value of the variable Example: `new value`
`variable_type`	String	Type of the variable. Defaults to `env_var` Allowed values: `env_var`, `file`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiVariable
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a variable for a pipeline schedule

`DELETE /api/v4/projects/{id}/pipeline_schedules/{pipeline_schedule_id}/variables/{key}`

Deletes a specified variable for a pipeline schedule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `18`
`pipeline_schedule_id` Path, required	Integer	ID of the pipeline schedule Example: `13`
`key` Path, required	String	Key of the pipeline schedule variable. Must be 255 characters or fewer and contain only `A-Z`, `a-z`, `0-9`, and `_` Example: `NEW_VARIABLE`

Responses

Code	Description	Schema
`202`	Accepted	APIEntitiesCiVariable
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiInput

Property	Type	Description
`name`	String	—
`value`	String	—

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiPipelineSchedule

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2017-05-19T13:31:08.849Z`
`cron`	String	Example: `* * * * *`
`cron_timezone`	String	Example: `Asia/Tokyo`
`description`	String	Example: `Test schedule pipeline`
`id`	Integer (int64)	Example: `13`
`inputs`	APIEntitiesCiInput	—
`next_run_at`	String (date-time)	Example: `2017-05-19T13:41:00.000Z`
`owner`	APIEntitiesUserBasic	—
`ref`	String	Example: `develop`
`updated_at`	String (date-time)	Example: `2017-05-19T13:40:17.727Z`

APIEntitiesCiPipelineScheduleDetails

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2017-05-19T13:31:08.849Z`
`cron`	String	Example: `* * * * *`
`cron_timezone`	String	Example: `Asia/Tokyo`
`description`	String	Example: `Test schedule pipeline`
`id`	Integer (int64)	Example: `13`
`inputs`	APIEntitiesCiInput	—
`last_pipeline`	APIEntitiesCiPipelineBasic	—
`next_run_at`	String (date-time)	Example: `2017-05-19T13:41:00.000Z`
`owner`	APIEntitiesUserBasic	—
`ref`	String	Example: `develop`
`updated_at`	String (date-time)	Example: `2017-05-19T13:40:17.727Z`
`variables`	APIEntitiesCiVariable	—

APIEntitiesCiVariable

Property	Type	Description
`description`	String	Example: `This variable is being used for ...`
`environment_scope`	String	Example: `*`
`hidden`	Boolean	—
`key`	String	Example: `TEST_VARIABLE_1`
`masked`	Boolean	—
`protected`	Boolean	—
`raw`	Boolean	—
`value`	String	Example: `TEST_1`
`variable_type`	String	Example: `env_var`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Pipelines

Use this API to manage CI/CD pipelines and the jobs they run.

List pipelines triggered by the authenticated user

`GET /api/v4/pipelines`

Lists recently created pipelines across all projects that were triggered by the authenticated user. By default, child pipelines are not included in the results. To return child pipelines, set `source` to `parent_pipeline`. This endpoint only supports keyset pagination.

Parameters

Name	Type	Description
`source` Query	String	The source of pipelines Allowed values: `unknown`, `push`, `web`, `trigger`, `schedule`, `api`, `external`, `pipeline`, `chat`, `webide`, `merge_request_event`, `external_pull_request_event`, `parent_pipeline`, `ondemand_dast_scan`, `ondemand_dast_validation`, `security_orchestration_policy`, `container_registry_push`, `duo_workflow`, `pipeline_execution_policy_schedule`, `dependency_management_security_update` Example: `push`
`created_before` Query	String (date-time)	Return pipelines created before the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ Example: `2015-12-24T15:51:21.880Z`
`created_after` Query	String (date-time)	Return pipelines created after the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ Example: `2015-12-24T15:51:21.880Z`
`order_by` Query	String	Return pipelines ordered by `created_at` Allowed values: `created_at` Default: `created_at` Example: `created_at`
`sort` Query	String	Return pipelines sorted in `desc` order

Name	Type	Description
		Allowed values: `desc` Default: `desc` Example: `desc`
`cursor` Query	String	Cursor for obtaining the next set of records Example: `eyJpZCI6IjE0In0`
`per_page` Query	Integer	Number of items per page Default: `20` Maximum: `100` Minimum: `1` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiUser Pipeline
`400`	Bad Request	—
`401`	Unauthorized	—

Create a pipeline

`POST /api/v4/projects/{id}/pipeline`

Creates a pipeline in the specified project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`inputs`	Object	Map of inputs, as key-value pairs, to use when creating the pipeline. Generally available in GitLab 18.1. Feature flag `ci_inputs_for_pipelines` removed
`ref` Required	String	Reference Example: `develop`
`variables`	Array of objects	Array of variables to make available in the pipeline. If `variable_type` is omitted, it defaults to `env_var`
`variables[].key`	String	Key of the variable Example: `UPLOAD_TO_S3`
`variables[].value`	String	Value of the variable Example: `true`
`variables[].variable_type`	String	Type of the variable Allowed values: `env_var`, `file` Default: `env_var`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all project pipelines

`GET /api/v4/projects/{id}/pipelines`

Lists all pipelines in a project. By default, child pipelines are not included in the results. To return child pipelines, set `source` to `parent_pipeline`.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`scope` Query	String	Return pipelines in the specified scope Allowed values: `running`, `pending`, `finished`, `branches`, `tags` Example: `pending`
`status` Query	String	Return pipelines with the specified status Allowed values: `created`, `waiting_for_resource`, `preparing`, `waiting_for_callback`, `pending`, `running`, `success`, `failed`, `canceling`, `canceled`, `skipped`, `manual`, `scheduled` Example: `pending`
`ref` Query	String	Return pipelines for the specified branch or tag Example: `develop`
`sha` Query	String	Return pipelines for the specified commit SHA Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`

Name	Type	Description
`yaml_errors` Query	Boolean	If `true`, returns only pipelines with invalid configurations Example: `false`
`username` Query	String	Return pipelines triggered by the specified username Example: `root`
`updated_before` Query	String (date-time)	Return pipelines updated on or before the specified time Example: `2015-12-24T15:51:21.880Z`
`updated_after` Query	String (date-time)	Return pipelines updated on or after the specified time Example: `2015-12-24T15:51:21.880Z`
`created_before` Query	String (date-time)	Return pipelines created before the specified time Example: `2015-12-24T15:51:21.880Z`
`created_after` Query	String (date-time)	Return pipelines created after the specified time Example: `2015-12-24T15:51:21.880Z`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `status`, `ref`, `updated_at`, `user_id` Default: `id` Example: `status`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc` Example: `asc`
`source` Query	String	The source of pipelines Allowed values: `unknown`,

Name	Type	Description
		`push`, `web`, `trigger`, `schedule`, `api`, `external`, `pipeline`, `chat`, `webide`, `merge_request_event`, `external_pull_request_event`, `parent_pipeline`, `ondemand_dast_scan`, `ondemand_dast_validation`, `security_orchestration_policy`, `container_registry_push`, `duo_workflow`, `pipeline_execution_policy_schedule`, `dependency_management_security_update` Example: `push`
`name` Query	String	Return pipelines with the specified name Example: `Build pipeline`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiPipelineBasic
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve the latest pipeline

`GET /api/v4/projects/{id}/pipelines/latest`

Retrieves the latest pipeline for the most recent commit on a specified ref in a project. If no pipeline exists for the commit, a `403` status code is returned. Use the `page` and `per_page` pagination parameters to control the pagination of results.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`ref` Query	String	Branch or tag to check for the latest pipeline. If omitted, uses the default branch Example: `develop`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiPipelineWithMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a pipeline

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}`

Retrieves a specified pipeline from a project. You can also get a child pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiPipelineWithMetadata
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a pipeline

`DELETE /api/v4/projects/{id}/pipelines/{pipeline_id}`

Deletes a specified pipeline for a project.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`204`	Pipeline was deleted	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—
`429`	Too Many Requests	—

List all bridge jobs by pipeline (deprecated)

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/bridges`

Deprecated in GitLab 19.2. Use trigger_jobs endpoint instead.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`
`scope` Query	String or array of strings	The scope of builds to show Example: `["pending", "running"]`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiBridge
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Cancel all jobs for a pipeline

`POST /api/v4/projects/{id}/pipelines/{pipeline_id}/cancel`

Cancels all jobs in a specified pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too Many Requests	—

List all jobs by pipeline

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/jobs`

Lists all jobs for a specified pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`
`include_retried` Query	Boolean	If `true`, includes retried jobs in the response Default: `false`
`scope` Query	String or array of strings	The scope of builds to show Example: `["pending", "running"]`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiJob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Update pipeline metadata

`PUT /api/v4/projects/{id}/pipelines/{pipeline_id}/metadata`

Updates pipeline metadata. The metadata contains the name of the pipeline. This feature was introduced in GitLab 16.6.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Request body (`application/json`)

Property	Type	Description
`name` Required	String	New name of the pipeline Example: `Deployment to production`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiPipelineWithMetadata
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retry jobs in a pipeline

`POST /api/v4/projects/{id}/pipelines/{pipeline_id}/retry`

Retries failed or canceled jobs in a pipeline. If there are no failed or canceled jobs in the pipeline, calling this endpoint has no effect.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiPipeline
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`429`	Too Many Requests	—

Retrieve a test report for a pipeline

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/test_report`

Retrieves a test report for a pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`200`	OK	TestReportEntity
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve a test report summary for a pipeline

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/test_report_summary`

Retrieves a test report summary for a pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`200`	OK	TestReportSummaryEntity
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all trigger jobs by pipeline

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/trigger_jobs`

Lists all trigger jobs for a specified pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`
`scope` Query	String or array of strings	The scope of builds to show Example: `["pending", "running"]`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiBridge
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all pipeline variables

`GET /api/v4/projects/{id}/pipelines/{pipeline_id}/variables`

Lists all pipeline variables for a specified pipeline.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `11`
`pipeline_id` Path, required	Integer	ID of the pipeline Example: `18`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesCiVariable
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiBridge

Property	Type	Description
`allow_failure`	Boolean	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`downstream_pipeline`	APIEntitiesCiPipelineBasic	—
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`

Property	Type	Description
`tag`	Boolean	—
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJob

Property	Type	Description
`allow_failure`	Boolean	—
`archived`	Boolean	Example: `false`
`artifacts`	Array of APIEntitiesCiJobArtifact	—
`artifacts_expire_at`	String (date-time)	Example: `2016-01-19T09:05:50.355Z`
`artifacts_file`	APIEntitiesCiJobArtifactFile	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	Object	—
`project.ci_job_token_scope_enabled`	String	Example: `false`
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`

Property	Type	Description
`runner`	APIEntitiesCiRunner	—
`runner_manager`	APIEntitiesCiRunnerManager	—
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`tag_list`	Array of strings	Example: `["ubuntu18", "docker runner"]`
`user`	APIEntitiesUser	—
`web_url`	String	Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiJobArtifact

Property	Type	Description
`file_format`	String	Allowed values: `raw`, `zip`, `gzip` Example: `zip`
`file_type`	String	Allowed values: `archive`, `metadata`, `trace`, `junit`, `sast`, `dependency_scanning`, `container_scanning`, `dast`, `codequality`, `license_scanning`, `performance`, `metrics`, `metrics_referee`, `network_referee`, `lsif`, `dotenv`, `cobertura`, `terraform`, `accessibility`, `cluster_applications`, `secret_detection`, `requirements`, `coverage_fuzzing`, `browser_performance`, `load_performance`, `api_fuzzing`, `cluster_image_scanning`, `cyclonedx`, `requirements_v2`, `annotations`, `repository_xray`, `jacoco`, `sarif` Example: `archive`
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiJobArtifactFile

Property	Type	Description
`filename`	String	Example: `artifacts.zip`
`size`	Integer	Example: `1000`

APIEntitiesCiPipeline

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`
`tag`	Boolean	Example: `false`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiPipelineMergeRequest

Property	Type	Description
`iid`	Integer	Example: `14`
`title`	String	Example: `Add rate limiting to the public API`
`web_url`	String	Example: `https://gitlab.example.com/foo/bar/-/merge_requests/14`

APIEntitiesCiPipelineWithMetadata

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`name`	String	Example: `Build pipeline`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`

Property	Type	Description
`tag`	Boolean	Example: `false`
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCiUserPipeline

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2022-10-21T16:52:12.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`merge_request`	APIEntitiesCiPipelineMergeRequest	—
`name`	String	Example: `Build pipeline`
`project`	APIEntitiesProjectIdentity	—
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent queued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2022-10-21T16:49:50.000+02:00`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiVariable

Property	Type	Description
`description`	String	Example: `This variable is being used for ...`
`environment_scope`	String	Example: `*`
`hidden`	Boolean	—
`key`	String	Example: `TEST_VARIABLE_1`
`masked`	Boolean	—
`protected`	Boolean	—
`raw`	Boolean	—
`value`	String	Example: `TEST_1`
`variable_type`	String	Example: `env_var`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

DetailedStatusEntity

Property	Type	Description
`action`	Object	—
`action.button_title`	String	Example: `Cancel this job`
`action.confirmation_message`	String	Example: `Are you sure?`
`action.icon`	String	Example: `cancel`
`action.method`	String	Example: `post`
`action.path`	String	Example: `/namespace1/project1/-/jobs/2/cancel`
`action.title`	String	Example: `Cancel`
`details_path`	String	Example: `/test-group/test-project/-/pipelines/287`
`favicon`	String	Example: `/assets/ci_favicons/favicon_status_success.png`
`group`	String	Example: `success`
`has_details`	Boolean	Example: `true`
`icon`	String	Example: `status_success`
`illustration`	Object	Example: `{"content":"This job depends on upstream jobs that need to succeed in order for this job to be triggered","image":"illustrations/empty-state/empty-job-not-triggered-md.svg","size":"","title":"This job has not been triggered yet"}`
`label`	String	Example: `passed`
`text`	String	Example: `passed`
`tooltip`	String	Example: `passed`

TestCaseEntity

Property	Type	Description
`attachment_url`	String	Example: `http://localhost/namespace1/project1/-/jobs/1/artifacts/file/some/path.png`
`classname`	String	Example: `vulnerability_management_spec`
`execution_time`	Integer	Example: `180`
`file`	String	Example: `./spec/test_spec.rb`
`name`	String	Default: `(No name)` Example: `Security Reports can create an auto-remediation MR`
`recent_failures`	Object	Example: `{"base_branch":"develop","count":3}`
`severity`	String	Example: `low`
`stack_trace`	String	Example: `Failure/Error: is_expected.to eq(3)`
`status`	String	Example: `success`
`system_output`	String	Example: `Failure/Error: is_expected.to eq(3)`

TestReportEntity

Property	Type	Description
`error_count`	Integer	Example: `0`
`failed_count`	Integer	Example: `0`
`skipped_count`	Integer	Example: `0`
`success_count`	Integer	Example: `1`
`test_suites`	Array of TestSuiteEntity	—
`total_count`	Integer	Example: `1`
`total_time`	Integer	Example: `180`

TestReportSummaryEntity

Property	Type	Description
`test_suites`	TestSuiteSummaryEntity	—
`total`	Object	Example: `{"count":2,"error":0,"failed":0,"skipped":0,"success":2,"suite_error":null,"time":0.42}`

TestSuiteEntity

Property	Type	Description
`error_count`	Integer	Example: `0`
`failed_count`	Integer	Example: `0`
`name`	String	Example: `test`
`skipped_count`	Integer	Example: `12`
`success_count`	Integer	Example: `3351`
`suite_error`	String	Example: `JUnit XML parsing failed: 1:1: FATAL: Document is empty`
`test_cases`	Array of TestCaseEntity	—
`total_count`	Integer	Example: `3363`
`total_time`	Integer	Example: `1904`

TestSuiteSummaryEntity

Property	Type	Description
`build_ids`	Array of integers	Example: `[66004]`
`error_count`	Integer	Example: `0`
`failed_count`	Integer	Example: `0`
`name`	String	Example: `test`
`severity_counts`	Object	Example: `{"critical": {"errored":0,"failed":1}}`
`skipped_count`	Integer	Example: `12`
`success_count`	Integer	Example: `3351`
`suite_error`	String	Example: `JUnit XML parsing failed: 1:1: FATAL: Document is empty`
`test_cases`	Array of TestCaseEntity	—
`total_count`	Integer	Example: `3363`
`total_time`	Integer	Example: `1904`

Plan limits

Use this API to retrieve and change the application limits of a subscription plan.

The available plans depend on the GitLab edition. In the Community Edition, only the `default` plan exists. The Enterprise Edition has additional plans.

These endpoints require administrator access to the instance.

Get current plan limits

`GET /api/v4/application/plan_limits`

List the current limits of a plan on the GitLab instance.

Parameters

Name	Type	Description
`plan_name` Query	String	Name of the plan to get the limits from. Default: default Allowed values: `default` Default: `default`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPlanLimit
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—

Change plan limits

`PUT /api/v4/application/plan_limits`

Modify the limits of a plan on the GitLab instance.

Request body (`application/json`)

Property	Type	Description
`cargo_max_file_size`	Integer	Maximum Cargo package file size in bytes
`ci_active_jobs`	Integer	Total number of jobs in currently active pipelines
`ci_instance_level_variables`	Integer	Maximum number of Instance-level CI/CD variables that can be defined
`ci_needs_size_limit`	Integer	Maximum number of needs dependencies that a job can have
`ci_pipeline_schedules`	Integer	Maximum number of pipeline schedules
`ci_pipeline_size`	Integer	Maximum number of jobs in a single pipeline
`ci_project_subscriptions`	Integer	Maximum number of pipeline subscriptions to and from a project
`ci_registered_group_runners`	Integer	Maximum number of runners created or active in a group during the past seven days
`ci_registered_project_runners`	Integer	Maximum number of runners created or active in a project during the past seven days
`conan_max_file_size`	Integer	Maximum Conan package file size in bytes
`dotenv_size`	Integer	Maximum size of a dotenv artifact in bytes
`dotenv_variables`	Integer	Maximum number of variables in a dotenv artifact
`enforcement_limit`	Integer	Maximum storage size for the root namespace enforcement in MiB

Property	Type	Description
`generic_packages_max_file_size`	Integer	Maximum generic package file size in bytes
`helm_max_file_size`	Integer	Maximum Helm chart file size in bytes
`maven_max_file_size`	Integer	Maximum Maven package file size in bytes
`max_pipelines_per_merge_train`	Integer	Maximum number of parallel pipelines per merge train
`notification_limit`	Integer	Maximum storage size for the root namespace notifications in MiB
`npm_max_file_size`	Integer	Maximum NPM package file size in bytes
`nuget_max_file_size`	Integer	Maximum NuGet package file size in bytes
`pipeline_hierarchy_size`	Integer	Maximum number of downstream pipelines in a pipeline’s hierarchy tree
`plan_name` Required	String	Name of the plan to update Allowed values: `default` Minimum length: `1`
`pypi_max_file_size`	Integer	Maximum PyPI package file size in bytes
`service_desk_outbound_emails_per_day`	Integer	Maximum number of outbound Service Desk emails per day, per top-level namespace. 0 for unlimited
`service_desk_outbound_emails_per_hour`	Integer	Maximum number of outbound Service Desk emails per hour, per top-level namespace. 0 for unlimited
`storage_size_limit`	Integer	Maximum storage size for the root namespace in MiB

Property	Type	Description
`terraform_module_max_file_size`	Integer	Maximum Terraform Module package file size in bytes
`web_hook_calls`	Integer	Maximum number of times a webhook can be called per minute, per top-level namespace. 0 for unlimited
`web_hook_calls_low`	Integer	Maximum number of times a webhook can be called per minute, per top-level namespace. 0 for unlimited (Git Lab.com only)
`web_hook_calls_mid`	Integer	Maximum number of times a webhook can be called per minute, per top-level namespace. 0 for unlimited (Git Lab.com only)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPlanLimit
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesPlanLimit

Property	Type	Description
`cargo_max_file_size`	Integer	Example: `5.36870912e+09`
`ci_active_jobs`	Integer	Example: `0`
`ci_instance_level_variables`	Integer	Example: `25`
`ci_needs_size_limit`	Integer	Example: `50`
`ci_pipeline_schedules`	Integer	Example: `10`
`ci_pipeline_size`	Integer	Example: `0`
`ci_project_subscriptions`	Integer	Example: `2`
`ci_registered_group_runners`	Integer	Example: `1000`
`ci_registered_project_runners`	Integer	Example: `1000`
`conan_max_file_size`	Integer	Example: `3.221225472e+09`
`dotenv_size`	Integer	Example: `5120`
`dotenv_variables`	Integer	Example: `20`
`enforcement_limit`	Integer	Example: `15000`
`generic_packages_max_file_size`	Integer	Example: `5.36870912e+09`
`helm_max_file_size`	Integer	Example: `5.24288e+06`
`limits_history`	Object	Example: `{"enforcement_limit":[{"timestamp":1686909124,"user_id":1,"username":"x","value":5}],"notification_limit":[{"timestamp":1686909124,"user_id":2,"username":"y","value":7}]}`
`maven_max_file_size`	Integer	Example: `3.221225472e+09`
`max_pipelines_per_merge_train`	Integer	Example: `20`

Property	Type	Description
`notification_limit`	Integer	Example: `15000`
`npm_max_file_size`	Integer	Example: `5.24288e+08`
`nuget_max_file_size`	Integer	Example: `5.24288e+08`
`pipeline_hierarchy_size`	Integer	Example: `1000`
`pypi_max_file_size`	Integer	Example: `3.221225472e+09`
`service_desk_outbound_emails_per_day`	Integer	Example: `700`
`service_desk_outbound_emails_per_hour`	Integer	Example: `100`
`storage_size_limit`	Integer	Example: `15000`
`terraform_module_max_file_size`	Integer	Example: `1.073741824e+09`
`web_hook_calls`	Integer	Example: `500`
`web_hook_calls_low`	Integer	Example: `500`
`web_hook_calls_mid`	Integer	Example: `500`

Project import

Use this API to import repositories from external sources and to migrate a project between GitLab instances.

If you first migrate the parent group structure with the group import and export API, you can preserve relationships that span the group, such as connections between project issues and group epics. After an import, use the project-level CI/CD variables API to restore CI/CD variables. You must still migrate your container registry over a series of Docker pulls and pushes, and re-run any CI/CD pipelines to retrieve build artifacts.

The project relations export endpoints are used by the destination instance during group migration by direct transfer to migrate a project structure, and you don't usually need to call them yourself. In this context, a relation is an exportable item such as a merge request, including any related items such as a label.

> User contribution mapping is not supported when you import projects to a personal namespace. All contributions are assigned to the personal namespace owner and cannot be reassigned.

Import repository from Bitbucket Cloud

`POST /api/v4/import/bitbucket`

Imports a repository from Bitbucket Cloud to GitLab. Prerequisites: - The prerequisites for Bitbucket Cloud importer. This feature was introduced in GitLab 17.0.

Request body (`application/json`)

Property	Type	Description
`bitbucket_api_token` Required	String	Bitbucket Cloud API token
`bitbucket_email` Required	String	Bitbucket Cloud email
`new_name`	String	Name of the new project. Also used as the new path, so it must not start or end with a special character and must not contain consecutive special characters
`repo_path` Required	String	Path to the repository
`target_namespace` Required	String	Namespace to import the repository into. Supports subgroups, for example `/namespace/subgroup`

Responses

Code	Description	Schema
`201`	Created	ProjectImportEntity
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`422`	Unprocessable entity	—
`503`	Service unavailable	—

Import repository from Bitbucket Server

`POST /api/v4/import/bitbucket_server`

Imports a repository from Bitbucket Server to GitLab. The Bitbucket Project Key is only used for finding the repository in Bitbucket. You must specify a `target_namespace` if you want to import the repository to a GitLab group.

Request body (`application/json`)

Property	Type	Description
`bitbucket_server_project` Required	String	Key of the Bitbucket Server project. Must contain only letters, numbers, hyphens, underscores, periods, or whitespace characters. Personal project keys start with `~`
`bitbucket_server_repo` Required	String	Name of the Bitbucket Server repository. Must contain only letters, numbers, hyphens, underscores, periods, or whitespace characters
`bitbucket_server_url` Required	String	URL of the Bitbucket Server instance
`bitbucket_server_username` Required	String	Bitbucket Server username
`new_name`	String	Name of the new project. Also used as the new path, so it must not start or end with a special character and must not contain consecutive special characters
`new_namespace`	String	Namespace to import the repository into
`personal_access_token` Required	String	Bitbucket Server personal access token or password
`timeout_strategy`	String	Strategy for handling import timeouts. `optimistic` continues to the next stage of the import. `pessimistic` fails the import immediately. Defaults to `pessimistic` Allowed values: `optimistic`, `pessimistic`

Responses

Code	Description	Schema
`201`	Created	ProjectEntity
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`422`	Unprocessable entity	—
`503`	Service unavailable	—

Import a repository from GitHub

`POST /api/v4/import/github`

Imports a repository from GitHub to GitLab.

Request body (`application/json`)

Property	Type	Description
`github_hostname`	String	Custom GitHub Enterprise hostname. Do not set for Git Hub.com. From GitLab 16.5 to GitLab 17.1, you must include the path `/api/v3`
`new_name`	String	Name of the new project. Also used as the new path, so it must not start or end with a special character and must not contain consecutive special characters
`optional_stages`	Object	Additional items to import. For a list of items, see select additional items to import
`pagination_limit`	Integer	Number of items retrieved per API request to GitHub. If omitted, defaults to `100`. Use a lower number for large repositories to reduce the risk of GitHub API errors, though this increases migration time Maximum: `100` Minimum: `1`
`personal_access_token` Required	String	GitHub personal access token
`repo_id` Required	Integer	GitHub repository ID
`target_namespace` Required	String	Namespace to import the repository into. Supports subgroups, for example `/namespace/subgroup`. Must not be blank
`timeout_strategy`	String	Strategy for handling import timeouts. `optimistic` continues to the next stage of the import. `pessimistic` fails

Property	Type	Description
		the import immediately. Defaults to `pessimistic` Allowed values: `optimistic`, `pessimistic`

Responses

Code	Description	Schema
`201`	Created	ProjectEntity
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`422`	Unprocessable entity	—
`503`	Service unavailable	—

Cancel a GitHub project import

`POST /api/v4/import/github/cancel`

Cancels an in-progress import of a GitHub project to GitLab.

Request body (`application/json`)

Property	Type	Description
`project_id` Required	Integer	ID of the project whose import to cancel

Responses

Code	Description	Schema
`200`	OK	ProjectImportEntity
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Import a project from a local archive

`POST /api/v4/projects/import`

Imports a project from a local archive.

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The project export file to be imported
`file.etag`	String	Etag of the file (generated by Workhorse)
`file.md5`	String	MD5 checksum of the file (generated by Workhorse)
`file.name`	String	Real filename as send in Content-Disposition (generated by Workhorse)
`file.path`	String	Path to locally stored body (generated by Workhorse)
`file.remote_id`	String	Remote_id of the file (generated by Workhorse)
`file.remote_url`	String	Remote_url of the file (generated by Workhorse)
`file.sha1`	String	SHA1 checksum of the file (generated by Workhorse)
`file.sha256`	String	SHA256 checksum of the file (generated by Workhorse)
`file.size`	Integer	Real size of file (generated by Workhorse)
`file.type`	String	Real content type as send in Content-Type (generated by Workhorse)
`name`	String	The name of the project to be imported. Defaults to the path of the project if not provided
`namespace`	String	(deprecated) The ID or path of the namespace to import the project to. Defaults to the current user’s namespace. Mutually exclusive with

Property	Type	Description
		`namespace_id`, `namespace_path`
`namespace_id`	Integer	The ID of the namespace that the project will be imported into. Defaults to the current user's namespace. Mutually exclusive with `namespace`, `namespace_path` Example: `1`
`namespace_path`	String	The path of the namespace that the project will be imported into. Defaults to the current user's namespace. Mutually exclusive with `namespace`, `namespace_id` Example: `new_path/gitlab`
`override_params`	Object	New project params to override values in the export
`override_params.allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped
`override_params.analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.auto_cancel_pending_pipelines`	String	Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`override_params.auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`override_params.auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
	Boolean	

Property	Type	Description
`override_params.autoclose_referenced_issues`		Flag indication if referenced issues auto-closing is enabled
`override_params.avatar`	String (binary)	Avatar image for project
`override_params.build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`override_params.build_timeout`	Integer	Build timeout
`override_params.builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`override_params.cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project
`override_params.container_expiration_policy_attributes`	Object	Object that contains information on the container expiration policy
`override_params.container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
`override_params.container_expiration_policy_attributes.enabled`	Boolean	Flag indication if container expiration policy is enabled
`override_params.container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`override_params.container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal

Property	Type	Description
`override_params.container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`override_params.container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`override_params.container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry Allowed values: `disabled`, `private`, `enabled`
`override_params.container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container registry is enabled for that project
`override_params.description`	String	The description of the project
`override_params.emails_disabled`	Boolean	Deprecated: Use emails_enabled instead
`override_params.emails_enabled`	Boolean	Enable email notifications
`override_params.enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`override_params.environments_access_level`	String	Environments access level. One of `disabled`, `private` or

Property	Type	Description
		`enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`override_params.infrastructure_access_level`	String	Infrastructure access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.issue_branch_template`	String	Template used to create a branch from an issue
`override_params.issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`override_params.jobs_enabled`	Boolean	Flag indication if jobs are enabled
`override_params.lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
	String	

Property	Type	Description
`override_params.merge_commit_template`		Template used to create merge commit message
`override_params.merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`override_params.merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`override_params.model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`override_params.mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`

Property	Type	Description
`override_params.only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`override_params.only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`override_params.package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`, `enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.packages_enabled`	Boolean	Deprecated: Use :package_registry_access_level instead. Enable project packages feature
`override_params.pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public` Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.printing_merge_request_link_enabled`	Boolean	Show link to create/view merge request when pushing from the command line
`override_params.public_builds`	Boolean	Deprecated: Use public_jobs instead

Property	Type	Description
`override_params.public_jobs`	Boolean	Perform public builds
`override_params.releases_access_level`	String	Releases access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`override_params.repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.repository_storage`	String	Which storage shard the repository is on. Available only to admins
`override_params.request_access_enabled`	Boolean	Allow users to request member access
`override_params.resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
`override_params.resource_group_default_process_mode`	String	The process mode of the resource group Allowed values: `unordered`, `oldest_first`, `newest_first`, `newest_ready_first`
`override_params.security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`

Property	Type	Description
`override_params.service_desk_enabled`	Boolean	Disable or enable the service desk
`override_params.shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`override_params.show_default_award_emojis`	Boolean	Show default award emojis
`override_params.show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`override_params.snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.snippets_enabled`	Boolean	Flag indication if snippets are enabled
`override_params.squash_commit_template`	String	Template used to create squash commit message
`override_params.squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`
`override_params.suggestion_commit_message`	String	The commit message used to apply merge request suggestions
`override_params.tag_list`	Array of strings	Deprecated: Use :topics instead
`override_params.topics`	Array of strings	The list of topics for a project
`override_params.visibility`	String	

Property	Type	Description
		The visibility of the project Allowed values: `private`, `internal`, `public`
`override_params.warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters
`override_params.wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.wiki_enabled`	Boolean	Flag indication if the wiki is enabled
`overwrite`	Boolean	If there is a project in the same namespace and with the same name overwrite it Default: `false`
`path` Required	String	The new project path and name

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Import project resources

`POST /api/v4/projects/import-relation`

Imports project resources included with a project archive. The type of item to import is controlled by the `relation` attribute. Skips items that were previously imported. This feature was introduced in GitLab 16.11.

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The project export file from which to extract the relation
`file.etag`	String	Etag of the file (generated by Workhorse)
`file.md5`	String	MD5 checksum of the file (generated by Workhorse)
`file.name`	String	Real filename as sent in Content-Disposition (generated by Workhorse)
`file.path`	String	Path to locally stored body (generated by Workhorse)
`file.remote_id`	String	Remote_id of the file (generated by Workhorse)
`file.remote_url`	String	Remote_url of the file (generated by Workhorse)
`file.sha1`	String	SHA1 checksum of the file (generated by Workhorse)
`file.sha256`	String	SHA256 checksum of the file (generated by Workhorse)
`file.size`	Integer	Real size of file (generated by Workhorse)
`file.type`	String	Real content type as send in Content-Type (generated by Workhorse)
`path` Required	String	The project path and name
`relation` Required	String	The relation to import Allowed values: `issues`, `merge_requests`, `milestones`, `ci_pipelines` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesRelationImportTracker
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Authorize project relation import

`POST /api/v4/projects/import-relation/authorize`

Authorizes uploading a project relation import file. This feature was introduced in GitLab 16.11.

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Workhorse authorize the project import upload

`POST /api/v4/projects/import/authorize`

This feature was introduced in GitLab 12.9

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Import a project from a remote archive

`POST /api/v4/projects/remote-import`

Imports a project from a remote archive.

Request body (`multipart/form-data`)

Property	Type	Description
`name`	String	The name of the project to be imported. Defaults to the path of the project if not provided
`namespace`	String	(deprecated) The ID or path of the namespace to import the project to. Defaults to the current user's namespace. Mutually exclusive with `namespace_id`, `namespace_path`
`namespace_id`	Integer	The ID of the namespace that the project will be imported into. Defaults to the current user's namespace. Mutually exclusive with `namespace`, `namespace_path` Example: `1`
`namespace_path`	String	The path of the namespace that the project will be imported into. Defaults to the current user's namespace. Mutually exclusive with `namespace`, `namespace_id` Example: `new_path/gitlab`
`override_params`	Object	New project params to override values in the export
`override_params.allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped
`override_params.analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.auto_cancel_pending_pipelines`	String	

Property	Type	Description
		Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`override_params.auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`override_params.auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
`override_params.autoclose_referenced_issues`	Boolean	Flag indication if referenced issues auto-closing is enabled
`override_params.avatar`	String (binary)	Avatar image for project
`override_params.build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`override_params.build_timeout`	Integer	Build timeout
`override_params.builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`override_params.cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project
`override_params.container_expiration_policy_attributes`	Object	Object that contains information on the container expiration policy
`override_params.container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
	Boolean	

Property	Type	Description
`override_params.container_expiration_policy_attributes.enabled`		Flag indication if container expiration policy is enabled
`override_params.container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`override_params.container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal
`override_params.container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`override_params.container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`override_params.container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry Allowed values: `disabled`, `private`, `enabled`
`override_params.container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container registry is enabled for that project
`override_params.description`	String	The description of the project

Property	Type	Description
`override_params.emails_disabled`	Boolean	Deprecated: Use emails_enabled instead
`override_params.emails_enabled`	Boolean	Enable email notifications
`override_params.enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`override_params.environments_access_level`	String	Environments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`override_params.infrastructure_access_level`	String	Infrastructure access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.issue_branch_template`	String	Template used to create a branch from an issue
`override_params.issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled`

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`
`override_params.issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`override_params.jobs_enabled`	Boolean	Flag indication if jobs are enabled
`override_params.lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
`override_params.merge_commit_template`	String	Template used to create merge commit message
`override_params.merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`override_params.merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`override_params.model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled`

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`
`override_params.mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`override_params.mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`
`override_params.only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`override_params.only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`override_params.package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`, `enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.packages_enabled`	Boolean	Deprecated: Use :package_registry_access_level instead. Enable project packages feature
`override_params.pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public`

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.printing_merge_request_link_enabled`	Boolean	Show link to create/view merge request when pushing from the command line
`override_params.public_builds`	Boolean	Deprecated: Use public_jobs instead
`override_params.public_jobs`	Boolean	Perform public builds
`override_params.releases_access_level`	String	Releases access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`override_params.repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.repository_storage`	String	Which storage shard the repository is on. Available only to admins
`override_params.request_access_enabled`	Boolean	Allow users to request member access
`override_params.resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
`override_params.resource_group_default_process_mode`	String	The process mode of the resource group Allowed values: `unordered`, `oldest_first`,

Property	Type	Description
		`newest_first`, `newest_ready_first`
`override_params.security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.service_desk_enabled`	Boolean	Disable or enable the service desk
`override_params.shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`override_params.show_default_award_emojis`	Boolean	Show default award emojis
`override_params.show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`override_params.snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.snippets_enabled`	Boolean	Flag indication if snippets are enabled
`override_params.squash_commit_template`	String	Template used to create squash commit message
`override_params.squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`
	String	

Property	Type	Description
`override_params.suggestion_commit_message`		The commit message used to apply merge request suggestions
`override_params.tag_list`	Array of strings	Deprecated: Use :topics instead
`override_params.topics`	Array of strings	The list of topics for a project
`override_params.visibility`	String	The visibility of the project Allowed values: `private`, `internal`, `public`
`override_params.warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters
`override_params.wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.wiki_enabled`	Boolean	Flag indication if the wiki is enabled
`overwrite`	Boolean	If there is a project in the same namespace and with the same name overwrite it Default: `false`
`path` Required	String	The new project path and name
`url` Required	String	The URL for the file

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—
`503`	Service unavailable	—

Import a project from an AWS S3 bucket

`POST /api/v4/projects/remote-import-s3`

Imports a project from an archive stored in a specified AWS S3 bucket.

Request body (`multipart/form-data`)

Property	Type	Description
`access_key_id` Required	String	Access key id
`bucket_name` Required	String	Bucket name
`file_key` Required	String	File key
`name`	String	The name of the project to be imported. Defaults to the path of the project if not provided
`namespace`	String	(deprecated) The ID or path of the namespace to import the project to. Defaults to the current user’s namespace. Mutually exclusive with `namespace_id`, `namespace_path`
`namespace_id`	Integer	The ID of the namespace that the project will be imported into. Defaults to the current user’s namespace. Mutually exclusive with `namespace`, `namespace_path` Example: `1`
`namespace_path`	String	The path of the namespace that the project will be imported into. Defaults to the current user’s namespace. Mutually exclusive with `namespace`, `namespace_id` Example: `new_path/gitlab`
`override_params`	Object	New project params to override values in the export
`override_params.allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped

Property	Type	Description
`override_params.analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.auto_cancel_pending_pipelines`	String	Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`override_params.auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`override_params.auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
`override_params.autoclose_referenced_issues`	Boolean	Flag indication if referenced issues auto-closing is enabled
`override_params.avatar`	String (binary)	Avatar image for project
`override_params.build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`override_params.build_timeout`	Integer	Build timeout
`override_params.builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`override_params.cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project
	Object	

Property	Type	Description
`override_params.container_expiration_policy_attributes`		Object that contains information on the container expiration policy
`override_params.container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
`override_params.container_expiration_policy_attributes.enabled`	Boolean	Flag indication if container expiration policy is enabled
`override_params.container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`override_params.container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal
`override_params.container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`override_params.container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`override_params.container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry Allowed values: `disabled`, `private`, `enabled`

Property	Type	Description
`override_params.container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container registry is enabled for that project
`override_params.description`	String	The description of the project
`override_params.emails_disabled`	Boolean	Deprecated: Use emails_enabled instead
`override_params.emails_enabled`	Boolean	Enable email notifications
`override_params.enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`override_params.environments_access_level`	String	Environments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`override_params.infrastructure_access_level`	String	Infrastructure access level. One of `disabled`, `private` or `enabled`

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`
`override_params.issue_branch_template`	String	Template used to create a branch from an issue
`override_params.issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`override_params.jobs_enabled`	Boolean	Flag indication if jobs are enabled
`override_params.lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
`override_params.merge_commit_template`	String	Template used to create merge commit message
`override_params.merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`override_params.merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`override_params.model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`

Property	Type	Description
`override_params.model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`override_params.mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`
`override_params.only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`override_params.only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`override_params.package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`, `enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.packages_enabled`	Boolean	Deprecated: Use :package_registry_access_level instead. Enable project packages feature
`override_params.pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public` Allowed values: `disabled`, `private`, `enabled`, `public`
`override_params.printing_merge_request_link_enabled`	Boolean	Show link to create/view merge request when pushing from the command line
`override_params.public_builds`	Boolean	Deprecated: Use public_jobs instead
`override_params.public_jobs`	Boolean	Perform public builds
`override_params.releases_access_level`	String	Releases access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`override_params.repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.repository_storage`	String	Which storage shard the repository is on. Available only to admins
	Boolean	

Property	Type	Description
`override_params.request_access_enabled`		Allow users to request member access
`override_params.resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
`override_params.resource_group_default_process_mode`	String	The process mode of the resource group Allowed values: `unordered`, `oldest_first`, `newest_first`, `newest_ready_first`
`override_params.security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.service_desk_enabled`	Boolean	Disable or enable the service desk
`override_params.shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`override_params.show_default_award_emojis`	Boolean	Show default award emojis
`override_params.show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`override_params.snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.snippets_enabled`	Boolean	Flag indication if snippets are enabled

Property	Type	Description
`override_params.squash_commit_template`	String	Template used to create squash commit message
`override_params.squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`
`override_params.suggestion_commit_message`	String	The commit message used to apply merge request suggestions
`override_params.tag_list`	Array of strings	Deprecated: Use :topics instead
`override_params.topics`	Array of strings	The list of topics for a project
`override_params.visibility`	String	The visibility of the project Allowed values: `private`, `internal`, `public`
`override_params.warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters
`override_params.wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`override_params.wiki_enabled`	Boolean	Flag indication if the wiki is enabled
`overwrite`	Boolean	If there is a project in the same namespace and with the same name overwrite it Default: `false`
`path` Required	String	The new project path and name

Property	Type	Description
`region` Required	String	AWS region
`secret_access_key` Required	String	Secret access key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—
`503`	Service unavailable	—

Retrieve the status of a project export

`GET /api/v4/projects/{id}/export`

Retrieves the status of the most recent export for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectExportStatus
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Export a project

`POST /api/v4/projects/{id}/export`

Exports a project. Use the `upload` hash parameter to upload the exported project to a web server or any S3-compatible platform.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`description`	String	Overrides the project description
`excluded_relations`	Array of strings	Project relation names to exclude from the export, for example `merge_requests` and `issues`
`upload`	Object	Details for uploading the exported project to a web server
`upload.http_method`	String	HTTP method to upload the exported project Allowed values: `PUT`, `POST` Default: `PUT`
`upload.url`	String	URL to upload the exported project to

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—
`503`	Service unavailable	—

Download a project export

`GET /api/v4/projects/{id}/export/download`

Downloads the most recent export of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Schedule a relations export for a project

`POST /api/v4/projects/{id}/export_relations`

Schedules a relations export for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`batched`	Boolean	If `true` , exports in batches

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Download a relations export for a project

`GET /api/v4/projects/{id}/export_relations/download`

Downloads a project relations export file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`relation` Query, required	String	Name of the project top-level relation to download or view
`batched` Query	Boolean	If `true`, downloads the batch specified by `batch_number`. Set to `true` only for batched exports. All or none of `batched`, `batch_number` must be provided
`batch_number` Query	Integer	Number of the export batch to download. All or none of `batched`, `batch_number` must be provided

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/gzip`) String (binary) (`application/octet-stream`)
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`500`	Internal Server Error	—
`503`	Service unavailable	—

Retrieve the status of an relations export for a project

`GET /api/v4/projects/{id}/export_relations/status`

Retrieves the status of a relations export for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`relation` Query	String	Name of the project top-level relation to download or view

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBulkImports ExportStatus
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Receive a GitHub webhook delivery for an OAuth-connected repository

`POST /api/v4/projects/{id}/github/webhooks`

Verifies the webhook signature and, once available, triggers repository reconciliation.

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`404`	Not found	—
`429`	Too many requests	—

Retrieve the status of a project import

`GET /api/v4/projects/{id}/import`

Retrieves the status of the most recent import for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Import a project from a Git URL

`POST /api/v4/projects/{id}/import/git`

Imports a project from a specified Git URL. This feature was introduced in GitLab 18.10.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`import_url` Required	String	URL from which the project is imported
`import_url_password`	String	Password for the import URL
`import_url_user`	String	Username for the import URL

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`422`	Unprocessable Entity	—

Retrieve the status of a project resource import

`GET /api/v4/projects/{id}/relation-imports`

Retrieves the status of the most recent relation import for a specified project. Because only one relation import can be scheduled at a time, you can use this endpoint to check whether the previous import completed successfully. This feature was introduced in GitLab 16.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectImportStatus
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`503`	Service unavailable	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBulkImportsExportBatchStatus

Property	Type	Description
`batch_number`	Integer	Example: `1`
`error`	String	Example: `Error message`
`objects_count`	Integer	Example: `100`
`status`	String	Allowed values: `started`, `finished`, `failed` Example: `started`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

APIEntitiesBulkImportsExportStatus

Property	Type	Description
`batched`	Boolean	Example: `true`
`batches`	APIEntitiesBulkImportsExportBatchStatus	—
`batches_count`	Integer	Example: `2`
`error`	String	Example: `Error message`
`relation`	String	Example: `issues`
`status`	String	Allowed values: `pending`, `started`, `finished`, `failed` Example: `started`
`total_objects_count`	Integer	Example: `100`
`updated_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`

APIEntitiesProjectExportStatus

Property	Type	Description
`_links`	Object	—
`_links.api_url`	String	Example: `https://gitlab.example.com/api/v4/projects/1/export/download`
`_links.web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-test/download_export`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`export_status`	String	Allowed values: `queued`, `started`, `finished`, `failed` Example: `finished`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesProjectImportFailedRelation

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`exception_class`	String	Example: `StandardError`
`exception_message`	String	—
`id`	Integer (int64)	Example: `1`
`line_number`	Integer	Example: `1`
`relation_name`	String	Example: `issues`
`source`	String	Example: `ImportRepositoryWorker`

APIEntitiesProjectImportStatus

Property	Type	Description
`correlation_id`	String	Example: `dfcf583058ed4508e4c7c617bd7f0edd`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`failed_relations`	Array of APIEntitiesProjectImportFailedRelation	—
`id`	Integer (int64)	Example: `1`
`import_error`	String	Example: `Error message`
`import_status`	String	Example: `scheduled`
`import_type`	String	Example: `gitlab_project`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`stats`	Object	—

APIEntitiesRelationImportTracker

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`id`	Integer (int64)	Example: `1`
`project_path`	String	Example: `namespace1/project1`
`relation`	String	Example: `issues`
`status`	String	Example: `pending`
`updated_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`

ProjectEntity

Property	Type	Description
`forked`	Boolean	Example: `true`
`full_name`	String	Example: `GitLab Org / GitLab`
`full_path`	String	Example: `gitlab-org/gitlab`
`id`	Integer	Example: `1`
`name`	String	Example: `GitLab`
`refs_url`	String	—

ProjectImportEntity

Property	Type	Description
`forked`	Boolean	Example: `true`
`full_name`	String	Example: `GitLab Org / GitLab`
`full_path`	String	Example: `gitlab-org/gitlab`
`human_import_status_name`	String	Example: `canceled`
`id`	Integer	Example: `1`
`import_error`	String	—
`import_source`	String	Example: `source/source-repo`
`import_status`	String	Allowed values: `scheduled`, `started`, `finished`, `failed`, `canceled` Example: `scheduled`
`import_warning`	String	—
`name`	String	Example: `GitLab`
`provider_link`	String	Example: `/source/source-repo`
`refs_url`	String	—
`relation_type`	String	—

Project snapshots

Use this API to download a snapshot of a project's repository, for use in salvaging a corrupt repository.

These endpoints require administrator access to the instance.

Download snapshot of a Git repository

`GET /api/v4/projects/{id}/snapshot`

Downloads snapshot of a Git repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`wiki` Query	Boolean	If `true`, downloads the wiki repository instead of the project repository

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/x-tar`)
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not Found	—

Project templates

Use this API to retrieve project-specific templates, in addition to the templates common to the entire instance:

- Dockerfile templates
- Gitignore templates
- GitLab CI/CD configuration templates
- Open source license templates
- Issue and merge request templates

These endpoints supersede the instance template endpoints, which are scheduled for removal in API version 5. File templates for groups are also supported.

List all templates of a particular type

`GET /api/v4/projects/{id}/templates/{type}`

Lists all templates of a specified type for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`type` Path, required	String	Type of the template Allowed values: `dockerfiles`, `gitignores`, `gitlab_ci_ymls`, `licenses`, `issues`, `merge_requests` Minimum length: `1`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesTemplatesList
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a template of a particular type

`GET /api/v4/projects/{id}/templates/{type}/{name}`

Retrieves a template of a specified type for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`type` Path, required	String	Type of the template Allowed values: `dockerfiles`, `gitignores`, `gitlab_ci_ymls`, `licenses`, `issues`, `merge_requests` Minimum length: `1`
`name` Path, required	String	Key of the template, as obtained from the collection endpoint Example: `MIT`
`source_template_project_id` Query	Integer	ID of the project where the template is stored. Useful when multiple templates from different projects have the same name. If omitted, the match from the closest ancestor is returned Example: `1`
`project` Query	String	Project name to use when expanding placeholders in the template. Affects only licenses Example: `GitLab`
`fullname` Query	String	Full name of the copyright holder to use when expanding placeholders in the template. Affects only licenses Example: `GitLab B.V.`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesLicense
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesLicense

Property	Type	Description
`conditions`	Array of strings	Example: `["include-copyright"]`
`content`	String	Example: `GNU GENERAL PUBLIC LICENSE`
`description`	String	Example: `A simple license`
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`limitations`	Array of strings	Example: `["liability"]`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`permissions`	Array of strings	Example: `["commercial-use"]`
`popular`	Boolean	—
`source_url`	String	—

APIEntitiesTemplatesList

Property	Type	Description
`key`	String	Example: `mit`
`name`	String	Example: `MIT License`

Project topics

Use this API to interact with project topics. For more information, see project topics.

List all topics

`GET /api/v4/topics`

Lists all project topics sorted by the number of associated projects.

Parameters

Name	Type	Description
`search` Query	String	Search topics against their `name` Example: `search`
`without_projects` Query	Boolean	If `true`, returns only topics that have no assigned projects
`organization_id` Query	Integer	Filter topics by organization ID
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsTopic
`400`	Bad Request	—

Create a project topic

`POST /api/v4/topics`

Creates a project topic. Administrators only.

Request body (`multipart/form-data`)

Property	Type	Description
`avatar`	String (binary)	Avatar image for topic
`description`	String	Description
`name` Required	String	Slug (name)
`organization_id`	Integer	The organization id for the topic
`title` Required	String	Title

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsTopic
`400`	Bad Request	—

Merge topics

`POST /api/v4/topics/merge`

Merges a source topic into a target topic. This action deletes the source topic and moves all assigned projects to the target topic. Administrators only.

Request body (`application/json`)

Property	Type	Description
`source_topic_id` Required	Integer	ID of the project topic to merge from
`target_topic_id` Required	Integer	ID of the project topic to merge into

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsTopic
`400`	Bad Request	—

Retrieve a topic

`GET /api/v4/topics/{id}`

Retrieves a specified project topic.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project topic

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsTopic
`400`	Bad Request	—
`404`	Not Found	—

Update a project topic

`PUT /api/v4/topics/{id}`

Updates a specified project topic. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project topic

Request body (`multipart/form-data`)

Property	Type	Description
`avatar`	String (binary)	Avatar image for topic
`description`	String	Description
`name`	String	Slug (name)
`title`	String	Title

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsTopic
`400`	Bad Request	—
`404`	Not Found	—

Delete a project topic

`DELETE /api/v4/topics/{id}`

Deletes a specified project topic. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project topic

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesProjectsTopic

Property	Type	Description
`avatar_url`	String	Example: `http://gitlab.example.com/uploads/topic/avatar/1/avatar.png`
`description`	String	Example: `A description`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `topic1`
`organization_id`	Integer (int64)	Example: `1`
`title`	String	Example: `Topic 1`
`total_projects_count`	Integer	Example: `1`

Projects

Use this API to manage projects and the resources they contain, including:

- Project settings, visibility, access permissions, and security settings.
- Archiving, transferring, forking, and starring projects.
- Project issues, issue statistics, and project statistics.
- Markdown uploads referenced in issues, merge requests, snippets, or wiki pages.
- Protection rules for container image tags, container repositories, and packages.

If a user is not a member of a private project, a `GET` request on that project results in a `404` status code.

List CODEOWNERS entries that match a file path

`GET /api/v4/code-owners-for-file`

Parameters

Name	Type	Description
`project_path` Query, required	String	—
`file_path` Query, required	String	—
`ref` Query	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

List all projects

`GET /api/v4/projects`

Lists all projects. Unauthenticated requests return only public projects with a limited subset of attributes. You can filter responses by custom attributes. To use the `updated_after` or `updated_before` filters, you must also set `order_by` to `updated_at`.

Parameters

Name	Type	Description
`order_by` Query	String	Return projects ordered by field. storage_size, repository_size, wiki_size, packages_size are only available to admins. Similarity is available when searching and is limited to projects the user has access to Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count`, `storage_size`, `repository_size`, `wiki_size`, `packages_size` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`archived` Query	Boolean	If `true`, returns only archived projects
`visibility` Query	String	Return projects with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return a list of projects with a `path`, `name`, or `description` matching the search criteria (case-insensitive, substring match). Multiple terms can be provided, separated by an escaped space, either `+` or `%20`, and are ANDed together. Example: `one+two` matches substrings `one` and `two` (in any order)
	Boolean	

Name	Type	Description
`search_namespaces` Query		If `true`, includes ancestor namespaces when matching search criteria
`owned` Query	Boolean	If `true`, returns only projects explicitly owned by the current user Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`imported` Query	Boolean	If `true`, returns only projects that were imported from external systems by the authenticated user Default: `false`
`membership` Query	Boolean	If `true`, returns only projects that the current user is a member of Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`with_programming_language` Query	String	Return projects that use the given programming language
`min_access_level` Query	Integer	Return results where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`

Name	Type	Description
`id_after` Query	Integer	Return projects with IDs greater than the specified ID
`id_before` Query	Integer	Return projects with IDs less than the specified ID
`last_activity_after` Query	String (date-time)	Return projects with last activity after the specified time
`last_activity_before` Query	String (date-time)	Return projects with last activity before the specified time
`repository_storage` Query	String	Return projects stored on the specified repository storage shard. Administrators only
`topic` Query	Array of strings	Comma-separated list of topics. Return projects having all topics
`topic_id` Query	Integer	Filter projects by the given topic ID
`updated_before` Query	String (date-time)	Return projects last updated on or before the specified time
`updated_after` Query	String (date-time)	Return projects last updated on or after the specified time
`include_pending_delete` Query	Boolean	If `true`, includes projects that are pending deletion in the response. Administrators only
`marked_for_deletion_on` Query	String (date)	Return projects that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only projects that are not archived and not marked for deletion
`page` Query	Integer	Page number of results to retrieve

Name	Type	Description
		Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`statistics` Query	Boolean	If `true`, includes project statistics. Available only to users with the Reporter, Developer, Maintainer, or Owner role Default: `false`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBasic ProjectDetails
`400`	Bad request	—

Create a project

`POST /api/v4/projects`

Creates a project owned by the authenticated user. If your HTTP repository is not publicly accessible, add authentication information to the URL `https://username:password@gitlab.company.com/group/project.git`, where `password` is a public access key with the `api` scope.

Request body (`multipart/form-data`)

Property	Type	Description
`allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped
`analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`auto_cancel_pending_pipelines`	String	Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
`autoclose_referenced_issues`	Boolean	Flag indication if referenced issues auto-closing is enabled
`avatar`	String (binary)	Avatar image for project
`build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`build_timeout`	Integer	Build timeout
`builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project

Property	Type	Description
`container_expiration_policy_attributes`	Object	Object that contains information on the container expiration policy
`container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
`container_expiration_policy_attributes.enabled`	Boolean	Flag indication if container expiration policy is enabled
`container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal
`container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry Allowed values: `disabled`, `private`, `enabled`
`container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container

Property	Type	Description
		registry is enabled for that project
`default_branch`	String	The default branch of the project Example: `main`
`description`	String	The description of the project
`emails_disabled`	Boolean	Deprecated: Use emails_ enabled instead
`emails_enabled`	Boolean	Enable email notifications
`enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`environments_access_level`	String	Environments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`import_url`	String	URL from which the project is imported. Mutually exclusive with `template_name`, `template_project_id`
	String	Infrastructure access level. One of `disabled`, `private` or

Property	Type	Description
`infrastructure_access_level`		`enabled` Allowed values: `disabled`, `private`, `enabled`
`initialize_with_readme`	Boolean	Initialize a project with a README.md
`issue_branch_template`	String	Template used to create a branch from an issue
`issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`jobs_enabled`	Boolean	Flag indication if jobs are enabled
`lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
`merge_commit_template`	String	Template used to create merge commit message
`merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`merge_request_title_regex`	String	Regex pattern for validating merge request titles. All or none of `merge_request_title_regex`, `merge_request_title_regex_description` must be provided Maximum length: `255`
`merge_request_title_regex_description`	String	Example merge request title that matches merge_request_title_regex. Shown when a title

Property	Type	Description
		does not match. All or none of `merge_request_title_regex`, `merge_request_title_regex_description` must be provided Maximum length: `255`
`merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`
`name`	String	The name of the project. At least one of `name`, `path` must

Property	Type	Description
		be provided Example: `New Project`
`namespace_id`	Integer	Namespace ID for the new project. Default to the user namespace
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`, `enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry Allowed values: `disabled`, `private`, `enabled`, `public`
`packages_enabled`	Boolean	Deprecated: Use :package_registry_access_level instead. Enable project packages feature
`pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public` Allowed values: `disabled`, `private`, `enabled`, `public`
`path`	String	The path of the repository. At least one of `name`, `path` must

Property	Type	Description
		be provided Example: `new_project`
`printing_merge_request_link_enabled`	Boolean	Show link to create/view merge request when pushing from the command line
`public_builds`	Boolean	Deprecated: Use public_jobs instead
`public_jobs`	Boolean	Perform public builds
`releases_access_level`	String	Releases access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`repository_object_format`	String	The object format of the project repository Allowed values: `sha1`, `sha256`
`repository_storage`	String	Which storage shard the repository is on. Available only to admins
`request_access_enabled`	Boolean	Allow users to request member access
`resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
	String	The process mode of the resource group

Property	Type	Description
`resource_group_default_process_mode`		Allowed values: `unordered`, `oldest_first`, `newest_first`, `newest_ready_first`
`security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`service_desk_enabled`	Boolean	Disable or enable the service desk
`shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`show_default_award_emojis`	Boolean	Show default award emojis
`show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`snippets_enabled`	Boolean	Flag indication if snippets are enabled
`squash_commit_template`	String	Template used to create squash commit message
`squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`

Property	Type	Description
`suggestion_commit_message`	String	The commit message used to apply merge request suggestions
`tag_list`	Array of strings	Deprecated: Use :topics instead
`template_name`	String	Name of template from which to create project. Mutually exclusive with `import_url`, `template_project_id`
`template_project_id`	Integer	Project ID of template from which to create project. Mutually exclusive with `import_url`, `template_name`
`topics`	Array of strings	The list of topics for a project
`visibility`	String	The visibility of the project Allowed values: `private`, `internal`, `public`
`warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters
`wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`wiki_enabled`	Boolean	Flag indication if the wiki is enabled

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Create a project for a user

`POST /api/v4/projects/user/{user_id}`

Creates a project for a user. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user to assign as the project owner Example: `1`

Request body (`multipart/form-data`)

Property	Type	Description
`allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped
`analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`auto_cancel_pending_pipelines`	String	Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
`autoclose_referenced_issues`	Boolean	Flag indication if referenced issues auto-closing is enabled
`avatar`	String (binary)	Avatar image for project
`build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`build_timeout`	Integer	Build timeout
`builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project

Property	Type	Description
`container_expiration_policy_attributes`	Object	Object that contains information on the container expiration policy
`container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
`container_expiration_policy_attributes.enabled`	Boolean	Flag indication if container expiration policy is enabled
`container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal
`container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry Allowed values: `disabled`, `private`, `enabled`
`container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container

Property	Type	Description
		registry is enabled for that project
`default_branch`	String	The default branch of the project Example: `main`
`description`	String	The description of the project
`emails_disabled`	Boolean	Deprecated: Use emails_ enabled instead
`emails_enabled`	Boolean	Enable email notifications
`enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`environments_access_level`	String	Environments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`import_url`	String	URL from which the project is imported. Mutually exclusive with `template_name`, `template_project_id`
	String	Infrastructure access level. One of `disabled`, `private` or

Property	Type	Description
`infrastructure_access_level`		`enabled` Allowed values: `disabled`, `private`, `enabled`
`initialize_with_readme`	Boolean	Initialize a project with a README.md
`issue_branch_template`	String	Template used to create a branch from an issue
`issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`jobs_enabled`	Boolean	Flag indication if jobs are enabled
`lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
`merge_commit_template`	String	Template used to create merge commit message
`merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`merge_request_title_regex`	String	Regex pattern for validating merge request titles. All or none of `merge_request_title_regex`, `merge_request_title_regex_description` must be provided Maximum length: `255`
`merge_request_title_regex_description`	String	Example merge request title that matches merge_request_title_regex. Shown when a title

Property	Type	Description
		does not match. All or none of `merge_request_title_regex`, `merge_request_title_regex_description` must be provided Maximum length: `255`
`merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`
`name` Required	String	The name of the project Example: `New Project`

Property	Type	Description
`namespace_id`	Integer	Namespace ID for the new project. Default to the user namespace
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`, `enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry Allowed values: `disabled`, `private`, `enabled`, `public`
`packages_enabled`	Boolean	Deprecated: Use :package_registry_access_level instead. Enable project packages feature
`pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public` Allowed values: `disabled`, `private`, `enabled`, `public`
`path`	String	The path of the repository Example: `new_project`
	Boolean	

Property	Type	Description
`printing_merge_request_link_enabled`		Show link to create/view merge request when pushing from the command line
`public_builds`	Boolean	Deprecated: Use public_jobs instead
`public_jobs`	Boolean	Perform public builds
`releases_access_level`	String	Releases access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`repository_object_format`	String	The object format of the project repository Allowed values: `sha1`, `sha256`
`repository_storage`	String	Which storage shard the repository is on. Available only to admins
`request_access_enabled`	Boolean	Allow users to request member access
`resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
`resource_group_default_process_mode`	String	The process mode of the resource group Allowed values: `unordered`, `oldest_first`,

Property	Type	Description
		`newest_first`, `newest_ready_first`
`security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`service_desk_enabled`	Boolean	Disable or enable the service desk
`shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`show_default_award_emojis`	Boolean	Show default award emojis
`show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`snippets_enabled`	Boolean	Flag indication if snippets are enabled
`squash_commit_template`	String	Template used to create squash commit message
`squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`
`suggestion_commit_message`	String	

Property	Type	Description
		The commit message used to apply merge request suggestions
`tag_list`	Array of strings	Deprecated: Use :topics instead
`template_name`	String	Name of template from which to create project. Mutually exclusive with `import_url`, `template_project_id`
`template_project_id`	Integer	Project ID of template from which to create project. Mutually exclusive with `import_url`, `template_name`
`topics`	Array of strings	The list of topics for a project
`visibility`	String	The visibility of the project Allowed values: `private`, `internal`, `public`
`warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters
`wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`wiki_enabled`	Boolean	Flag indication if the wiki is enabled

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Retrieve a project

`GET /api/v4/projects/{id}`

Retrieves details on a specified project. This endpoint can be accessed without authentication if the project is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`statistics` Query	Boolean	If `true`, includes project statistics. Available only to users with the Reporter, Developer, Maintainer, or Owner role Default: `false`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only
`license` Query	Boolean	If `true`, includes the project license data in the response Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsWithAccessAndCatalogSetting
`400`	Bad Request	—
`404`	Not Found	—

Update a project

`PUT /api/v4/projects/{id}`

Updates an existing project. If your HTTP repository is not publicly accessible, add authentication information to the URL `https://username:password@gitlab.company.com/group/project.git`, where `password` is a public access key with the `api` scope. At least one project attribute must be updated.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`allow_merge_on_skipped_pipeline`	Boolean	Allow to merge if pipeline is skipped
`allow_pipeline_trigger_approve_deployment`	Boolean	Allow the pipeline triggerer to approve deployments
`analytics_access_level`	String	Analytics access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`auto_cancel_pending_pipelines`	String	Auto-cancel pending pipelines Allowed values: `disabled`, `enabled`
`auto_devops_deploy_strategy`	String	Auto Deploy strategy Allowed values: `continuous`, `manual`, `timed_incremental`
`auto_devops_enabled`	Boolean	Flag indication if Auto DevOps is enabled
`autoclose_referenced_issues`	Boolean	Flag indication if referenced issues auto-closing is enabled
`automatic_rebase_enabled`	Boolean	Enable automatic rebase of the source branch before merge
`avatar`	String (binary)	Avatar image for project
`build_git_strategy`	String	The Git strategy. Defaults to `fetch` Allowed values: `fetch`, `clone`
`build_timeout`	Integer	Build timeout
`builds_access_level`	String	Builds access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`

Property	Type	Description
`ci_allow_fork_pipelines_to_run_in_parent_project`	Boolean	Allow fork merge request pipelines to run in parent project
`ci_config_path`	String	The path to CI config file. Defaults to `.gitlab-ci.yml`
`ci_default_git_depth`	Integer	Default number of revisions for shallow cloning
`ci_delete_pipelines_in_seconds`	Integer	Pipelines older than the configured time are deleted
`ci_display_pipeline_variables`	Boolean	Display all manually-defined variables in the pipeline details page after running a pipeline manually
`ci_forward_deployment_enabled`	Boolean	Prevent older deployment jobs that are still pending
`ci_forward_deployment_rollback_allowed`	Boolean	Allow job retries for rollback deployments
`ci_id_token_sub_claim_components`	Array of strings	Claims that will be used to build the sub claim in id tokens
`ci_pipeline_variables_minimum_override_role`	String	Limit ability to override CI/CD variables when triggering a pipeline to only users with at least the set minimum role Allowed values: `no_one_allowed`, `developer`, `maintainer`, `owner`
`ci_push_repository_for_job_token_allowed`	Boolean	Allow pushing to this project's repository by authenticating with a CI/CD job token generated in this project
`ci_separated_caches`	Boolean	Enable or disable separated caches based on branch protection

Property	Type	Description
`ci_skip_branch_pipelines_for_mrs`	Boolean	Skip the branch pipeline when a branch has an open merge request, and run only the merge request pipeline
`cicd_catalog_enabled`	Boolean	Set whether the project is a CI/CD catalog project
`container_expiration_policy_attributes`	Object	Object that contains information on the container expiration policy
`container_expiration_policy_attributes.cadence`	String	Container expiration policy cadence for recurring job
`container_expiration_policy_attributes.enabled`	Boolean	Flag indication if container expiration policy is enabled
`container_expiration_policy_attributes.keep_n`	Integer	Container expiration policy number of images to keep
`container_expiration_policy_attributes.name_regex`	String	Container expiration policy regex for image removal
`container_expiration_policy_attributes.name_regex_keep`	String	Container expiration policy regex for image retention
`container_expiration_policy_attributes.older_than`	String	Container expiration policy remove images older than value
`container_registry_access_level`	String	Controls visibility of the container registry. One of `disabled`, `private` or `enabled`. `private` will make the container registry accessible only to project members (reporter role and above). `enabled` will make the container registry accessible to everyone who has access to the project. `disabled` will disable the container registry

Property	Type	Description
		Allowed values: `disabled`, `private`, `enabled`
`container_registry_enabled`	Boolean	Deprecated: Use :container_registry_access_level instead. Flag indication if the container registry is enabled for that project
`default_branch`	String	The default branch of the project Example: `main`
`description`	String	The description of the project
`emails_disabled`	Boolean	Deprecated: Use emails_enabled instead
`emails_enabled`	Boolean	Enable email notifications
`enforce_auth_checks_on_uploads`	Boolean	Enforce auth check on uploads
`environments_access_level`	String	Environments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`fe_project_setting_attributes`	Object	FE project settings attributes
`fe_project_setting_attributes.opensearch_indexed_branches_regex`	String	Regex for indexing additional branches in OpenSearch (default branch is always indexed)
`feature_flags_access_level`	String	Feature flags access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`feature_flags_minimum_role`	String	Limit the ability to create, update, toggle, and delete

Property	Type	Description
		feature flags to only users with at least the set minimum role Allowed values: `no_one_allowed`, `developer`, `maintainer`, `owner`
`forking_access_level`	String	Forks access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`group_runners_enabled`	Boolean	Flag indication if group runners are enabled for that project
`infrastructure_access_level`	String	Infrastructure access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`issue_branch_template`	String	Template used to create a branch from an issue
`issues_access_level`	String	Issues access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`issues_enabled`	Boolean	Flag indication if the issue tracker is enabled
`jaga_task_required_to_merge`	Boolean	Require the merge request title or description to name a Jaga task. Ignored unless the project has an active Jaga integration
`jobs_enabled`	Boolean	Flag indication if jobs are enabled
`keep_latest_artifact`	Boolean	Indicates if the latest artifact should be kept for this project

Property	Type	Description
`lfs_enabled`	Boolean	Flag indication if Git LFS is enabled for that project
`max_artifacts_size`	Integer	Set the maximum file size for each job's artifacts
`max_pipelines_per_merge_train`	Integer	Set the maximum number of parallel pipelines for a merge train
`merge_commit_template`	String	Template used to create merge commit message
`merge_method`	String	The merge method used when merging merge requests Allowed values: `ff`, `rebase_merge`, `merge`
`merge_pipelines_enabled`	Boolean	Enable or disable merged results pipelines
`merge_request_title_regex`	String	Regex pattern for validating merge request titles Maximum length: `255`
`merge_request_title_regex_description`	String	Example merge request title that matches merge_request_title_regex. Shown when a title does not match Maximum length: `255`
`merge_requests_access_level`	String	Merge requests access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`merge_requests_enabled`	Boolean	Flag indication if merge requests are enabled
`merge_requests_template`	String	—
`merge_train_enforced`	Boolean	Require merge requests to merge through a merge train

Property	Type	Description
`merge_trains_enabled`	Boolean	Enable or disable merge trains
`merge_trains_skip_train_allowed`	Boolean	Allow merge requests to merge immediately without restarting the merge train
`model_experiments_access_level`	String	Model experiments access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`model_registry_access_level`	String	Model registry access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`monitor_access_level`	String	Monitor access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`mr_default_target_self`	Boolean	Merge requests of this forked project targets itself by default
`mr_default_title_template`	String	Template used to generate the default merge request title. Maximum 100 characters Maximum length: `100`
`name`	String	The name of the project Example: `project`
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	Only allow to merge if all threads are resolved
`only_allow_merge_if_pipeline_succeeds`	Boolean	Only allow to merge if builds succeed
`package_registry_access_level`	String	Controls visibility of the package registry. One of `disabled`, `private`,

Property	Type	Description
		`enabled` or `public`. `private` will make the package registry accessible only to project members (reporter role and above). `enabled` will make the package registry accessible to everyone who has access to the project. `public` will make the package registry accessible to everyone. `disabled` will disable the package registry Allowed values: `disabled`, `private`, `enabled`, `public`
`packages_enabled`	Boolean	Deprecated: Use :package_ registry_access_level instead. Enable project packages feature
`pages_access_level`	String	Pages access level. One of `disabled`, `private`, `enabled` or `public` Allowed values: `disabled`, `private`, `enabled`, `public`
`path`	String	The path of the repository Example: `group/project`
`printing_merge_request_link_enabled`	Boolean	Show link to create/view merge request when pushing from the command line
`protect_merge_request_pipelines`	Boolean	Make protected CI/CD variables and runners available in merge request pipelines
`public_builds`	Boolean	Deprecated: Use public_jobs instead
`public_jobs`	Boolean	Perform public builds
`releases_access_level`	String	Releases access level. One of `disabled`, `private` or

Property	Type	Description
		`enabled` Allowed values: `disabled`, `private`, `enabled`
`remove_source_branch_after_merge`	Boolean	Remove the source branch by default after merge
`repository_access_level`	String	Repository access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`repository_storage`	String	Which storage shard the repository is on. Available only to admins
`request_access_enabled`	Boolean	Allow users to request member access
`resolve_outdated_diff_discussions`	Boolean	Automatically resolve merge request diff threads on lines changed with a push
`resource_group_default_process_mode`	String	The process mode of the resource group Allowed values: `unordered`, `oldest_first`, `newest_first`, `newest_ready_first`
`restrict_user_defined_variables`	Boolean	Restrict use of user-defined variables when triggering a pipeline
`security_and_compliance_access_level`	String	Security and compliance access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`service_desk_enabled`	Boolean	Disable or enable the service desk

Property	Type	Description
`shared_runners_enabled`	Boolean	Flag indication if shared runners are enabled for that project
`show_default_award_emojis`	Boolean	Show default award emojis
`show_diff_preview_in_email`	Boolean	Include the code diff preview in merge request notification emails
`snippets_access_level`	String	Snippets access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`snippets_enabled`	Boolean	Flag indication if snippets are enabled
`squash_commit_template`	String	Template used to create squash commit message
`squash_option`	String	Squash default for project. One of `never`, `always`, `default_on`, or `default_off` Allowed values: `never`, `always`, `default_on`, `default_off`
`suggestion_commit_message`	String	The commit message used to apply merge request suggestions
`tag_list`	Array of strings	Deprecated: Use :topics instead
`topics`	Array of strings	The list of topics for a project
`visibility`	String	The visibility of the project Allowed values: `private`, `internal`, `public`
`warn_about_potentially_unwanted_characters`	Boolean	Warn about potentially unwanted characters

Property	Type	Description
`wiki_access_level`	String	Wiki access level. One of `disabled`, `private` or `enabled` Allowed values: `disabled`, `private`, `enabled`
`wiki_enabled`	Boolean	Flag indication if the wiki is enabled

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsWithCatalogSetting
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not Found	—

Delete a project

`DELETE /api/v4/projects/{id}`

Deletes a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Search a project

`GET /api/v4/projects/{id}/(-/)search`

Searches for content in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query, required	String	Search query
`scope` Query, required	String	Scope of the search Allowed values: `issues`, `work_items`, `merge_requests`, `milestones`, `notes`, `wiki_blobs`, `commits`, `blobs`, `users` Minimum length: `1`
`ref` Query	String	Name of the repository branch or tag. If omitted, uses the default branch
`state` Query	String	Filter results by state. Applies only to the `issues`, `work_items`, and `merge_requests` scopes Allowed values: `all`, `opened`, `closed`, `merged`
`confidential` Query	Boolean	If `true`, returns only confidential results. If `false`, returns only non-confidential results. Applies only to the `issues` and `work_items` scopes
`type` Query	Array of strings	Filter work items by type. Only applies to work_items scope. Available types: issue, task, incident, ticket
`fields` Query	Array of strings	Array of fields to search. Only “title” is supported (work_items / issues scope)
	Array of strings	

Name	Type	Description
`language` Query		Array of languages to filter blobs by language
`label_name` Query	Array of strings	Array of labels to filter work_ items / issues / merge_ requests by label
`source_branch` Query	String	Filter merge requests by source branch Maximum length: `255`
`target_branch` Query	String	Filter merge requests by target branch Maximum length: `255`
`not_source_branch` Query	String	Exclude merge requests by source branch Maximum length: `255`
`not_target_branch` Query	String	Exclude merge requests by target branch Maximum length: `255`
`author_username` Query	String	Filter merge requests by author username Maximum length: `255`
`not_author_username` Query	String	Exclude merge requests by author username Maximum length: `255`
`num_context_lines` Query	Integer	Number of context lines around each match. Only supported for blobs scope Maximum: `20` Minimum: `0`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Archive a project

`POST /api/v4/projects/{id}/archive`

Archives a specified project. You must be an administrator or be assigned the Owner role on the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

Create a fork of a project

`POST /api/v4/projects/{id}/fork`

Creates a fork of a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`branches`	String	Branches to fork. Leave empty to fork all branches
`description`	String	Description assigned to the new project after forking Example: `Description`
`mr_default_target_self`	Boolean	If `true`, merge requests from the forked project target this project. If `false`, they target the upstream project
`name`	String	Name assigned to the new project after forking Example: `Fork`
`namespace`	String	ID or path of the namespace to fork the new project into. Deprecated. Use `namespace_id` or `namespace_path` instead Example: `gitlab`
`namespace_id`	Integer	ID of the namespace to fork the new project into Example: `1`
`namespace_path`	String	Path of the namespace to fork the new project into Example: `new_path/gitlab`
`path`	String	Path assigned to the new project after forking Example: `fork`
`visibility`	String	Visibility level assigned to the new project after forking Allowed values: `private`, `internal`, `public`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—
`409`	Conflict	—

Delete a fork relationship

`DELETE /api/v4/projects/{id}/fork`

Deletes a fork relationship between projects.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`204`	No Content	—
`304`	Not modified	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Create a fork relationship

`POST /api/v4/projects/{id}/fork/{forked_from_id}`

Creates a fork relationship between a project and an upstream project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`forked_from_id` Path, required	String	ID of the project that was forked from Example: `gitlab`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Unauthenticated	—
`404`	Not found	—

List all forks of a project

`GET /api/v4/projects/{id}/forks`

Lists all forks of a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`order_by` Query	String	Return projects ordered by field. storage_size, repository_size, wiki_size, packages_size are only available to admins. Similarity is available when searching and is limited to projects the user has access to Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count`, `storage_size`, `repository_size`, `wiki_size`, `packages_size` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`archived` Query	Boolean	If `true`, returns only archived projects
`visibility` Query	String	Return projects with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return a list of projects with a `path`, `name`, or `description` matching the search criteria (case-insensitive, substring match). Multiple terms can be provided, separated by an escaped space, either `+` or `%20`, and are ANDed together. Example: `one+two` matches

Name	Type	Description
		substrings `one` and `two` (in any order)
`search_namespaces` Query	Boolean	If `true`, includes ancestor namespaces when matching search criteria
`owned` Query	Boolean	If `true`, returns only projects explicitly owned by the current user Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`imported` Query	Boolean	If `true`, returns only projects that were imported from external systems by the authenticated user Default: `false`
`membership` Query	Boolean	If `true`, returns only projects that the current user is a member of Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`with_programming_language` Query	String	Return projects that use the given programming language
`min_access_level` Query	Integer	Return results where the current user has at least the specified access level. For a list of values, see roles

Name	Type	Description
		Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`id_after` Query	Integer	Return projects with IDs greater than the specified ID
`id_before` Query	Integer	Return projects with IDs less than the specified ID
`last_activity_after` Query	String (date-time)	Return projects with last activity after the specified time
`last_activity_before` Query	String (date-time)	Return projects with last activity before the specified time
`repository_storage` Query	String	Return projects stored on the specified repository storage shard. Administrators only
`topic` Query	Array of strings	Comma-separated list of topics. Return projects having all topics
`topic_id` Query	Integer	Filter projects by the given topic ID
`updated_before` Query	String (date-time)	Return projects last updated on or before the specified time
`updated_after` Query	String (date-time)	Return projects last updated on or after the specified time
`include_pending_delete` Query	Boolean	If `true`, includes projects that are pending deletion in the response. Administrators only
`marked_for_deletion_on` Query	String (date)	Return projects that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only projects that are not archived and not marked for deletion

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProject
`400`	Bad Request	—
`404`	Not Found	—

List all ancestor groups

`GET /api/v4/projects/{id}/groups`

Lists all ancestor groups for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query	String	Return groups matching the search criteria Example: `group`
`skip_groups` Query	Array of integers	Skip the specified group IDs
`with_shared` Query	Boolean	If `true`, includes groups the project is shared with Default: `false`
`shared_visible_only` Query	Boolean	If `true`, returns only shared groups the authenticated user can access Default: `false`
`shared_min_access_level` Query	Integer	Return shared groups with at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPublicGroupDetails
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Start the housekeeping task for a project

`POST /api/v4/projects/{id}/housekeeping`

Starts the housekeeping task for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`task`	String	`prune` to trigger manual prune of unreachable objects or `eager` to trigger eager housekeeping Allowed values: `eager` , `prune` Default: `eager`

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Unauthenticated	—
`404`	Not Found	—
`409`	Conflict	—

Import members

`POST /api/v4/projects/{id}/import_project_members/{project_id}`

Imports members from another project. If the role of the importing member for the target project is a Maintainer, then members with the Owner role for the source project are imported with the Maintainer role. If the importing member is an Owner, then members with the Owner role for the source project are imported with the Owner role.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`project_id` Path, required	Integer	ID of the source project to import the members from

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden - Project	—
`404`	Project Not Found	—
`422`	Import failed	—

List all invited groups in a project

`GET /api/v4/projects/{id}/invited_groups`

Lists all invited groups in a project. Unauthenticated requests return only public invited groups. Limited to 60 requests a minute per user account for authenticated requires and per IP address for unauthenticated requests. Supports offset-based pagination (up to 50,000 projects) and keyset-based pagination (greater than 50,000 projects).

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`relation` Query	Array of strings	Filter groups by relation
`search` Query	String	Return the list of authorized groups matching the search criteria
`min_access_level` Query	Integer	Return results where the current user has at least the specified access level. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPublic GroupDetails
`400`	Bad Request	—
`404`	Not Found	—

List all project issues

`GET /api/v4/projects/{id}/issues`

Lists all issues for a specified project. If the project is private, you need to provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`with_labels_details` Query	Boolean	If `true`, the response returns more details for each label in the labels field: `name`, `color`, `description`, `description_html`, `text_color` Default: `false`
`state` Query	String	Filter issues by state Allowed values: `opened`, `closed`, `all` Default: `all`
`closed_by_id` Query	Integer	Return issues closed by the user with the given ID
`order_by` Query	String	Sort results by the specified field Allowed values: `created_at`, `due_date`, `label_priority`, `milestone_due`, `popularity`, `priority`, `relative_position`, `title`, `updated_at` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`due_date` Query	String	Return issues that have no due date, are overdue, or whose due date is this week, this month, or between two weeks ago and next month Allowed values: `0`, `any`, `today`, `tomorrow`, `overdue`, `week`, `month`, `next_month_and_previous_two_weeks`, ``

Name	Type	Description
`issue_type` Query	String	The type of the issue. Accepts: issue, incident, test_case, requirement, task, ticket Allowed values: `issue`, `incident`, `test_case`, `requirement`, `task`, `ticket`
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for

Name	Type	Description
		`Upcoming` and `Started` differs from the logic used in the GraphQL API. Mutually exclusive with `milestone` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid

Name	Type	Description
		parameter error is returned. When multiple usernames are given, only issues assigned to all of them are returned. Mutually exclusive with `assignee_id`
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`

Name	Type	Description
`not[author_username]` Query	String	Return issues not authored by the user with the given username. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer	Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `all`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Name	Type	Description
`cursor` Query	String	Cursor for obtaining the next set of records

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Create an issue

`POST /api/v4/projects/{id}/issues`

Creates an issue for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`add_labels`	Array of strings	Comma-separated label names to add to the issue. If a label does not already exist, this creates a new project label and assigns it to the issue
`assignee_id`	Integer	ID of the user to assign the issue to. Available only on Git Lab Free. Deprecated. Use `assignee_ids` instead
`assignee_ids`	Array of integers	IDs of the users to assign to the issue. Set to `0` or leave empty to unassign all assignees. Assigning more than one user is Premium and Ultimate only
`confidential`	Boolean	If `true`, the issue is confidential
`created_at`	String (date-time)	Date and time the issue was created. Requires administrator or project/group owner rights
`description`	String	Description of the issue. Limited to 1,048,576 characters
`discussion_locked`	Boolean	If `true`, locks the issue's discussion so only project members can add or edit comments
`discussion_to_resolve`	String	ID of a discussion to resolve. Use in combination with `merge_request_to_resolve_discussions_of`
`due_date`	String	Due date, in the format `YYYY-MM-DD`, for example `2016-03-11`

Property	Type	Description
`iid`	Integer	Internal ID to assign to the new issue. Administrators or project owners only
`issue_type`	String	The type of the issue. Accepts: issue, incident, test_case, requirement, task, ticket Allowed values: `issue`, `incident`, `test_case`, `requirement`, `task`, `ticket`
`labels`	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`merge_request_to_resolve_discussions_of`	Integer	Internal ID of a merge request for which to resolve discussions. This fills the issue with a default description and marks all discussions as resolved, unless a title or description is provided
`milestone`	String	The title of a project or ancestor-group milestone to assign the issue to. Mutually exclusive with `milestone_id` Maximum length: `255`
`milestone_id`	Integer	Global ID of a milestone to assign to the issue. Set to `0` or leave empty to unassign the

Property	Type	Description
		milestone. Mutually exclusive with `milestone`
`remove_labels`	Array of strings	Comma-separated label names to remove from the issue
`severity`	String	The severity of the issue. Only applies to incidents. Accepts: unknown, low, medium, high, critical Allowed values: `unknown`, `low`, `medium`, `high`, `critical`
`start_date`	String	Start date, in the format `YYYY-MM-DD`, for example `2016-03-11`. Introduced in GitLab 19.1
`title` Required	String	Title of the issue

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a project issue

`GET /api/v4/projects/{id}/issues/{issue_iid}`

Retrieves a specified issue for a project. If the project is private or the issue is confidential, you need to provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Update an issue

`PUT /api/v4/projects/{id}/issues/{issue_iid}`

Updates a specified issue for a project. This request is also used to close or reopen an issue using the `state_event` parameter. At least one of the following parameters is required for the request to be successful: `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `issue_type`, `labels`, `milestone_id`, `state_event`, `title`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`add_labels`	Array of strings	Comma-separated label names to add to the issue. If a label does not already exist, this creates a new project label and assigns it to the issue. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`assignee_id`	Integer	ID of the user to assign the issue to. Available only on Git Lab Free. Deprecated. Use `assignee_ids` instead. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`assignee_ids`	Array of integers	IDs of the users to assign to the issue. Set to `0` or leave empty to unassign all assignees. Assigning more than one user is Premium and Ultimate only. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_`

Property	Type	Description
		`date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`confidential`	Boolean	If `true`, the issue is confidential. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`created_at`	String	At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`description`	String	Description of the issue. Limited to 1,048,576 characters. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided

Property	Type	Description
`discussion_locked`	Boolean	If `true`, locks the issue's discussion so only project members can add or edit comments. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`due_date`	String	Due date, in the format `YYYY-MM-DD`, for example `2016-03-11`. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`issue_type`	String	The type of the issue. Accepts: issue, incident, test_case, requirement, task, ticket. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided Allowed values: `issue`,

Property	Type	Description
		`incident`, `test_case`, `requirement`, `task`, `ticket`
`labels`	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`milestone`	String	The title of a project or ancestor-group milestone to assign the issue to. Mutually exclusive with `milestone_id`. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided Maximum length: `255`

Property	Type	Description
`milestone_id`	Integer	Global ID of a milestone to assign to the issue. Set to `0` or leave empty to unassign the milestone. Mutually exclusive with `milestone`. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`remove_labels`	Array of strings	Comma-separated label names to remove from the issue. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`severity`	String	The severity of the issue. Only applies to incidents. Accepts: unknown, low, medium, high, critical. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_`

Property	Type	Description
		`type` must be provided Allowed values: `unknown`, `low`, `medium`, `high`, `critical`
`start_date`	String	Start date, in the format `YYYY-MM-DD`, for example `2016-03-11`. Introduced in GitLab 19.1. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided
`state_event`	String	Event to change the state of the issue. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`, `severity`, `state_event`, `title`, `issue_type` must be provided Allowed values: `reopen`, `close`
`title`	String	Title of the issue. At least one of `assignee_id`, `assignee_ids`, `confidential`, `created_at`, `description`, `discussion_locked`, `due_date`, `start_date`, `labels`, `add_labels`, `remove_labels`, `milestone_id`, `milestone`,

Property	Type	Description
		`severity`, `state_event`, `title`, `issue_type` must be provided
`updated_at`	String (date-time)	Date and time the issue was updated. Administrators or project owners only. Empty or null values are not accepted

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Delete an issue

`DELETE /api/v4/projects/{id}/issues/{issue_iid}`

Deletes a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Clone an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/clone`

Clones a specified issue to a project. Copies as much data as possible as long as the target project contains equivalent criteria, such as labels or milestones. If you have insufficient permissions, an error message with status code `400` is returned.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`to_project_id` Required	Integer	ID of the new project
`with_notes`	Boolean	If `true` , clones the issue with its notes Default: `false`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

List all merge requests that close an issue on merge

`GET /api/v4/projects/{id}/issues/{issue_iid}/closed_by`

Lists all merge requests that close a specified issue when merged. If the project is private or the issue is confidential, you need to provide credentials to authorize. In most cases, you should authenticate with a personal access token.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesMergeRequestBasic
`400`	Bad Request	—
`404`	Not Found	—

Move an issue

`POST /api/v4/projects/{id}/issues/{issue_iid}/move`

Moves a specified issue to a different project. If the target project is the source project or the user has insufficient permissions, an error message with status code `400` is returned. If a label or milestone with the same name also exists in the target project, it is then assigned to the issue being moved.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`to_project_id` Required	Integer	ID of the new project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Update the order of an issue

`PUT /api/v4/projects/{id}/issues/{issue_iid}/reorder`

Updates the order of a specified issue in a project. You can see the results when sorting issues manually.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	Internal ID of the issue

Request body (`application/json`)

Property	Type	Description
`move_after_id`	Integer	Global ID of the project issue to place this issue after. At least one of `move_after_id` , `move_before_id` must be provided
`move_before_id`	Integer	Global ID of the project issue to place this issue before. At least one of `move_after_id` , `move_before_id` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Retrieve issues statistics for a project

`GET /api/v4/projects/{id}/issues_statistics`

Retrieves statistics for issues in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`labels` Query	Array of strings	Comma-separated list of label names. `None` means no labels are assigned. `Any` means at least one label is assigned. `No+Label` (deprecated) means no labels are assigned. Set to an empty string to unassign all labels. If a label does not already exist, this creates a new project label and assigns it to the issue. Predefined names are case-insensitive
`milestone` Query	String	Milestone title. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. Support for `None` and `Any` is planned for removal. Use the `milestone_id` attribute instead. Mutually exclusive with `milestone_id`
`milestone_id` Query	String	Return issues assigned to milestones with a given timebox value. `None` lists all issues with no milestone. `Any` lists all issues that have an assigned milestone. `Upcoming` lists all issues assigned to milestones due in the future. `Started` lists all issues assigned to open, started milestones. The logic for `Upcoming` and `Started` differs from the logic used in the GraphQL API. Mutually exclusive with `milestone`

Name	Type	Description
		Allowed values: `Any`, `None`, `Upcoming`, `Started`
`iids` Query	Array of integers	Return only the issues having the given `iid`
`search` Query	String	Search issues against their `title` and `description`
`in` Query	String	Modify the scope of the `search` attribute to `title`, `description`, or `title,description`. If omitted, defaults to `title,description`
`author_id` Query	Integer	Return issues created by the given user ID. Combine with `scope=all` or `scope=assigned_to_me`. Mutually exclusive with `author_username`
`author_username` Query	String	Return issues created by the given `username`. Mutually exclusive with `author_id`
`assignee_id` Query	Integer or string	Return issues assigned to the given user `id`. `None` returns unassigned issues and `Any` returns issues with an assignee. Mutually exclusive with `assignee_username`
`assignee_username` Query	Array of strings	Return issues assigned to the given username. In GitLab Community Edition, only a single value is accepted. Otherwise, an invalid parameter error is returned. When multiple usernames are given, only issues assigned to all of them are returned.

Name	Type	Description
		Mutually exclusive with `assignee_id`
`created_after` Query	String (date-time)	Return issues created on or after the specified time
`created_before` Query	String (date-time)	Return issues created on or before the specified time
`updated_after` Query	String (date-time)	Return issues updated on or after the specified time
`updated_before` Query	String (date-time)	Return issues updated on or before the specified time
`not` Query	Object	Return issues that do not match the specified parameters
`not[labels]` Query	Array of strings	Comma-separated list of label names
`not[milestone]` Query	String	Milestone title. Mutually exclusive with `not[milestone_id]`
`not[milestone_id]` Query	String	Return issues assigned to milestones without the specified timebox value. Mutually exclusive with `not[milestone]` Allowed values: `Any`, `None`, `Upcoming`, `Started`
`not[iids]` Query	Array of integers	Internal IDs of issues to exclude
`not[author_id]` Query	Integer	Return issues not authored by the user with the given ID. Mutually exclusive with `not[author_username]`
`not[author_username]` Query	String	Return issues not authored by the user with the given

Name	Type	Description
		username. Mutually exclusive with `not[author_id]`
`not[assignee_id]` Query	Integer	Return issues not assigned to the user with the given ID. Mutually exclusive with `not[assignee_username]`
`not[assignee_username]` Query	Array of strings	Return issues not assigned to the user with the given username. Mutually exclusive with `not[assignee_id]`
`scope` Query	String	Return issues for the given scope Allowed values: `created-by-me`, `assigned-to-me`, `created_by_me`, `assigned_to_me`, `all`
`my_reaction_emoji` Query	String	Return issues reacted to by the authenticated user with the given `emoji`. `None` returns issues with no reaction and `Any` returns issues with at least one reaction
`confidential` Query	Boolean	If `true`, returns only confidential issues. If `false`, returns only public issues

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve programming language usage information

`GET /api/v4/projects/{id}/languages`

Retrieves information about all programming languages used in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not found	—

List all package protection rules

`GET /api/v4/projects/{id}/packages/protection/rules`

Lists all package protection rules for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProjects PackagesProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a package protection rule

`POST /api/v4/projects/{id}/packages/protection/rules`

Creates a package protection rule for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete`	String	Minimum GitLab access level required to delete a package. If the value is `null`, the default minimum access level is `maintainer`. Must be provided when `minimum_access_level_for_push` is not set. Behind a feature flag named `packages_protected_packages_delete`. Disabled by default. At least one of `minimum_access_level_for_push`, `minimum_access_level_for_delete` must be provided Allowed values: `owner`, `admin`
`minimum_access_level_for_push`	String	Minimum GitLab access level required to push a package. If the value is `null`, the default minimum access level is `developer`. Must be provided when `minimum_access_level_for_delete` is not set. At least one of `minimum_access_level_for_push`, `minimum_access_level_for_delete` must be provided Allowed values: `maintainer`, `owner`, `admin`
`package_name_pattern` Required	String	Package name protected by the protection rule, for example `@my-scope/my-package-*`. Wildcard character `*` is allowed
`package_type` Required	String	Package type protected by the protection rule, for example `npm` Allowed values: `cargo`,

Property	Type	Description
		`conan`, `generic`, `helm`, `maven`, `npm`, `nuget`, `pypi`, `terraform_module` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsPackagesProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Update a package protection rule

`PATCH /api/v4/projects/{id}/packages/protection/rules/{package_protection_rule_id}`

Updates a package protection rule for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_protection_rule_id` Path, required	Integer	ID of the package protection rule

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete`	String	Minimum GitLab access level required to delete a package. If the value is `null`, the default minimum access level is `maintainer`. Must be provided when `minimum_access_level_for_push` is not set. Behind a feature flag named `packages_protected_packages_delete`. Disabled by default Allowed values: `owner`, `admin`
`minimum_access_level_for_push`	String	Minimum GitLab access level required to push a package. If the value is `null`, the default minimum access level is `developer`. Must be provided when `minimum_access_level_for_delete` is not set Allowed values: `maintainer`, `owner`, `admin`
`package_name_pattern`	String	Package name protected by the protection rule, for example `@my-scope/my-package-*`. Wildcard character `*` is allowed
`package_type`	String	Package type protected by the protection rule, for example `npm` Allowed values: `cargo`, `conan`, `generic`, `helm`, `maven`, `npm`, `nuget`, `pypi`, `terraform_module`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsPackagesProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Delete a package protection rule

`DELETE /api/v4/projects/{id}/packages/protection/rules/{package_protection_rule_id}`

Deletes a package protection rule from a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`package_protection_rule_id` Path, required	Integer	ID of the package protection rule

Responses

Code	Description	Schema
`204`	204 No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Check pages access of this project

`GET /api/v4/projects/{id}/pages_access`

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all container repository protection rules

`GET /api/v4/projects/{id}/registry/protection/repository/rules`

Lists all container repository protection rules for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProjectsContainerRegistryProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a container repository protection rule

`POST /api/v4/projects/{id}/registry/protection/repository/rules`

Creates a container repository protection rule for a specified project to control who can push or delete container images.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete`	String	Minimum GitLab access level required to delete container images in the container registry. Must be provided when `minimum_access_level_for_push` is not set. At least one of `minimum_access_level_for_push`, `minimum_access_level_for_delete` must be provided Allowed values: `maintainer`, `owner`, `admin`
`minimum_access_level_for_push`	String	Minimum GitLab access level required to push container images to the container registry. Must be provided when `minimum_access_level_for_delete` is not set. At least one of `minimum_access_level_for_push`, `minimum_access_level_for_delete` must be provided Allowed values: `maintainer`, `owner`, `admin`
`repository_path_pattern` Required	String	Container repository path pattern protected by the protection rule, for example `flight/flight-*`. Wildcard character `*` is allowed

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsContainerRegistryProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Update a container repository protection rule

`PATCH /api/v4/projects/{id}/registry/protection/repository/rules/{protection_rule_id}`

Updates a container repository protection rule for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`protection_rule_id` Path, required	Integer	ID of the container repository protection rule

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete`	String	Minimum GitLab access level required to delete container images in the container registry. At least one of `minimum_access_level_for_push` or `minimum_access_level_for_delete` must stay set. To unset the value, use an empty string (`""`) Allowed values: `maintainer` , `owner` , `admin` , ``
`minimum_access_level_for_push`	String	Minimum GitLab access level required to push container images to the container registry. At least one of `minimum_access_level_for_push` or `minimum_access_level_for_delete` must stay set. To unset the value, use an empty string (`""`) Allowed values: `maintainer` , `owner` , `admin` , ``
`repository_path_pattern`	String	Container repository path pattern protected by the protection rule, for example `flight/flight-*` . Wildcard character `*` is allowed

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsContainerRegistryProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Delete a container repository protection rule

`DELETE /api/v4/projects/{id}/registry/protection/repository/rules/{protection_rule_id}`

Deletes a specified container repository protection rule.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`protection_rule_id` Path, required	Integer	ID of the container repository protection rule

Responses

Code	Description	Schema
`204`	204 No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all container registry protection tag rules

`GET /api/v4/projects/{id}/registry/protection/tag/rules`

Lists all container registry protection tag rules for a project. This feature was introduced in GitLab 18.7.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProjectsContainerRegistryProtectionTagRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a container registry protection tag rule

`POST /api/v4/projects/{id}/registry/protection/tag/rules`

Creates a container registry protection tag rule for a project to control who can push or delete container tags. This feature was introduced in GitLab 18.8.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete` Required	String	Minimum GitLab access level required to delete container tags Allowed values: `maintainer`, `owner`, `admin` Minimum length: `1`
`minimum_access_level_for_push` Required	String	Minimum GitLab access level required to push container tags Allowed values: `maintainer`, `owner`, `admin` Minimum length: `1`
`tag_name_pattern` Required	String	Container tag name pattern protected by the protection rule, for example `v*-release`. Wildcard character `*` is allowed

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsContainerRegistryProtectionTagRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Update a container registry protection tag rule

`PATCH /api/v4/projects/{id}/registry/protection/tag/rules/{protection_rule_id}`

Updates a container registry protection tag rule for a project. This feature was introduced in GitLab 18.9.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`protection_rule_id` Path, required	Integer	ID of the container registry protection tag rule

Request body (`application/json`)

Property	Type	Description
`minimum_access_level_for_delete`	String	Minimum GitLab access level required to delete container tags. To unset the value, use an empty string (`""`) Allowed values: `maintainer` , `owner` , `admin` , ``
`minimum_access_level_for_push`	String	Minimum GitLab access level required to push container tags. To unset the value, use an empty string (`""`) Allowed values: `maintainer` , `owner` , `admin` , ``
`tag_name_pattern`	String	Container tag name pattern protected by the protection rule, for example `v*-release` . Wildcard character `*` is allowed

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsContainerRegistryProtectionTagRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Delete a container registry protection tag rule

`DELETE /api/v4/projects/{id}/registry/protection/tag/rules/{protection_rule_id}`

Deletes a container registry protection tag rule from a project. This feature was introduced in GitLab 18.9.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`protection_rule_id` Path, required	Integer	ID of the container registry protection tag rule

Responses

Code	Description	Schema
`204`	Delete a container protection tag rule	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Start a task to recalculate repository size for a project

`POST /api/v4/projects/{id}/repository_size`

This feature was introduced in GitLab 15.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Unauthenticated	—
`404`	Not Found	—

Restore a project marked for deletion

`POST /api/v4/projects/{id}/restore`

Restores a specified project that was marked for deletion.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad Request	—
`404`	Not Found	—

List all runners for a project

`GET /api/v4/projects/{id}/runners`

List all runners available in the project, including from ancestor groups and any allowed shared runners.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`scope` Query	String	Scope of runners to return. Deprecated. Use `type` or `status` instead Allowed values: `specific`, `shared`, `instance_type`, `group_type`, `project_type`, `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
	Integer	

Name	Type	Description
`per_page` Query		Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`403`	No access granted	—
`404`	Not Found	—

Assign a runner to a project

`POST /api/v4/projects/{id}/runners`

Assigns an available project runner to a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`runner_id` Required	Integer	ID of the runner

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiRunner
`400`	Bad Request	—
`403`	Runner is locked	—
`404`	Runner not found	—

Reset runner registration token

`POST /api/v4/projects/{id}/runners/reset_registration_token`

FE version

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Project Not Found	—

Unassign a runner from a project

`DELETE /api/v4/projects/{id}/runners/{runner_id}`

Unassigns a specified project runner from a project. You cannot unassign a runner from the owner project. Use the delete a runner operation instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`runner_id` Path, required	Integer	ID of the runner

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Bad Request	—
`403`	You cannot unassign a runner from the owner project. Delete the runner instead	—
`404`	Runner not found	—
`412`	Precondition Failed	—

Share a project with a group

`POST /api/v4/projects/{id}/share`

Shares a specified project with a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the share expires
`group_access` Required	Integer	Access level to grant to the invited group. For a list of values, see roles Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`group_id` Required	Integer	ID of the group to share the project with Example: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectGroupLink
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete a shared project link in a group

`DELETE /api/v4/projects/{id}/share/{group_id}`

Deletes a shared project link in a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`group_id` Path, required	Integer	ID of the shared group

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`404`	Not found	—

List all groups available to invite to a project

`GET /api/v4/projects/{id}/share_locations`

Lists all groups that can be invited to a project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project
`search` Query	String	Return groups matching the search criteria

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPublicGroupDetails
`400`	Bad Request	—
`404`	Not Found	—

Star a project

`POST /api/v4/projects/{id}/star`

Stars a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`304`	Not modified	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all users who starred a project

`GET /api/v4/projects/{id}/starrers`

Lists all users who starred a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query	String	Search for a specific user by their username or name Example: `user`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Retrieve the statistics of the last 30 days

`GET /api/v4/projects/{id}/statistics`

Retrieves the clone and pull statistics for the last 30 days from a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectDailyStatistics
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—

Retrieve the path to repository storage

`GET /api/v4/projects/{id}/storage`

Retrieves the path to repository storage for a specified project. If you are using Gitaly Cluster (Praefect), see Praefect-generated replica paths instead. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectRepositoryStorage
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all Terraform state protection rules for a project

`GET /api/v4/projects/{id}/terraform/state_protection_rules`

Lists all Terraform state protection rules for a project. This feature was introduced in GitLab 18.11.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesTerraformStateProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a Terraform state protection rule for a project

`POST /api/v4/projects/{id}/terraform/state_protection_rules`

This feature was introduced in GitLab 19.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`allowed_from`	String	Source restriction for write requests. Valid values: anywhere, ci_only, ci_on_ protected_branch_only Allowed values: `anywhere` , `ci_only` , `ci_on_protected_ branch_only` Default: `anywhere`
`minimum_access_level_for_wr ite` Required	String	Minimum GitLab access level required to write to the Terraform state. Valid values: developer, maintainer, owner, admin Allowed values: `developer` , `maintainer` , `owner` , `admin` Minimum length: `1`
`state_name` Required	String	Terraform state name to protect. Maximum 255 characters. Must be unique per project Maximum length: `255`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTerraformStateProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Update a Terraform state protection rule for a project

`PATCH /api/v4/projects/{id}/terraform/state_protection_rules/{terraform_state_protection_rule_id}`

This feature was introduced in GitLab 19.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`terraform_state_protection_rule_id` Path, required	Integer	The ID of the Terraform state protection rule

Request body (`application/json`)

Property	Type	Description
`allowed_from`	String	If defined, write requests must be made from the specific source Allowed values: `anywhere`, `ci_only`, `ci_on_protected_branch_only`
`minimum_access_level_for_write`	String	If defined, sets the minimum GitLab access level required to write to the Terraform state Allowed values: `developer`, `maintainer`, `owner`, `admin`
`state_name`	String	Terraform state name to protect

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTerraformStateProtectionRule
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`422`	Unprocessable Entity	—

Delete a Terraform state protection rule

`DELETE /api/v4/projects/{id}/terraform/state_protection_rules/{terraform_state_protection_rule_id}`

This feature was introduced in GitLab 19.0.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`terraform_state_protection_rule_id` Path, required	Integer	The ID of the Terraform state protection rule

Responses

Code	Description	Schema
`204`	204 No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Transfer a project to another namespace

`PUT /api/v4/projects/{id}/transfer`

Transfers a project to another namespace.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`namespace` Required	String	ID or path of the namespace to transfer the project to Example: `gitlab`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProject
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

List all transferable namespaces for a project

`GET /api/v4/projects/{id}/transfer_locations`

Lists all namespaces where a specified project can be transferred.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query	String	Group name to search for Example: `search`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPublicGroupDetails
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

Unarchive a project

`POST /api/v4/projects/{id}/unarchive`

Unarchives a specified project. You must be an administrator or have the Owner role on the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

Unstar a project

`POST /api/v4/projects/{id}/unstar`

Unstars a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProject
`304`	Not modified	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not Found	—

List all uploads

`GET /api/v4/projects/{id}/uploads`

Lists all uploads of a project sorted by `created_at` in descending order. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesMarkdown UploadAdmin
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Create an upload

`POST /api/v4/projects/{id}/uploads`

Creates an upload.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The attachment file to be uploaded

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectUpload
`400`	Bad Request	—
`404`	Not found	—

Workhorse authorize the file upload

`POST /api/v4/projects/{id}/uploads/authorize`

This feature was introduced in GitLab 13.11

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not found	—

Download an uploaded file by secret and filename

`GET /api/v4/projects/{id}/uploads/{secret}/{filename}`

Downloads an uploaded file with a specified secret and filename. You must have the Guest, Planner, Reporter, Developer, Maintainer, or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`secret` Path, required	String	Secret of the upload
`filename` Path, required	String	Filename of the upload

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete an uploaded file by secret and filename

`DELETE /api/v4/projects/{id}/uploads/{secret}/{filename}`

Deletes an uploaded file with a specified secret and filename. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`secret` Path, required	String	Secret of the upload
`filename` Path, required	String	Filename of the upload

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

Download an uploaded file by ID

`GET /api/v4/projects/{id}/uploads/{upload_id}`

Downloads an uploaded file with a specified ID. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`upload_id` Path, required	Integer	ID of the upload

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete an uploaded file by ID

`DELETE /api/v4/projects/{id}/uploads/{upload_id}`

Deletes an uploaded file with a specified ID. You must have the Maintainer or Owner role for the project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`upload_id` Path, required	Integer	ID of the upload

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—

List all members of a project

`GET /api/v4/projects/{id}/users`

Lists all members with access to a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query	String	Search for a specific user by their username or name Example: `user`
`skip_users` Query	Array of integers	Filter out members with the specified IDs
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Get projects associated with a runner

`GET /api/v4/runners/{id}/projects`

Get a paginated list of all projects associated with the specified runner. Access is restricted based on user permissions.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicProjectDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

List all projects contributions for a user

`GET /api/v4/users/{user_id}/contributed_projects`

Lists all contributions to visible projects for a specified user. Returns only contributions in the past year.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user
`order_by` Query	String	Return projects ordered by field. storage_size, repository_size, wiki_size, packages_size are only available to admins. Similarity is available when searching and is limited to projects the user has access to Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count`, `storage_size`, `repository_size`, `wiki_size`, `packages_size` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBasic ProjectDetails
`400`	Bad Request	—
`404`	404 User Not Found	—

List all personal projects for a user

`GET /api/v4/users/{user_id}/projects`

Lists all personal projects for a specified user. Does not return group or subgroup projects. If the user profile is private, returns only an empty list.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user
`order_by` Query	String	Return projects ordered by field. storage_size, repository_size, wiki_size, packages_size are only available to admins. Similarity is available when searching and is limited to projects the user has access to Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count`, `storage_size`, `repository_size`, `wiki_size`, `packages_size` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`archived` Query	Boolean	If `true`, returns only archived projects
`visibility` Query	String	Return projects with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return a list of projects with a `path`, `name`, or `description` matching the search criteria (case-insensitive, substring match). Multiple terms can be provided, separated by an escaped space, either `+` or `%20`, and are ANDed together. Example: `one+two` matches

Name	Type	Description
		substrings `one` and `two` (in any order)
`search_namespaces` Query	Boolean	If `true`, includes ancestor namespaces when matching search criteria
`owned` Query	Boolean	If `true`, returns only projects explicitly owned by the current user Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`imported` Query	Boolean	If `true`, returns only projects that were imported from external systems by the authenticated user Default: `false`
`membership` Query	Boolean	If `true`, returns only projects that the current user is a member of Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`with_programming_language` Query	String	Return projects that use the given programming language
`min_access_level` Query	Integer	Return results where the current user has at least the specified access level. For a list of values, see roles

Name	Type	Description
		Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`id_after` Query	Integer	Return projects with IDs greater than the specified ID
`id_before` Query	Integer	Return projects with IDs less than the specified ID
`last_activity_after` Query	String (date-time)	Return projects with last activity after the specified time
`last_activity_before` Query	String (date-time)	Return projects with last activity before the specified time
`repository_storage` Query	String	Return projects stored on the specified repository storage shard. Administrators only
`topic` Query	Array of strings	Comma-separated list of topics. Return projects having all topics
`topic_id` Query	Integer	Filter projects by the given topic ID
`updated_before` Query	String (date-time)	Return projects last updated on or before the specified time
`updated_after` Query	String (date-time)	Return projects last updated on or after the specified time
`include_pending_delete` Query	Boolean	If `true`, includes projects that are pending deletion in the response. Administrators only
`marked_for_deletion_on` Query	String (date)	Return projects that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only projects that are not archived and not marked for deletion

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`statistics` Query	Boolean	If `true`, includes project statistics. Available only to users with the Reporter, Developer, Maintainer, or Owner role Default: `false`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBasic ProjectDetails
`400`	Bad Request	—
`404`	404 User Not Found	—

List all projects starred by a user

`GET /api/v4/users/{user_id}/starred_projects`

Lists all visible projects starred by a specified user. Unauthenticated requests return only public projects.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user
`order_by` Query	String	Return projects ordered by field. storage_size, repository_size, wiki_size, packages_size are only available to admins. Similarity is available when searching and is limited to projects the user has access to Allowed values: `id`, `name`, `path`, `created_at`, `updated_at`, `last_activity_at`, `similarity`, `star_count`, `storage_size`, `repository_size`, `wiki_size`, `packages_size` Default: `created_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`archived` Query	Boolean	If `true`, returns only archived projects
`visibility` Query	String	Return projects with the specified visibility Allowed values: `private`, `internal`, `public`
`search` Query	String	Return a list of projects with a `path`, `name`, or `description` matching the search criteria (case-insensitive, substring match). Multiple terms can be provided, separated by an escaped space, either `+` or `%20`, and are ANDed together. Example: `one+two` matches

Name	Type	Description
		substrings `one` and `two` (in any order)
`search_namespaces` Query	Boolean	If `true`, includes ancestor namespaces when matching search criteria
`owned` Query	Boolean	If `true`, returns only projects explicitly owned by the current user Default: `false`
`starred` Query	Boolean	If `true`, returns only projects starred by the current user Default: `false`
`imported` Query	Boolean	If `true`, returns only projects that were imported from external systems by the authenticated user Default: `false`
`membership` Query	Boolean	If `true`, returns only projects that the current user is a member of Default: `false`
`with_issues_enabled` Query	Boolean	If `true`, returns only projects that have the issues feature enabled Default: `false`
`with_merge_requests_enabled` Query	Boolean	If `true`, returns only projects that have the merge requests feature enabled Default: `false`
`with_programming_language` Query	String	Return projects that use the given programming language
`min_access_level` Query	Integer	Return results where the current user has at least the specified access level. For a list of values, see roles

Name	Type	Description
		Allowed values: `10`, `15`, `20`, `25`, `30`, `40`, `50`
`id_after` Query	Integer	Return projects with IDs greater than the specified ID
`id_before` Query	Integer	Return projects with IDs less than the specified ID
`last_activity_after` Query	String (date-time)	Return projects with last activity after the specified time
`last_activity_before` Query	String (date-time)	Return projects with last activity before the specified time
`repository_storage` Query	String	Return projects stored on the specified repository storage shard. Administrators only
`topic` Query	Array of strings	Comma-separated list of topics. Return projects having all topics
`topic_id` Query	Integer	Filter projects by the given topic ID
`updated_before` Query	String (date-time)	Return projects last updated on or before the specified time
`updated_after` Query	String (date-time)	Return projects last updated on or after the specified time
`include_pending_delete` Query	Boolean	If `true`, includes projects that are pending deletion in the response. Administrators only
`marked_for_deletion_on` Query	String (date)	Return projects that are marked for deletion on this date. Premium and Ultimate only. Introduced in GitLab 17.1
`active` Query	Boolean	If `true`, returns only projects that are not archived and not marked for deletion

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`simple` Query	Boolean	If `true`, returns only limited fields for each project. Without authentication, only limited fields are returned regardless of this setting Default: `false`
`statistics` Query	Boolean	If `true`, includes project statistics. Available only to users with the Reporter, Developer, Maintainer, or Owner role Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBasicProjectDetails
`400`	Bad Request	—
`404`	404 User Not Found	—

Validate CODEOWNERS file by project full path

`GET /api/v4/validate-code-owner-file`

Parameters

Name	Type	Description
`project_path` Query, required	String	—
`ref` Query	String	—
`file_path` Query	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCiResetTokenResult

Property	Type	Description
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesContainerExpirationPolicy

Property	Type	Description
`cadence`	String	—
`enabled`	String	—
`keep_n`	String	—
`name_regex`	String	—
`name_regex_keep`	String	—
`next_run_at`	String	—
`older_than`	String	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesGroupAccess

Property	Type	Description
`access_level`	String	—
`notification_level`	String	—

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesIssue

Property	Type	Description
`_links`	Object	—
`_links.award_emoji`	String	Example: `http://example.com/api/v4/projects/1/issues/2/award_emoji`
`_links.closed_as_duplicate_of`	String	Example: `http://example.com/api/v4/projects/1/issues/75`
`_links.notes`	String	Example: `http://example.com/api/v4/projects/1/issues/2/notes`
`_links.project`	String	Example: `http://example.com/api/v4/projects/1`
`_links.self`	String	Example: `http://example.com/api/v4/projects/1/issues/2`
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`

Property	Type	Description
`has_tasks`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`moved_to_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `4`
`references`	APIEntitiesIssuableReferences	—
`service_desk_reply_to`	String	Example: `user@example.com`
`severity`	String	The severity of the issue Allowed values: `UNKNOWN`, `LOW`, `MEDIUM`, `HIGH`, `CRITICAL`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`subscribed`	Boolean	Example: `false`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`task_status`	String	Example: `2 of 4 tasks completed`
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`

Property	Type	Description
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesMarkdownUploadAdmin

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`filename`	String	Example: `image.png`
`id`	Integer (int64)	Example: `1`
`size`	Integer	Example: `1024`
`uploaded_by`	APIEntitiesUserSafe	—

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMergeRequestBasic

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`draft`	Boolean	—
`force_remove_source_branch`	Boolean	—
`has_conflicts`	Boolean	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`

Property	Type	Description
`labels`	Array of strings	—
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`

Property	Type	Description
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesProject

Property	Type	Description
`_links`	Object	—
`_links.cluster_agents`	String	Example: `https://gitlab.example.com/api/v4/projects/4/cluster_agents`
`_links.events`	String	Example: `https://gitlab.example.com/api/v4/projects/4/events`
`_links.issues`	String	Example: `https://gitlab.example.com/api/v4/projects/4/issues`
`_links.labels`	String	Example: `https://gitlab.example.com/api/v4/projects/4/labels`
`_links.members`	String	Example: `https://gitlab.example.com/api/v4/projects/4/members`
`_links.merge_requests`	String	Example: `https://gitlab.example.com/api/v4/projects/4/merge_requests`
`_links.repo_branches`	String	Example: `https://gitlab.example.com/api/v4/projects/4/repository/branches`
`_links.self`	String	Example: `https://gitlab.example.com/api/v4/projects/4`
`allow_merge_on_skipped_pipeline`	Boolean	—
`allow_pipeline_trigger_approve_deployment`	Boolean	—
`analytics_access_level`	String	Example: `enabled`
`archived`	Boolean	—
	String	Example: `enabled`

Property	Type	Description
`auto_cancel_pending_pipelines`		
`auto_devops_deploy_strategy`	String	Example: `continuous`
`auto_devops_enabled`	Boolean	—
`autoclose_referenced_issues`	Boolean	—
`automatic_rebase_enabled`	Boolean	—
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`build_git_strategy`	String	Example: `fetch`
`build_timeout`	Integer	Example: `3600`
`builds_access_level`	String	Example: `enabled`
`can_create_merge_request_in`	Boolean	—
`ci_allow_fork_pipelines_to_run_in_parent_project`	Boolean	—
`ci_config_path`	String	Example: ``
`ci_default_git_depth`	Integer	Example: `20`
`ci_delete_pipelines_in_seconds`	Integer	Example: `86400`
`ci_display_pipeline_variables`	Boolean	—
`ci_forward_deployment_enabled`	Boolean	—
`ci_forward_deployment_rollback_allowed`	Boolean	—
`ci_id_token_sub_claim_components`	Array of strings	—

Property	Type	Description
`ci_job_token_scope_enabled`	Boolean	—
`ci_pipeline_variables_minimum_override_role`	String	—
`ci_push_repository_for_job_token_allowed`	Boolean	—
`ci_separated_caches`	Boolean	—
`ci_skip_branch_pipelines_for_mrs`	Boolean	—
`container_expiration_policy`	APIEntitiesContainerExpirationPolicy	—
`container_registry_access_level`	String	Example: `enabled`
`container_registry_enabled`	Boolean	—
`container_registry_image_prefix`	String	Example: `registry.gitlab.example.com/gitlab/gitlab-client`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`creator_id`	Integer (int64)	Example: `1`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`description_html`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`empty_repo`	Boolean	—
`enforce_auth_checks_on_uploads`	Boolean	—

Property	Type	Description
`environments_access_level`	String	Example: `enabled`
`fe_project_setting_attributes`	Object	—
`feature_flags_access_level`	String	Example: `enabled`
`forked_from_project`	APIEntitiesBasicProjectDetails	—
`forking_access_level`	String	Example: `enabled`
`forks_count`	Integer	Example: `1`
`group_runners_enabled`	Boolean	—
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`import_error`	String	Example: `Import error`
`import_status`	String	Example: `none`
`import_type`	String	Example: `git`
`import_url`	String	Example: `https://gitlab.com/gitlab/gitlab.git`
`infrastructure_access_level`	String	Example: `enabled`
`issue_branch_template`	String	Example: `%(title)`
`issues_access_level`	String	Example: `enabled`
`issues_enabled`	Boolean	—
`jaga_task_required_to_merge`	Boolean	—
`jobs_enabled`	Boolean	—
`keep_latest_artifact`	Boolean	—
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`

Property	Type	Description
`lfs_enabled`	Boolean	—
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`marked_for_deletion_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`marked_for_deletion_on`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`max_artifacts_size`	Integer	—
`max_pipelines_per_merge_train`	Integer	—
`merge_commit_template`	String	Example: `%(title)`
`merge_method`	String	Example: `merge`
`merge_pipelines_enabled`	Boolean	—
`merge_request_title_regex`	String	—
`merge_request_title_regex_description`	String	—
`merge_requests_access_level`	String	Example: `enabled`
`merge_requests_enabled`	Boolean	—
`merge_train_enforced`	Boolean	—
`merge_trains_enabled`	Boolean	—
`merge_trains_skip_train_allowed`	Boolean	—
`model_experiments_access_level`	String	Example: `enabled`
`model_registry_access_level`	String	Example: `enabled`

Property	Type	Description
`monitor_access_level`	String	Example: `enabled`
`mr_default_target_self`	Boolean	—
`mr_default_title_template`	String	Example: `%(source_branch)`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	—
`only_allow_merge_if_pipeline_succeeds`	Boolean	—
`open_issues_count`	Integer	Example: `1`
`owner`	APIEntitiesUserBasic	—
`package_registry_access_level`	String	Example: `enabled`
`packages_enabled`	Boolean	—
`pages_access_level`	String	Example: `enabled`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`printing_merge_request_link_enabled`	Boolean	—
`protect_merge_request_pipelines`	Boolean	—
`public_jobs`	Boolean	—
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`releases_access_level`	String	Example: `enabled`
`remove_source_branch_after_merge`	Boolean	—
`repository_access_level`	String	Example: `enabled`
`repository_object_format`	String	Example: `sha1`
`repository_storage`	String	Example: `default`
`request_access_enabled`	Boolean	—
`resolve_outdated_diff_discussions`	Boolean	—
`resource_group_default_process_mode`	String	Example: `unordered`
`restrict_user_defined_variables`	Boolean	—
`runner_token_expiration_interval`	Integer	Example: `3600`
`runners_token`	String	Example: `b8547b1dc37721d05889db52fa2f02`
`security_and_compliance_access_level`	String	Example: `enabled`
`service_desk_address`	String	Example: `address@example.com`
`service_desk_enabled`	Boolean	—
`shared_runners_enabled`	Boolean	—
`shared_with_groups`	Array of objects	—
`show_diff_preview_in_email`	Boolean	—
`snippets_access_level`	String	Example: `enabled`
`snippets_enabled`	Boolean	—

Property	Type	Description
`squash_commit_template`	String	Example: `%(source_branch)`
`squash_option`	String	Example: `default_off`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`statistics`	APIEntitiesProjectStatistics	—
`suggestion_commit_message`	String	Example: `Suggestion message`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`updated_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`visibility`	String	Example: `public`
`warn_about_potentially_unwanted_characters`	Boolean	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`
`wiki_access_level`	String	Example: `enabled`
`wiki_enabled`	Boolean	—

APIEntitiesProjectAccess

Property	Type	Description
`access_level`	String	—
`notification_level`	String	—

APIEntitiesProjectDailyFetches

Property	Type	Description
`count`	Integer	Example: `3`
`date`	String (date)	Example: `2022-01-01`

APIEntitiesProjectDailyStatistics

Property	Type	Description
`fetches`	Object	—
`fetches.days`	Array of APIEntitiesProjectDailyFetches	—
`fetches.total`	Integer	Example: `3`

APIEntitiesProjectGroupLink

Property	Type	Description
`expires_at`	String (date)	Example: `2016-09-26`
`group_access`	Integer	Example: `10`
`group_id`	Integer (int64)	Example: `1`
`id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `1`

APIEntitiesProjectRepositoryStorage

Property	Type	Description
`created_at`	String (date-time)	Example: `2012-10-12T17:04:47Z`
`disk_path`	String	Example: `@hashed/6b/86/6b86b273ff34fce19d6b804eff5a3f5747ada4eaa22f1d49c01e52ddb7875b4b`
`project_id`	Integer (int64)	Example: `1`
`repository_storage`	String	Example: `default`

APIEntitiesProjectsContainerRegistryProtectionRule

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`minimum_access_level_for_delete`	String	Example: `maintainer`
`minimum_access_level_for_push`	String	Example: `maintainer`
`project_id`	Integer (int64)	Example: `1`
`repository_path_pattern`	String	Example: `flightjs/flight0`

APIEntitiesProjectsContainerRegistryProtectionTagRule

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`minimum_access_level_for_delete`	String	Example: `maintainer`
`minimum_access_level_for_push`	String	Example: `maintainer`
`project_id`	Integer (int64)	Example: `123`
`tag_name_pattern`	String	Example: `v*-release`

APIEntitiesProjectsPackagesProtectionRule

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`minimum_access_level_for_delete`	String	Example: `owner`
`minimum_access_level_for_push`	String	Example: `maintainer`
`package_name_pattern`	String	Example: `flightjs/flight`
`package_type`	String	Example: `npm`
`project_id`	Integer (int64)	Example: `1`

APIEntitiesProjectStatistics

Property	Type	Description
`commit_count`	Integer	Example: `37`
`container_registry_size`	Integer	Example: `0`
`job_artifacts_size`	Integer	Example: `0`
`lfs_objects_size`	Integer	Example: `0`
`packages_size`	Integer	Example: `0`
`pipeline_artifacts_size`	Integer	Example: `0`
`repository_size`	Integer	Example: `1.03809e+06`
`snippets_size`	Integer	Example: `0`
`storage_size`	Integer	Example: `1.03809e+06`
`uploads_size`	Integer	Example: `0`
`wiki_size`	Integer	Example: `0`

APIEntitiesProjectsWithAccessAndCatalogSetting

Property	Type	Description
`_links`	Object	—
`_links.cluster_agents`	String	Example: `https://gitlab.example.com/api/v4/projects/4/cluster_agents`
`_links.events`	String	Example: `https://gitlab.example.com/api/v4/projects/4/events`
`_links.issues`	String	Example: `https://gitlab.example.com/api/v4/projects/4/issues`
`_links.labels`	String	Example: `https://gitlab.example.com/api/v4/projects/4/labels`
`_links.members`	String	Example: `https://gitlab.example.com/api/v4/projects/4/members`
`_links.merge_requests`	String	Example: `https://gitlab.example.com/api/v4/projects/4/merge_requests`
`_links.repo_branches`	String	Example: `https://gitlab.example.com/api/v4/projects/4/repository/branches`
`_links.self`	String	Example: `https://gitlab.example.com/api/v4/projects/4`
`allow_merge_on_skipped_pipeline`	Boolean	—
`allow_pipeline_trigger_approve_deployment`	Boolean	—
`analytics_access_level`	String	Example: `enabled`
`archived`	Boolean	—
	String	Example: `enabled`

Property	Type	Description
`auto_cancel_pending_pipelines`		
`auto_devops_deploy_strategy`	String	Example: `continuous`
`auto_devops_enabled`	Boolean	—
`autoclose_referenced_issues`	Boolean	—
`automatic_rebase_enabled`	Boolean	—
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`build_git_strategy`	String	Example: `fetch`
`build_timeout`	Integer	Example: `3600`
`builds_access_level`	String	Example: `enabled`
`can_create_merge_request_in`	Boolean	—
`ci_allow_fork_pipelines_to_run_in_parent_project`	Boolean	—
`ci_config_path`	String	Example: ``
`ci_default_git_depth`	Integer	Example: `20`
`ci_delete_pipelines_in_seconds`	Integer	Example: `86400`
`ci_display_pipeline_variables`	Boolean	—
`ci_forward_deployment_enabled`	Boolean	—
`ci_forward_deployment_rollback_allowed`	Boolean	—
`ci_id_token_sub_claim_components`	Array of strings	—

Property	Type	Description
`ci_job_token_scope_enabled`	Boolean	—
`ci_pipeline_variables_minimum_override_role`	String	—
`ci_push_repository_for_job_token_allowed`	Boolean	—
`ci_separated_caches`	Boolean	—
`ci_skip_branch_pipelines_for_mrs`	Boolean	—
`cicd_catalog_enabled`	Boolean	—
`container_expiration_policy`	APIEntitiesContainerExpirationPolicy	—
`container_registry_access_level`	String	Example: `enabled`
`container_registry_enabled`	Boolean	—
`container_registry_image_prefix`	String	Example: `registry.gitlab.example.com/gitlab/gitlab-client`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`creator_id`	Integer (int64)	Example: `1`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`description_html`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`empty_repo`	Boolean	—
	Boolean	—

Property	Type	Description
`enforce_auth_checks_on_uploads`		
`environments_access_level`	String	Example: `enabled`
`fe_project_setting_attributes`	Object	—
`feature_flags_access_level`	String	Example: `enabled`
`forked_from_project`	APIEntitiesBasicProjectDetails	—
`forking_access_level`	String	Example: `enabled`
`forks_count`	Integer	Example: `1`
`group_runners_enabled`	Boolean	—
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`import_error`	String	Example: `Import error`
`import_status`	String	Example: `none`
`import_type`	String	Example: `git`
`import_url`	String	Example: `https://gitlab.com/gitlab/gitlab.git`
`infrastructure_access_level`	String	Example: `enabled`
`issue_branch_template`	String	Example: `%(title)`
`issues_access_level`	String	Example: `enabled`
`issues_enabled`	Boolean	—
`jaga_task_required_to_merge`	Boolean	—
`jobs_enabled`	Boolean	—
`keep_latest_artifact`	Boolean	—

Property	Type	Description
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`lfs_enabled`	Boolean	—
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`marked_for_deletion_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`marked_for_deletion_on`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`max_artifacts_size`	Integer	—
`max_pipelines_per_merge_train`	Integer	—
`merge_commit_template`	String	Example: `%(title)`
`merge_method`	String	Example: `merge`
`merge_pipelines_enabled`	Boolean	—
`merge_request_title_regex`	String	—
`merge_request_title_regex_description`	String	—
`merge_requests_access_level`	String	Example: `enabled`
`merge_requests_enabled`	Boolean	—
`merge_train_enforced`	Boolean	—
`merge_trains_enabled`	Boolean	—
`merge_trains_skip_train_allowed`	Boolean	—
`model_experiments_access_level`	String	Example: `enabled`

Property	Type	Description
`model_registry_access_level`	String	Example: `enabled`
`monitor_access_level`	String	Example: `enabled`
`mr_default_target_self`	Boolean	—
`mr_default_title_template`	String	Example: `%(source_branch)`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	—
`only_allow_merge_if_pipeline_succeeds`	Boolean	—
`open_issues_count`	Integer	Example: `1`
`owner`	APIEntitiesUserBasic	—
`package_registry_access_level`	String	Example: `enabled`
`packages_enabled`	Boolean	—
`pages_access_level`	String	Example: `enabled`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`permissions`	Object	—
`permissions.group_access`	APIEntitiesGroupAccess	—
`permissions.project_access`	APIEntitiesProjectAccess	—
`printing_merge_request_link_enabled`	Boolean	—
	Boolean	—

Property	Type	Description
`protect_merge_request_pipelines`		
`public_jobs`	Boolean	—
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`
`releases_access_level`	String	Example: `enabled`
`remove_source_branch_after_merge`	Boolean	—
`repository_access_level`	String	Example: `enabled`
`repository_object_format`	String	Example: `sha1`
`repository_storage`	String	Example: `default`
`request_access_enabled`	Boolean	—
`resolve_outdated_diff_discussions`	Boolean	—
`resource_group_default_process_mode`	String	Example: `unordered`
`restrict_user_defined_variables`	Boolean	—
`runner_token_expiration_interval`	Integer	Example: `3600`
`runners_token`	String	Example: `b8547b1dc37721d05889db52fa2f02`
`security_and_compliance_access_level`	String	Example: `enabled`
`service_desk_address`	String	Example: `address@example.com`
`service_desk_enabled`	Boolean	—
`shared_runners_enabled`	Boolean	—

Property	Type	Description
`shared_with_groups`	Array of objects	—
`show_diff_preview_in_email`	Boolean	—
`snippets_access_level`	String	Example: `enabled`
`snippets_enabled`	Boolean	—
`squash_commit_template`	String	Example: `%(source_branch)`
`squash_option`	String	Example: `default_off`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`statistics`	APIEntitiesProjectStatistics	—
`suggestion_commit_message`	String	Example: `Suggestion message`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`updated_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`visibility`	String	Example: `public`
`warn_about_potentially_unwanted_characters`	Boolean	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`
`wiki_access_level`	String	Example: `enabled`
`wiki_enabled`	Boolean	—

APIEntitiesProjectsWithCatalogSetting

Property	Type	Description
`_links`	Object	—
`_links.cluster_agents`	String	Example: `https://gitlab.example.com/api/v4/projects/4/cluster_agents`
`_links.events`	String	Example: `https://gitlab.example.com/api/v4/projects/4/events`
`_links.issues`	String	Example: `https://gitlab.example.com/api/v4/projects/4/issues`
`_links.labels`	String	Example: `https://gitlab.example.com/api/v4/projects/4/labels`
`_links.members`	String	Example: `https://gitlab.example.com/api/v4/projects/4/members`
`_links.merge_requests`	String	Example: `https://gitlab.example.com/api/v4/projects/4/merge_requests`
`_links.repo_branches`	String	Example: `https://gitlab.example.com/api/v4/projects/4/repository/branches`
`_links.self`	String	Example: `https://gitlab.example.com/api/v4/projects/4`
`allow_merge_on_skipped_pipeline`	Boolean	—
`allow_pipeline_trigger_approve_deployment`	Boolean	—
`analytics_access_level`	String	Example: `enabled`
`archived`	Boolean	—
	String	Example: `enabled`

Property	Type	Description
`auto_cancel_pending_pipelines`		
`auto_devops_deploy_strategy`	String	Example: `continuous`
`auto_devops_enabled`	Boolean	—
`autoclose_referenced_issues`	Boolean	—
`automatic_rebase_enabled`	Boolean	—
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`build_git_strategy`	String	Example: `fetch`
`build_timeout`	Integer	Example: `3600`
`builds_access_level`	String	Example: `enabled`
`can_create_merge_request_in`	Boolean	—
`ci_allow_fork_pipelines_to_run_in_parent_project`	Boolean	—
`ci_config_path`	String	Example: ``
`ci_default_git_depth`	Integer	Example: `20`
`ci_delete_pipelines_in_seconds`	Integer	Example: `86400`
`ci_display_pipeline_variables`	Boolean	—
`ci_forward_deployment_enabled`	Boolean	—
`ci_forward_deployment_rollback_allowed`	Boolean	—
`ci_id_token_sub_claim_components`	Array of strings	—

Property	Type	Description
`ci_job_token_scope_enabled`	Boolean	—
`ci_pipeline_variables_minimum_override_role`	String	—
`ci_push_repository_for_job_token_allowed`	Boolean	—
`ci_separated_caches`	Boolean	—
`ci_skip_branch_pipelines_for_mrs`	Boolean	—
`cicd_catalog_enabled`	Boolean	—
`container_expiration_policy`	APIEntitiesContainerExpirationPolicy	—
`container_registry_access_level`	String	Example: `enabled`
`container_registry_enabled`	Boolean	—
`container_registry_image_prefix`	String	Example: `registry.gitlab.example.com/gitlab/gitlab-client`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`creator_id`	Integer (int64)	Example: `1`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`description_html`	String	—
`emails_disabled`	Boolean	—
`emails_enabled`	Boolean	—
`empty_repo`	Boolean	—
	Boolean	—

Property	Type	Description
`enforce_auth_checks_on_uploads`		
`environments_access_level`	String	Example: `enabled`
`fe_project_setting_attributes`	Object	—
`feature_flags_access_level`	String	Example: `enabled`
`forked_from_project`	APIEntitiesBasicProjectDetails	—
`forking_access_level`	String	Example: `enabled`
`forks_count`	Integer	Example: `1`
`group_runners_enabled`	Boolean	—
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`import_error`	String	Example: `Import error`
`import_status`	String	Example: `none`
`import_type`	String	Example: `git`
`import_url`	String	Example: `https://gitlab.com/gitlab/gitlab.git`
`infrastructure_access_level`	String	Example: `enabled`
`issue_branch_template`	String	Example: `%(title)`
`issues_access_level`	String	Example: `enabled`
`issues_enabled`	Boolean	—
`jaga_task_required_to_merge`	Boolean	—
`jobs_enabled`	Boolean	—
`keep_latest_artifact`	Boolean	—

Property	Type	Description
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`lfs_enabled`	Boolean	—
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`marked_for_deletion_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`marked_for_deletion_on`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`max_artifacts_size`	Integer	—
`max_pipelines_per_merge_train`	Integer	—
`merge_commit_template`	String	Example: `%(title)`
`merge_method`	String	Example: `merge`
`merge_pipelines_enabled`	Boolean	—
`merge_request_title_regex`	String	—
`merge_request_title_regex_description`	String	—
`merge_requests_access_level`	String	Example: `enabled`
`merge_requests_enabled`	Boolean	—
`merge_train_enforced`	Boolean	—
`merge_trains_enabled`	Boolean	—
`merge_trains_skip_train_allowed`	Boolean	—
`model_experiments_access_level`	String	Example: `enabled`

Property	Type	Description
`model_registry_access_level`	String	Example: `enabled`
`monitor_access_level`	String	Example: `enabled`
`mr_default_target_self`	Boolean	—
`mr_default_title_template`	String	Example: `%(source_branch)`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`only_allow_merge_if_all_discussions_are_resolved`	Boolean	—
`only_allow_merge_if_pipeline_succeeds`	Boolean	—
`open_issues_count`	Integer	Example: `1`
`owner`	APIEntitiesUserBasic	—
`package_registry_access_level`	String	Example: `enabled`
`packages_enabled`	Boolean	—
`pages_access_level`	String	Example: `enabled`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`printing_merge_request_link_enabled`	Boolean	—
`protect_merge_request_pipelines`	Boolean	—
`public_jobs`	Boolean	—
`readme_url`	String	

Property	Type	Description
		Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`
`releases_access_level`	String	Example: `enabled`
`remove_source_branch_after_merge`	Boolean	—
`repository_access_level`	String	Example: `enabled`
`repository_object_format`	String	Example: `sha1`
`repository_storage`	String	Example: `default`
`request_access_enabled`	Boolean	—
`resolve_outdated_diff_discussions`	Boolean	—
`resource_group_default_process_mode`	String	Example: `unordered`
`restrict_user_defined_variables`	Boolean	—
`runner_token_expiration_interval`	Integer	Example: `3600`
`runners_token`	String	Example: `b8547b1dc37721d05889db52fa2f02`
`security_and_compliance_access_level`	String	Example: `enabled`
`service_desk_address`	String	Example: `address@example.com`
`service_desk_enabled`	Boolean	—
`shared_runners_enabled`	Boolean	—
`shared_with_groups`	Array of objects	—
`show_diff_preview_in_email`	Boolean	—

Property	Type	Description
`snippets_access_level`	String	Example: `enabled`
`snippets_enabled`	Boolean	—
`squash_commit_template`	String	Example: `%(source_branch)`
`squash_option`	String	Example: `default_off`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`statistics`	APIEntitiesProjectStatistics	—
`suggestion_commit_message`	String	Example: `Suggestion message`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`updated_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`visibility`	String	Example: `public`
`warn_about_potentially_unwanted_characters`	Boolean	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`
`wiki_access_level`	String	Example: `enabled`
`wiki_enabled`	Boolean	—

APIEntitiesProjectUpload

Property	Type	Description
`alt`	String	Example: `filename`
`full_path`	String	Example: `/-/project/1/uploads/secret/filename`
`id`	Integer (int64)	Example: `1`
`markdown`	String	Example: `[filename](/uploads/secret/filename)`
`url`	String	Example: `/uploads/secret/filename`

APIEntitiesPublicGroupDetails

Property	Type	Description
`avatar_url`	String	Example: `http://gitlab.example.com/uploads/group/avatar/1/avatar.png`
`full_name`	String	Example: `Foobar Group`
`full_path`	String	Example: `foo-bar`
`id`	Integer (int64)	—
`name`	String	Example: `Diaspora`
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesTerraformStateProtectionRule

Property	Type	Description
`allowed_from`	String	Example: `ci_only`
`id`	Integer (int64)	Example: `1`
`minimum_access_level_for_write`	String	Example: `maintainer`
`project_id`	Integer (int64)	Example: `1`
`state_name`	String	Example: `production`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserSafe

Property	Type	Description
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`username`	String	Example: `admin`

Projects job token scope

Use this API to interact with CI/CD job token scopes.

> All requests to the CI/CD job token scope API endpoint must be authenticated. The authenticated user must have the Maintainer or Owner role for the project.

Retrieve the CI/CD job token access settings for a project

`GET /api/v4/projects/{id}/job_token_scope`

Retrieves the CI/CD job token access settings (job token scope) of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectJobTokenScope
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Update the CI/CD job token access settings for a project

`PATCH /api/v4/projects/{id}/job_token_scope`

Updates the **Authorized groups and projects** setting (job token scope) of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`enabled` Required	Boolean	Indicates CI/CD job tokens generated in other projects have restricted access to this project

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all projects in a CI/CD job token allowlist

`GET /api/v4/projects/{id}/job_token_scope/allowlist`

Lists all projects in the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicProjectDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Add a project to a CI/CD job token allowlist

`POST /api/v4/projects/{id}/job_token_scope/allowlist`

Adds a project to the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project Example: `1`

Request body (`application/json`)

Property	Type	Description
`target_project_id` Required	Integer	ID of the project to add to the CI/CD job token inbound allowlist Example: `2`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectScopeLink
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a project from a CI/CD job token allowlist

`DELETE /api/v4/projects/{id}/job_token_scope/allowlist/{target_project_id}`

Deletes a project from the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project Example: `1`
`target_project_id` Path, required	Integer	ID of the project to remove from the CI/CD job token inbound allowlist Example: `2`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all groups in a CI/CD job token allowlist

`GET /api/v4/projects/{id}/job_token_scope/groups_allowlist`

Lists all groups in the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicGroupDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Add a group to a CI/CD job token allowlist

`POST /api/v4/projects/{id}/job_token_scope/groups_allowlist`

Adds a group to the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project Example: `1`

Request body (`application/json`)

Property	Type	Description
`target_group_id` Required	Integer	ID of the group to add to the CI/CD job token groups allowlist Example: `2`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupScopeLink
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a group from a CI/CD job token allowlist

`DELETE /api/v4/projects/{id}/job_token_scope/groups_allowlist/{target_group_id}`

Deletes a group from the CI/CD job token allowlist of a specified project.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the project Example: `1`
`target_group_id` Path, required	Integer	ID of the group to remove from the CI/CD job token groups allowlist Example: `2`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicGroupDetails

Property	Type	Description
`id`	Integer (int64)	—
`name`	String	Example: `Diaspora`
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesGroupScopeLink

Property	Type	Description
`source_project_id`	Integer (int64)	—
`target_group_id`	Integer (int64)	—

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesProjectJobTokenScope

Property	Type	Description
`inbound_enabled`	Boolean	—
`outbound_enabled`	Boolean	—

APIEntitiesProjectScopeLink

Property	Type	Description
`source_project_id`	Integer (int64)	—
`target_project_id`	Integer (int64)	—

Protected branches

Use this API to manage protected branches for a project, and the protected branch settings that all projects in a group inherit.

GitLab Premium and GitLab Ultimate support more granular protections for pushing to branches. Administrators can grant permission to modify and push to protected branches only to deploy keys, instead of specific users.

> Group protected branch settings are restricted to top-level groups. They support only valid access levels, and cannot name individual users or groups.

List all protected branches

`GET /api/v4/projects/{id}/protected_branches`

Lists all protected branches for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`search` Query	String	Name or part of the name of protected branches to search for Example: `mai`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProtected Branch
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—

Protect repository branches

`POST /api/v4/projects/{id}/protected_branches`

Protects a specified repository branch or several project repository branches using a wildcard protected branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`

Request body (`application/json`)

Property	Type	Description
`allow_force_push`	Boolean	If `true`, members who can push to this branch can also force push Default: `false`
`allowed_to_merge`	Array of objects	Array of merge access levels, with each described by a hash of the form `{user_id: integer}`, `{group_id: integer}`, or `{access_level: integer}`
`allowed_to_push`	Array of objects	Array of push access levels, with each described by a hash of the form `{user_id: integer}`, `{group_id: integer}`, `{deploy_key_id: integer}`, or `{access_level: integer}`
`code_owner_approval_required`	Boolean	Whether merge requests into the branch need approval from code owners. Defaults to `true` when the branch is protected. Leave out on update to keep the current value
`merge_access_level`	Integer	Access level allowed to merge. Defaults to `40` (Maintainer role) Allowed values: `30`, `40`, `60`, `0`
`name` Required	String	Name of the branch or wildcard Example: `main`
`push_access_level`	Integer	Access level allowed to push. Defaults to `40` (Maintainer role) Allowed values: `30`, `40`, `60`, `0`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProtectedBranch
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—
`409`	Protected branch ‘main’ already exists	—
`422`	name is missing	—

Retrieve a protected branch or wildcard protected branch

`GET /api/v4/projects/{id}/protected_branches/{name}`

Retrieves a specified protected branch or wildcard protected branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`name` Path, required	String	Name of the branch or wildcard Example: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedBranch
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—

Update a protected branch

`PATCH /api/v4/projects/{id}/protected_branches/{name}`

Updates a protected branch for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`name` Path, required	String	Name of the branch or wildcard Example: `main`

Request body (`application/json`)

Property	Type	Description
`allow_force_push`	Boolean	If `true` , members who can push to this branch can also force push
`allowed_to_merge`	Array of objects	Array of merge access levels, with each described by a hash of the form `{user_id: integer}` , `{group_id: integer}` , or `{access_level: integer}`
`allowed_to_push`	Array of objects	Array of push access levels, with each described by a hash of the form `{user_id: integer}` , `{group_id: integer}` , `{deploy_key_id: integer}` , or `{access_level: integer}`
`code_owner_approval_required`	Boolean	Whether merge requests into the branch need approval from code owners. Defaults to `true` when the branch is protected. Leave out on update to keep the current value

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedBranch
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—
`422`	Push access levels access level has already been taken	—

Unprotect repository branches

`DELETE /api/v4/projects/{id}/protected_branches/{name}`

Unprotects a specified protected branch or wildcard protected branch.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`name` Path, required	String	Name of the branch or wildcard Example: `main`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesProtectedBranch

Property	Type	Description
`allow_force_push`	Boolean	—
`code_owner_approval_required`	Boolean	—
`id`	Integer (int64)	Example: `1`
`merge_access_levels`	Array of APIEntitiesProtectedRefAccess	—
`name`	String	Example: `main`
`push_access_levels`	Array of APIEntitiesProtectedRefAccess	—

APIEntitiesProtectedRefAccess

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`deploy_key_id`	Integer (int64)	Example: `1`
`group_id`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`user_id`	Integer	Example: `1`

Protected environments

Use this API to manage protected environments for a project, and protected environments for a group for a group.

> The project endpoints require CI/CD to be turned on for the project. If CI/CD is turned off, requests return `403 Forbidden`, even for users who otherwise have permission to manage protected environments.

List protected environments for a group

`GET /api/v4/groups/{id}/protected_environments`

Lists the protected environments configured for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProtected Environment
`400`	Bad Request	—
`404`	Not Found	—

Create a protected environment for a group

`POST /api/v4/groups/{id}/protected_environments`

Creates a protected environment and its nested configuration for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rules_attributes`	Array of objects	—
`approval_rules_attributes[]._destroy`	Boolean	—
`approval_rules_attributes[].access_level`	Integer	Example: `40`
`approval_rules_attributes[].group_id`	Integer	Example: `3`
`approval_rules_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`approval_rules_attributes[].id`	Integer	Example: `1`
`approval_rules_attributes[].required_approvals`	Integer	Example: `1`
`approval_rules_attributes[].user_id`	Integer	Example: `2`
`deploy_access_levels_attributes`	Array of objects	—
`deploy_access_levels_attributes[]._destroy`	Boolean	—
`deploy_access_levels_attributes[].access_level`	Integer	Example: `40`
`deploy_access_levels_attributes[].group_id`	Integer	Example: `3`
`deploy_access_levels_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`deploy_access_levels_attributes[].id`	Integer	Example: `1`
`deploy_access_levels_attributes[].user_id`	Integer	Example: `2`
	String	

Property	Type	Description
`name` Required		Allowed values: `production`, `staging`, `testing`, `development`, `other` Minimum length: `1`
`required_approval_count`	Integer	Example: `0`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a protected environment for a group

`GET /api/v4/groups/{id}/protected_environments/{name}`

Retrieves a protected environment configured for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`
`name` Path, required	String	Allowed values: `production`, `staging`, `testing`, `development`, `other` Minimum length: `1` Example: `production`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Update nested protected environment configuration for a group

`PUT /api/v4/groups/{id}/protected_environments/{name}`

Updates nested configuration without changing the protected environment name.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`
`name` Path, required	String	Allowed values: `production`, `staging`, `testing`, `development`, `other` Minimum length: `1` Example: `production`

Request body (`application/json`)

Property	Type	Description
`approval_rules_attributes`	Array of objects	—
`approval_rules_attributes[]._destroy`	Boolean	—
`approval_rules_attributes[].access_level`	Integer	Example: `40`
`approval_rules_attributes[].group_id`	Integer	Example: `3`
`approval_rules_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`approval_rules_attributes[].id`	Integer	Example: `1`
`approval_rules_attributes[].required_approvals`	Integer	Example: `1`
`approval_rules_attributes[].user_id`	Integer	Example: `2`
`deploy_access_levels_attributes`	Array of objects	—
`deploy_access_levels_attributes[]._destroy`	Boolean	—
`deploy_access_levels_attributes[].access_level`	Integer	Example: `40`
`deploy_access_levels_attributes[].group_id`	Integer	Example: `3`
`deploy_access_levels_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`deploy_access_levels_attributes[].id`	Integer	Example: `1`
`deploy_access_levels_attributes[].user_id`	Integer	Example: `2`
`required_approval_count`	Integer	Example: `0`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Delete a protected environment from a group

`DELETE /api/v4/groups/{id}/protected_environments/{name}`

Deletes a protected environment and its nested configuration from a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`
`name` Path, required	String	Allowed values: `production`, `staging`, `testing`, `development`, `other` Minimum length: `1` Example: `production`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List protected environments for a project

`GET /api/v4/projects/{id}/protected_environments`

Lists the protected environments configured for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProtected Environment
`400`	Bad Request	—
`404`	Not Found	—

Create a protected environment for a project

`POST /api/v4/projects/{id}/protected_environments`

Creates a protected environment and its nested configuration for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rules_attributes`	Array of objects	—
`approval_rules_attributes[]._destroy`	Boolean	—
`approval_rules_attributes[].access_level`	Integer	Example: `40`
`approval_rules_attributes[].group_id`	Integer	Example: `3`
`approval_rules_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`approval_rules_attributes[].id`	Integer	Example: `1`
`approval_rules_attributes[].required_approvals`	Integer	Example: `1`
`approval_rules_attributes[].user_id`	Integer	Example: `2`
`deploy_access_levels_attributes`	Array of objects	—
`deploy_access_levels_attributes[]._destroy`	Boolean	—
`deploy_access_levels_attributes[].access_level`	Integer	Example: `40`
`deploy_access_levels_attributes[].group_id`	Integer	Example: `3`
`deploy_access_levels_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`deploy_access_levels_attributes[].id`	Integer	Example: `1`
`deploy_access_levels_attributes[].user_id`	Integer	Example: `2`
	String	Example: `production`

Property	Type	Description
`name` Required		
`required_approval_count`	Integer	Example: `0`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

List inherited protected environments for a project

`GET /api/v4/projects/{id}/protected_environments/inherited`

Lists protected environments inherited from the project group hierarchy.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesInherited ProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a protected environment for a project

`GET /api/v4/projects/{id}/protected_environments/{name}`

Retrieves a protected environment configured for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`name` Path, required	String	Example: `production`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Update nested protected environment configuration for a project

`PUT /api/v4/projects/{id}/protected_environments/{name}`

Updates nested configuration without changing the protected environment name.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`name` Path, required	String	Example: `production`

Request body (`application/json`)

Property	Type	Description
`approval_rules_attributes`	Array of objects	—
`approval_rules_attributes[]._destroy`	Boolean	—
`approval_rules_attributes[].access_level`	Integer	Example: `40`
`approval_rules_attributes[].group_id`	Integer	Example: `3`
`approval_rules_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`approval_rules_attributes[].id`	Integer	Example: `1`
`approval_rules_attributes[].required_approvals`	Integer	Example: `1`
`approval_rules_attributes[].user_id`	Integer	Example: `2`
`deploy_access_levels_attributes`	Array of objects	—
`deploy_access_levels_attributes[]._destroy`	Boolean	—
`deploy_access_levels_attributes[].access_level`	Integer	Example: `40`
`deploy_access_levels_attributes[].group_id`	Integer	Example: `3`
`deploy_access_levels_attributes[].group_inheritance_type`	Integer	Allowed values: `0`, `1`
`deploy_access_levels_attributes[].id`	Integer	Example: `1`
`deploy_access_levels_attributes[].user_id`	Integer	Example: `2`
`required_approval_count`	Integer	Example: `0`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedEnvironment
`400`	Bad Request	—
`404`	Not Found	—

Delete a protected environment from a project

`DELETE /api/v4/projects/{id}/protected_environments/{name}`

Deletes a protected environment and its nested configuration from a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`name` Path, required	String	Example: `production`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesInheritedProtectedEnvironment

Property	Type	Description
`approval_rules`	Array of APIEntitiesProtectedEnvironmentApprovalRule	—
`deploy_access_levels`	Array of APIEntitiesProtectedEnvironmentAccessLevel	—
`group`	APIEntitiesProtectedEnvironmentOwnerGroup	—
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `production`
`required_approval_count`	Integer	Example: `0`

APIEntitiesProtectedEnvironment

Property	Type	Description
`approval_rules`	Array of APIEntitiesProtectedEnvironmentApprovalRule	—
`deploy_access_levels`	Array of APIEntitiesProtectedEnvironmentAccessLevel	—
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `production`
`required_approval_count`	Integer	Example: `0`

APIEntitiesProtectedEnvironmentAccessLevel

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`group_id`	Integer (int64)	Example: `3`
`group_inheritance_type`	Integer	Example: `0`
`id`	Integer (int64)	Example: `1`
`user_id`	Integer (int64)	Example: `2`

APIEntitiesProtectedEnvironmentApprovalRule

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`group_id`	Integer (int64)	Example: `3`
`group_inheritance_type`	Integer	Example: `0`
`id`	Integer (int64)	Example: `1`
`required_approvals`	Integer	Example: `1`
`user_id`	Integer (int64)	Example: `2`

APIEntitiesProtectedEnvironmentOwnerGroup

Property	Type	Description
`full_name`	String	Example: `Platform`
`full_path`	String	Example: `platform`
`id`	Integer (int64)	Example: `1`
`web_path`	String	Example: `/groups/platform`

Protected tags

Use this API to manage protected tags.

List all protected tags

`GET /api/v4/projects/{id}/protected_tags`

Lists all protected tags for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProtectedTag
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Protect a repository tag

`POST /api/v4/projects/{id}/protected_tags`

Protects a specified repository tag using a wildcard protected tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`allowed_to_create`	Array of objects	Array of deploy keys allowed to create protected tags
`create_access_level`	Integer	Access level allowed to create the protected tag. Defaults to `40` (Maintainer role) Allowed values: `30` , `40` , `60` , `0`
`name` Required	String	Name of the tag or wildcard Example: `release-1-0`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProtectedTag
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve a protected tag or wildcard protected tag

`GET /api/v4/projects/{id}/protected_tags/{name}`

Retrieves a specified protected tag or wildcard protected tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the tag or wildcard Example: `release*`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProtectedTag
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Unprotect repository tags

`DELETE /api/v4/projects/{id}/protected_tags/{name}`

Unprotects a specified protected tag or wildcard protected tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the tag or wildcard Example: `release-1-0`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—
`412`	Precondition Failed	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesProtectedRefAccess

Property	Type	Description
`access_level`	Integer	Example: `40`
`access_level_description`	String	Example: `Maintainers`
`deploy_key_id`	Integer (int64)	Example: `1`
`group_id`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`user_id`	Integer	Example: `1`

APIEntitiesProtectedTag

Property	Type	Description
`create_access_levels`	Array of APIEntitiesProtectedRefAccess	—
`name`	String	Example: `release-1-0`

Releases

Use this API to manage releases for a project, list the releases across a group, and manage the asset links attached to a release.

Asset links support the `http`, `https`, and `ftp` protocols.

List all releases in a group

`GET /api/v4/groups/{id}/releases`

Lists all releases for projects in a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group owned by the authenticated user
`sort` Query	String	The direction of the order. Either `desc` (default) for descending order or `asc` for ascending order Allowed values: `asc`, `desc` Default: `desc`
`simple` Query	Boolean	Return only limited fields for each release Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesRelease
`400`	Bad request	—
`403`	Forbidden	—
`404`	Not found	—

List all releases in a project

`GET /api/v4/projects/{id}/releases`

Lists all releases for a specified project. Sorted by `released_at`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	The field to use as order. Either `released_at` (default) or `created_at` Allowed values: `released_at`, `created_at` Default: `released_at`
`sort` Query	String	The direction of the order. Either `desc` (default) for descending order or `asc` for ascending order Allowed values: `asc`, `desc` Default: `desc`
`include_html_description` Query	Boolean	If `true`, a response includes HTML rendered markdown of the release description
`updated_before` Query	String (date-time)	Return releases updated before the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ
`updated_after` Query	String (date-time)	Return releases updated after the specified datetime. Format: ISO 8601 YYYY-MM-DDTHH:MM:SSZ

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesRelease
`400`	Bad Request	—
`404`	Not Found	—

Create a release

`POST /api/v4/projects/{id}/releases`

Creates a release. Developer level access to the project is required to create a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`assets`	Object	Object that contains assets for the release
`assets.links`	Array of objects	Link information about the release
`assets.links[].direct_asset_path`	String	Optional path for a direct asset link
`assets.links[].filepath`	String	Deprecated: optional path for a direct asset link
`assets.links[].link_type`	String	The type of the link Allowed values: `other`, `runbook`, `package`, `image` Default: `other`
`assets.links[].name` Required	String	The name of the link. Link names must be unique within the release
`assets.links[].url` Required	String	The URL of the link. Link URLs must be unique within the release
`description`	String	The description of the release. You can use Markdown
`legacy_catalog_publish`	Boolean	If true, the release will be published to the CI catalog. This parameter is for internal use only and will be removed in a future release. If the feature flag ci_release_cli_catalog_publish_option is disabled, this parameter will be ignored and the release will published to the CI catalog as it was before this parameter was introduced
`milestone_ids`	String or integer	The ID of each milestone the release is associated with. Git Lab Premium customers can specify group milestones.

Property	Type	Description
		Cannot be combined with `milestones` parameter. Mutually exclusive with `milestones`
`milestones`	Array of strings	The title of each milestone the release is associated with. Git Lab Premium customers can specify group milestones. Cannot be combined with `milestone_ids` parameter. Mutually exclusive with `milestone_ids`
`name`	String	The release name
`ref`	String	If a tag specified in `tag_name` doesn't exist, the release is created from `ref` and tagged with `tag_name`. It can be a commit SHA, another tag name, or a branch name
`released_at`	String (date-time)	Date and time for the release. Defaults to the current time. Expected in ISO 8601 format (`2019-03-15T08:00:00Z`). Only provide this field if creating an upcoming or historical release
`tag_message`	String	Message to use if creating a new annotated tag
`tag_name` Required	String	The tag where the release is created from

Responses

Code	Description	Schema
`201`	Created	APIEntitiesRelease
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`422`	Unprocessable entity	—

Get the latest project release

`GET /api/v4/projects/{id}/releases/permalink/latest`

This feature was introduced in GitLab 15.4.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Get the latest project release

`GET /api/v4/projects/{id}/releases/permalink/latest/{suffix_path}`

This feature was introduced in GitLab 15.4.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`suffix_path` Path, required	String	The path to be suffixed to the latest release

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a release by tag name

`GET /api/v4/projects/{id}/releases/{tag_name}`

Retrieves a release with a specified tag name.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`tag_name` Path, required	String	The Git tag the release is associated with
`include_html_description` Query	Boolean	If `true`, a response includes HTML rendered markdown of the release description

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRelease
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a release

`PUT /api/v4/projects/{id}/releases/{tag_name}`

Updates a release. Developer level access to the project is required to update a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`tag_name` Path, required	String	The Git tag the release is associated with

Request body (`application/json`)

Property	Type	Description
`description`	String	The description of the release. You can use Markdown
`milestone_ids`	String or integer	The ID of each milestone the release is associated with. Git Lab Premium customers can specify group milestones. Cannot be combined with `milestones` parameter. To remove all milestones from the release, specify `[]`. Mutually exclusive with `milestones`
`milestones`	Array of strings	The title of each milestone to associate with the release. Git Lab Premium customers can specify group milestones. Cannot be combined with `milestone_ids` parameter. To remove all milestones from the release, specify `[]`. Mutually exclusive with `milestone_ids`
`name`	String	The release name
`released_at`	String (date-time)	The date when the release is/ was ready. Expected in ISO 8601 format (`2019-03-15T08:00:00Z`)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRelease
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Delete a release

`DELETE /api/v4/projects/{id}/releases/{tag_name}`

Delete a release. Deleting a release doesn't delete the associated tag. Requires at least the Developer role for the project. This feature was introduced in GitLab 11.7.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`tag_name` Path, required	String	The Git tag the release is associated with

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRelease
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

List all release links

`GET /api/v4/projects/{id}/releases/{tag_name}/assets/links`

Lists all assets as links from a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Tag associated with the release
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesReleases Link
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a release link

`POST /api/v4/projects/{id}/releases/{tag_name}/assets/links`

Creates an asset link for a specified release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Tag associated with the release

Request body (`application/json`)

Property	Type	Description
`direct_asset_path`	String	Path for a direct asset link
`filepath`	String	Path for a direct asset link. Deprecated. Use `direct_asset_path` instead
`link_type`	String	Type of the link Allowed values: `other`, `runbook`, `image`, `package` Default: `other`
`name` Required	String	Name of the link. Link names must be unique in the release
`url` Required	String	URL of the link. Link URLs must be unique in the release

Responses

Code	Description	Schema
`201`	Created	APIEntitiesReleasesLink
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Retrieve a release link

`GET /api/v4/projects/{id}/releases/{tag_name}/assets/links/{link_id}`

Retrieves a specified asset as a link from a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Tag associated with the release
`link_id` Path, required	Integer	ID of the link

Responses

Code	Description	Schema
`200`	OK	APIEntitiesReleasesLink
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a release link

`PUT /api/v4/projects/{id}/releases/{tag_name}/assets/links/{link_id}`

Updates a specified asset link for a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Tag associated with the release
`link_id` Path, required	Integer	ID of the link

Request body (`application/json`)

Property	Type	Description
`direct_asset_path`	String	Path for a direct asset link
`filepath`	String	Path for a direct asset link. Deprecated. Use `direct_asset_path` instead
`link_type`	String	Type of the link Allowed values: `other`, `runbook`, `image`, `package` Default: `other`
`name`	String	Name of the link. At least one of `name`, `url` must be provided
`url`	String	URL of the link. At least one of `name`, `url` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesReleasesLink
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Delete a release link

`DELETE /api/v4/projects/{id}/releases/{tag_name}/assets/links/{link_id}`

Deletes a specified asset link from a release.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Tag associated with the release
`link_id` Path, required	Integer	ID of the link

Responses

Code	Description	Schema
`200`	OK	APIEntitiesReleasesLink
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not Found	—

Download a project release asset file

`GET /api/v4/projects/{id}/releases/{tag_name}/downloads/{direct_asset_path}`

This feature was introduced in GitLab 15.4.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`tag_name` Path, required	String	The Git tag the release is associated with
`direct_asset_path` Path, required	String	The path to the file to download, as specified when creating the release asset

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesMilestoneWithStats

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`issue_stats`	Object	—
`issue_stats.closed`	Integer	Example: `5`
`issue_stats.total`	Integer	Example: `10`
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesRelease

Property	Type	Description
`_links`	Object	—
`_links.closed_issues_url`	String	—
`_links.closed_merge_requests_url`	String	—
`_links.edit_url`	String	—
`_links.merged_merge_requests_url`	String	—
`_links.opened_issues_url`	String	—
`_links.opened_merge_requests_url`	String	—
`_links.self`	String	—
`assets`	Object	—
`assets.count`	Integer	Example: `2`
`assets.links`	APIEntitiesReleasesLink	—
`assets.sources`	APIEntitiesReleasesSource	—
`author`	APIEntitiesUserBasic	—
`commit`	APIEntitiesCommit	—
`commit_path`	String	Example: `/root/app/commit/588440f66559714280628a4f9799f0c4eb880a4a`
`created_at`	String (date-time)	Example: `2019-01-03T01:56:19.539Z`
`description`	String	Example: `Finally released v1.0`
`description_html`	String	—
`evidences`	APIEntitiesReleasesEvidence	—
`milestones`	APIEntitiesMilestoneWithStats	—
`name`	String	Example: `Release v1.0`

Property	Type	Description
`released_at`	String (date-time)	Example: `2019-01-03T01:56:19.539Z`
`tag_name`	String	Example: `v1.0`
`tag_path`	String	Example: `/root/app/-/tags/v1.0`
`upcoming_release`	Boolean	—

APIEntitiesReleasesEvidence

Property	Type	Description
`collected_at`	String (date-time)	Example: `2019-01-03T01:56:19.539Z`
`filepath`	String	Example: `https://gitlab.example.com/root/app/-/releases/v1.0/evidence.json`
`sha`	String	Example: `760d6cdfb0879c3ffedec13af470e0f71cf52c6cde4d`

APIEntitiesReleasesLink

Property	Type	Description
`direct_asset_url`	String	Example: `https://gitlab.example.com/root/app/-/releases/v1.0/downloads/app-v1.0.dmg`
`id`	Integer (int64)	Example: `1`
`link_type`	String	Example: `other`
`name`	String	Example: `app-v1.0.dmg`
`url`	String	Example: `https://gitlab.example.com/root/app/-/jobs/688/artifacts/raw/bin/app-v1.0.dmg`

APIEntitiesReleasesSource

Property	Type	Description
`format`	String	Example: `zip`
`url`	String	Example: `https://gitlab.example.com/root/app/-/archive/v1.0/app-v1.0.zip`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Remote mirrors

Use this API to manage remote mirrors. You can query and modify the state of these mirrors.

For security reasons, the `url` attribute in the API response is always scrubbed of username and password information.

> Pull mirrors use a different API endpoint to display and update them.

List all remote mirrors for a project

`GET /api/v4/projects/{id}/remote_mirrors`

Lists all remote mirrors for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesRemote Mirror
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Create a push mirror

`POST /api/v4/projects/{id}/remote_mirrors`

Creates a push mirror.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`auth_method`	String	Mirror authentication method Allowed values: `ssh_public_key` , `password`
`enabled`	Boolean	If `true` , the mirror is enabled
`host_keys`	Array of strings	SSH host keys in bare format, such as `ssh-ed25519 AAAA...` , or full `known_hosts` format, such as `hostname ssh-ed25519 AAAA...` . Bare keys use the hostname from the mirror URL
`keep_divergent_refs`	Boolean	If `true` , divergent refs are kept when mirroring
`only_protected_branches`	Boolean	If `true` , only protected branches are mirrored
`url` Required	String	Target URL to which the repository is mirrored Example: `https://*****:*****@example.com/gitlab/example.git`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesRemoteMirror
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a remote mirror for a project

`GET /api/v4/projects/{id}/remote_mirrors/{mirror_id}`

Retrieves a specified remote mirror for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`mirror_id` Path, required	String	ID of the remote mirror

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRemoteMirror
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Update a remote mirror in a project

`PUT /api/v4/projects/{id}/remote_mirrors/{mirror_id}`

Updates the configuration or operational status of a specified remote mirror.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`mirror_id` Path, required	String	ID of the remote mirror

Request body (`application/json`)

Property	Type	Description
`auth_method`	String	Mirror authentication method
`enabled`	Boolean	If `true`, the mirror is enabled Example: `true`
`host_keys`	Array of strings	SSH host keys in bare format, such as `ssh-ed25519 AAAA...`, or full `known_hosts` format, such as `hostname ssh-ed25519 AAAA...`. Bare keys use the hostname from the mirror URL
`keep_divergent_refs`	Boolean	If `true`, divergent refs are kept when mirroring
`only_protected_branches`	Boolean	If `true`, only protected branches are mirrored

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRemoteMirror
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Delete a remote mirror from a project

`DELETE /api/v4/projects/{id}/remote_mirrors/{mirror_id}`

Deletes a specified remote mirror from a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`mirror_id` Path, required	String	ID of the remote mirror

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Retrieve a public key for a remote mirror

`GET /api/v4/projects/{id}/remote_mirrors/{mirror_id}/public_key`

Retrieves the public key of a specified remote mirror that uses SSH authentication.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`mirror_id` Path, required	String	ID of the remote mirror

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Force push mirror update

`POST /api/v4/projects/{id}/remote_mirrors/{mirror_id}/sync`

Forces an update to a push mirror.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`mirror_id` Path, required	String	ID of the remote mirror

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`401`	Unauthorized	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesMirrorHostKey

Property	Type	Description
`fingerprint_sha256`	String	Example: `SHA256:abcd1234`

APIEntitiesRemoteMirror

Property	Type	Description
`auth_method`	String	Example: `password`
`enabled`	Boolean	Example: `true`
`host_keys`	Array of APIEntitiesMirrorHostKey	—
`id`	Integer (int64)	Example: `101486`
`keep_divergent_refs`	Boolean	—
`last_error`	String	Example: `The remote mirror URL is invalid.`
`last_successful_update_at`	String (date-time)	Example: `2020-01-06T17:31:55.864Z`
`last_update_at`	String (date-time)	Example: `2020-01-06T17:32:02.823Z`
`last_update_started_at`	String (date-time)	Example: `2020-01-06T17:32:02.823Z`
`only_protected_branches`	Boolean	—
`update_status`	String	Example: `finished`
`url`	String	Example: `https://*****:*****@example.com/gitlab/example.git`

Repositories

Use this API to manage Git repositories.

Retrieve file archive from a repository

`GET /api/v4/projects/{id}/repository/archive`

Retrieves the file archive of a specified repository. This endpoint can be accessed without authentication if the repository is publicly accessible. For GitLab.com users, this endpoint has a rate limit threshold of 5 requests per minute.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`sha` Query	String	Commit SHA to download. Accepts a tag, branch reference, or SHA. If omitted, defaults to the tip of the default branch Example: `7d70e02340bac451f281cecf0a980907974bd8be`
`ref_type` Query	String	Type of ref in `sha`. Use to select the correct ref when a branch and tag share a name Allowed values: `heads`, `tags`
`format` Query	String	Archive format Example: `tar.gz`
`path` Query	String	Subpath of the repository to download. An empty string returns the whole repository Example: `files/archives`
`include_lfs_blobs` Query	Boolean	If `true`, LFS objects are included in the archive. When set to `false`, LFS objects are excluded Default: `true`
`exclude_paths` Query	Array of strings	Comma-separated list of paths to exclude from the archive Default: `[]`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Get contents of multiple files in a single request

`POST /api/v4/projects/{id}/repository/blobs/batch`

Each blob is truncated to the first 1 MB; the `truncated` field indicates when this happens.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`

Request body (`application/json`)

Property	Type	Description
`files` Required	Array of objects	Array of file objects to retrieve, up to a maximum of 20
`files[].path` Required	String	Path of the file in the repository Example: `app/models/user.rb`
`files[].ref`	String	Branch, tag, or commit to read the file from. Defaults to the default branch Example: `main`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesBatchBlob
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a blob from a repository

`GET /api/v4/projects/{id}/repository/blobs/{sha}`

Retrieves information, such as size and content, about blobs in a repository. Blob content is Base64 encoded. This endpoint can be accessed without authentication, if the repository is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`sha` Path, required	String	SHA of the blob Example: `7d70e02340bac451f281cecf0a980907974bd8be`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve raw blob content

`GET /api/v4/projects/{id}/repository/blobs/{sha}/raw`

Retrieves the raw file contents for a blob, by blob SHA. This endpoint can be accessed without authentication if the repository is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`sha` Path, required	String	SHA of the blob Example: `7d70e02340bac451f281cecf0a980907974bd8be`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

List files changed between two commits

`GET /api/v4/projects/{id}/repository/changed_paths`

Returns the path, change status, and file modes for every path that differs between two refs.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`from` Query, required	String	The commit SHA or branch/tag to compare from Maximum length: `255` Example: `main`
`to` Query, required	String	The commit SHA or branch/tag to compare to Maximum length: `255` Example: `feature`
`find_renames` Query	Boolean	If true, detect file renames Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesChanged Path
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—
`429`	Too Many Requests	—

Generate changelog data

`GET /api/v4/projects/{id}/repository/changelog`

Generates changelog data based on commits in a repository, without committing them to a changelog file. Works exactly like `POST /projects/:id/repository/changelog`, except the changelog data is not committed to any changelog file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`version` Query, required	String	Version to generate the changelog for. Must follow semantic versioning format Pattern: `(?:(?:(0|[1-9]\d*))\.(?:(0|[1-9]\d*))\.(?:(0|[1-9]\d*)))(?:(?:-((?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0)(?:\.(?:\d*[a-zA-Z\-][0-9a-zA-Z\-]*|[1-9]\d*|0))*))?(?:\+([0-9a-zA-Z\-]+(?:\.[0-9a-zA-Z\-]+)*))?)` Example: `1.0.0`
`from` Query	String	First commit in the range of commits, as a SHA, to use for the changelog. This commit is not included in the changelog. If omitted, the most recent stable version tag before `version` is used Example: `ed899a2f4b50b4370feeea94676502b42383c746`
`to` Query	String	Last commit in the range of commits, as a SHA, to use for the changelog. This commit is included in the changelog. If omitted, uses the latest commit on the default branch. For the Add changelog data to file operation, uses the latest commit on `branch` instead. The range can contain at most 15,000 commits Example:

Name	Type	Description
		`6104942438c14ec7bd21c6cd5bd995272b3faff6`
`date` Query	String (date-time)	Date and time of the release. Defaults to the current time Example: `2021-09-20T11:50:22.001+00:00`
`trailer` Query	String	Git trailer to use for including commits. Case-sensitive: `Example` does not match `example` or `eXaMpLE` Default: `Changelog` Example: `Changelog`
`config_file` Query	String	Path to the changelog configuration file in the project's Git repository. If omitted, uses `.gitlab/changelog_config.yml` Example: `.gitlab/changelog_config.yml`
`config_file_ref` Query	String	Git reference, for example a branch, where the changelog configuration file is defined. Defaults to the default branch of the repository Example: `main`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesChangelog
`400`	Bad Request	—
`404`	Not Found	—

Add changelog data to file

`POST /api/v4/projects/{id}/repository/changelog`

Adds changelog data to file.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`

Request body (`application/json`)

Property	Type	Description
`branch`	String	Name of the branch to commit the changelog changes to. If omitted, defaults to the project's default branch Example: `main`
`config_file`	String	Path to the changelog configuration file in the project's Git repository. If omitted, uses `.gitlab/changelog_config.yml` Example: `.gitlab/changelog_config.yml`
`config_file_ref`	String	Git reference, for example a branch, where the changelog configuration file is defined. Defaults to the default branch of the repository Example: `main`
`date`	String (date-time)	Date and time of the release. Defaults to the current time Example: `2021-09-20T11:50:22.001+00:00`
`file`	String	File to commit the changes to Default: `CHANGELOG.md` Example: `CHANGELOG.md`
`from`	String	First commit in the range of commits, as a SHA, to use for the changelog. This commit is not included in the changelog. If omitted, the most recent stable version tag before `version` is used Example: `ed899a2f4b50b4370feeea94676502b42383c746`
`message`	String	

Property	Type	Description
		Commit message to use when committing the changelog changes. If omitted, defaults to `Add changelog for version X`, where `X` is the value of the `version` parameter Example: `Initial commit`
`to`	String	Last commit in the range of commits, as a SHA, to use for the changelog. This commit is included in the changelog. If omitted, uses the latest commit on the default branch. For the Add changelog data to file operation, uses the latest commit on `branch` instead. The range can contain at most 15,000 commits Example: `6104942438c14ec7bd21c6cd5bd995272b3faff6`
`trailer`	String	Git trailer to use for including commits. Case-sensitive: `Example` does not match `example` or `eXaMpLE` Default: `Changelog` Example: `Changelog`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Compare branches, tags, or commits

`GET /api/v4/projects/{id}/repository/compare`

Compares branches, tags, or commits. Retrieves the differences between two branches, tags, or commits in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`from` Query, required	String	Ref to compare from. Accepts a commit SHA, branch name, or tag name Example: `main`
`to` Query, required	String	Ref to compare to. Accepts a commit SHA, branch name, or tag name Example: `feature`
`from_project_id` Query	Integer	ID of the project to compare from Example: `1`
`straight` Query	Boolean	If `true`, comparison method is direct comparison between `from` and `to` (`from`..`to`). If `false`, compares using merge base (`from`...`to`) Default: `false`
`unidiff` Query	Boolean	If `true`, presents diffs in the unified diff format Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCompare
`400`	Bad Request	—
`404`	Not Found	—

Retrieve contributors metrics

`GET /api/v4/projects/{id}/repository/contributors`

Retrieves a list of contributors to a specified repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`ref` Query	String	Name of a repository branch or tag. If omitted, uses the default branch Example: `main`
`order_by` Query	String	Sort results by the specified field Allowed values: `email`, `name`, `commits` Default: `commits`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `asc`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesContributor
`400`	Bad Request	—
`404`	Not Found	—

Get diff statistics between two commits

`GET /api/v4/projects/{id}/repository/diff_stats`

Returns the number of added and deleted lines for every file that differs between two refs.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`from` Query, required	String	The commit SHA or branch/tag to compare from Maximum length: `255` Example: `main`
`to` Query, required	String	The commit SHA or branch/tag to compare to Maximum length: `255` Example: `feature`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesDiffStat
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—

Retrieve diverging commit counts between two refs

`GET /api/v4/projects/{id}/repository/diverging_commits`

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`from` Query, required	String	Ref to compare from. Accepts a commit SHA, branch name, or tag name Maximum length: `255` Example: `main`
`to` Query, required	String	Ref to compare to. Accepts a commit SHA, branch name, or tag name Maximum length: `255` Example: `feature`
`max_count` Query	Integer	Maximum number of commits to count. Use `0` for unlimited Default: `0` Minimum: `0` Example: `1000`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesDivergingCommitCount
`400`	Bad request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—
`429`	Too many requests	—

Retrieve repository health statistics

`GET /api/v4/projects/{id}/repository/health`

Retrieves statistics related to the health of a project repository. This endpoint is rate-limited to 5 requests/hour per project when `generate` is `true`. Available only to users with push access to the repository.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`generate` Query	Boolean	If `true`, generates a new health report. Set this if the endpoint returns `404` Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesRepositoryHealth
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge base

`GET /api/v4/projects/{id}/repository/merge_base`

Retrieves the merge base for two specified commits.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`refs` Query, required	Array of strings	Refs to find the common ancestor of. Accepts multiple refs Example: `["main","feature"]`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommit
`400`	Bad Request	—
`404`	Not Found	—

List all repository trees in a project

`GET /api/v4/projects/{id}/repository/tree`

Lists all repository files and directories in a specified project. This endpoint can be accessed without authentication if the repository is publicly accessible. This command provides essentially the same features as the `git ls-tree` command. Use `with_last_commit` to include the last commit that changed each entry. `with_last_commit` cannot be combined with `recursive`.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project Example: `1`
`ref` Query	String	Name of a repository branch or tag. If omitted, uses the default branch Maximum length: `1024` Example: `main`
`path` Query	String	Path inside the repository, used to get the contents of subdirectories Maximum length: `1024` Example: `files/html`
`recursive` Query	Boolean	If `true`, returns a recursive tree Default: `false`
`with_last_commit` Query	Boolean	If `true`, includes the last commit for each tree entry. Cannot be combined with `recursive`. Introduced in Git Lab 19.3 Default: `false`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`pagination` Query	String	Pagination method. `keyset` uses the keyset-based pagination method. `none` is valid only if `recursive` is `true` Allowed values: `legacy`,

Name	Type	Description
		`keyset` , `none` Default: `legacy`
`page_token` Query	String	Tree record ID at which to fetch the next page. Used only with keyset pagination Example: `a1e8f8d745cc87e3a9248358d9352bb7f9a0aeba`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTreeObject
`400`	Bad Request	—
`404`	404 Project Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBatchBlob

Property	Type	Description
`content`	String	Example: `VGhpcyBpcyBhIGZpbGU=`
`encoding`	String	Example: `base64`
`path`	String	Example: `app/models/user.rb`
`ref`	String	Example: `main`
`size`	Integer	Example: `1476`
`truncated`	Boolean	Example: `false`

APIEntitiesChangedPath

Property	Type	Description
`new_mode`	String	Example: `100644`
`old_mode`	String	Example: `100644`
`old_path`	String	Example: `app/models/user.rb`
`path`	String	Example: `app/models/user.rb`
`status`	String	Example: `MODIFIED`

APIEntitiesChangelog

Property	Type	Description
`notes`	String	Example: `## 1.0.0 (2023-01-01)`

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCompare

Property	Type	Description
`commit`	APIEntitiesCommit	—
`commits`	Array of APIEntitiesCommit	—
`compare_same_ref`	Boolean	—
`compare_timeout`	Boolean	—
`diffs`	Array of APIEntitiesDiff	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab-foss/-/compare/main...feature`

APIEntitiesContributor

Property	Type	Description
`additions`	Integer	Example: `3`
`commits`	Integer	Example: `117`
`deletions`	Integer	Example: `5`
`email`	String	Example: `johndoe@example.com`
`name`	String	Example: `John Doe`

APIEntitiesDiff

Property	Type	Description
`a_mode`	String	Example: `100755`
`b_mode`	String	Example: `100644`
`collapsed`	Boolean	—
`deleted_file`	Boolean	—
`diff`	String	Example: `@@ -71,6 +71,8 @@\n...`
`generated_file`	Boolean	—
`new_file`	Boolean	—
`new_path`	String	Example: `doc/update/5.4-to-6.0.md`
`old_path`	String	Example: `doc/update/5.4-to-6.0.md`
`renamed_file`	Boolean	—
`too_large`	Boolean	—

APIEntitiesDiffStat

Property	Type	Description
`additions`	Integer	Example: `10`
`deletions`	Integer	Example: `3`
`old_path`	String	Example: `app/models/user.rb`
`path`	String	Example: `app/models/user.rb`

APIEntitiesDivergingCommitCount

Property	Type	Description
`ahead`	Integer	Example: `5`
`behind`	Integer	Example: `3`

APIEntitiesRepositoryHealth

Property	Type	Description
`alternates`	Object	—
`bitmap`	APIEntitiesRepositoryHealthBitmap	—
`commit_graph`	APIEntitiesRepositoryHealthCommitGraph	—
`is_object_pool`	Boolean	—
`last_full_repack`	APIEntitiesRepositoryHealthLastFullRepack	—
`multi_pack_index`	APIEntitiesRepositoryHealthMultiPackIndex	—
`multi_pack_index_bitmap`	APIEntitiesRepositoryHealthBitmap	—
`objects`	APIEntitiesRepositoryHealthObjects	—
`references`	APIEntitiesRepositoryHealthReferences	—
`size`	Integer	—
`updated_at`	String (date-time)	Example: `2025-02-24T09:05:50.355Z`

APIEntitiesRepositoryHealthBitmap

Property	Type	Description
`has_hash_cache`	Boolean	—
`has_lookup_table`	Boolean	—
`version`	Integer	—

APIEntitiesRepositoryHealthCommitGraph

Property	Type	Description
`commit_graph_chain_length`	Integer	—
`has_bloom_filters`	Boolean	—
`has_generation_data`	Boolean	—
`has_generation_data_overflow`	Boolean	—

APIEntitiesRepositoryHealthLastFullRepack

Property	Type	Description
`nanos`	Integer	—
`seconds`	Integer	—

APIEntitiesRepositoryHealthMultiPackIndex

Property	Type	Description
`packfile_count`	Integer	—
`version`	Integer	—

APIEntitiesRepositoryHealthObjects

Property	Type	Description
`cruft_count`	Integer	—
`keep_count`	Integer	—
`keep_size`	Integer	—
`loose_objects_count`	Integer	—
`loose_objects_garbage_count`	Integer	—
`packfile_count`	Integer	—
`recent_size`	Integer	—
`reverse_index_count`	Integer	—
`size`	Integer	—
`stale_loose_objects_count`	Integer	—
`stale_size`	Integer	—

APIEntitiesRepositoryHealthReferences

Property	Type	Description
`loose_count`	Integer	—
`packed_size`	Integer	—
`reference_backend`	String	—

APIEntitiesTreeObject

Property	Type	Description
`id`	String	Example: `a1e8f8d745cc87e3a9248358d9352bb7f9a0aeba`
`last_commit`	APIEntitiesCommit	—
`mode`	String	Example: `040000`
`name`	String	Example: `html`
`path`	String	Example: `files/html`
`type`	String	Example: `tree`

Resource events

Use this API to retrieve the events that track changes to issues, merge requests, and epics:

- Label events record who added or removed a label, and when.
- Iteration events record iteration changes for issues.
- Milestone events record milestone changes for issues and merge requests.
- State events record state changes for issues, merge requests, and epics.
- Weight events record weight changes for issues.

State events do not track the initial state (`created` or `opened`) of a resource. For a resource that was never closed or reopened, an empty list is returned.

List all project issue label events

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_label_events`

Lists all label events for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer or string	The IID of the issue
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceLabelEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an issue label event

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_label_events/{event_id}`

Retrieves a label event for a specified project issue.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`event_id` Path, required	String	The ID of a resource label event
`eventable_id` Path, required	Integer or string	The IID of the issue

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceLabelEvent
`400`	Bad Request	—
`404`	Not Found	—

List all project issue milestone events

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_milestone_events`

Lists all milestone events for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`eventable_id` Path, required	Integer or string	The ID of the eventable
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesResourceMilestoneEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an issue milestone event

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_milestone_events/{event_id}`

Retrieves a specified milestone event for a project issue.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`event_id` Path, required	String	The ID of a resource milestone event
`eventable_id` Path, required	Integer or string	The ID of the eventable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceMilestoneEvent
`400`	Bad Request	—
`404`	Not found	—

List all project issue state events

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_state_events`

Lists all state events for a specified issue.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer	The IID of the issue
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceStateEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve an issue state event

`GET /api/v4/projects/{id}/issues/{eventable_id}/resource_state_events/{event_id}`

Retrieves a state event for a specified project issue.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer	The IID of the issue
`event_id` Path, required	Integer	The ID of a resource state event

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceStateEvent
`400`	Bad Request	—
`404`	Not Found	—

List all project merge request label events

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_label_events`

Lists all label events for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer or string	The IID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceLabelEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge request label event

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_label_events/{event_id}`

Retrieves a label event for a specified project merge request.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`event_id` Path, required	String	The ID of a resource label event
`eventable_id` Path, required	Integer or string	The IID of the merge request

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceLabelEvent
`400`	Bad Request	—
`404`	Not Found	—

List all project merge request milestone events

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_milestone_events`

Lists all milestone events for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`eventable_id` Path, required	Integer or string	The ID of the eventable
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesResource MilestoneEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge request milestone event

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_milestone_events/{event_id}`

Retrieves a specified milestone event for a project merge request.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`event_id` Path, required	String	The ID of a resource milestone event
`eventable_id` Path, required	Integer or string	The ID of the eventable

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceMilestoneEvent
`400`	Bad Request	—
`404`	Not found	—

List all project merge request state events

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_state_events`

Lists all state events for a specified merge request.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer	The IID of the merge request
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceStateEvent
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a merge request state event

`GET /api/v4/projects/{id}/merge_requests/{eventable_id}/resource_state_events/{event_id}`

Retrieves a state event for a specified project merge request.

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project
`eventable_id` Path, required	Integer	The IID of the merge request
`event_id` Path, required	Integer	The ID of a resource state event

Responses

Code	Description	Schema
`200`	OK	APIEntitiesResourceStateEvent
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesLabelBasic

Property	Type	Description
`archived`	Boolean	Example: `false`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `bug`
`text_color`	String	Example: `#FFFFFF`

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesResourceLabelEvent

Property	Type	Description
`action`	String	Example: `add`
`created_at`	String (date-time)	Example: `2018-08-20T13:38:20.077Z`
`id`	Integer (int64)	Example: `142`
`label`	APIEntitiesLabelBasic	—
`resource_id`	Integer (int64)	Example: `253`
`resource_type`	String	Example: `Issue`
`user`	APIEntitiesUserBasic	—

APIEntitiesResourceMilestoneEvent

Property	Type	Description
`action`	String	Example: `add`
`created_at`	String (date-time)	Example: `2018-08-20T13:38:20.077Z`
`id`	Integer (int64)	Example: `142`
`milestone`	APIEntitiesMilestone	—
`resource_id`	Integer (int64)	Example: `253`
`resource_type`	String	Example: `Issue`
`state`	String	Example: `active`
`user`	APIEntitiesUserBasic	—

APIEntitiesResourceStateEvent

Property	Type	Description
`created_at`	String (date-time)	Example: `2018-08-20T13:38:20.077Z`
`id`	Integer (int64)	Example: `142`
`resource_id`	Integer (int64)	Example: `253`
`resource_type`	String	Example: `Issue`
`source_commit`	String	Example: `7b09ce7e6f80347baf0316c8c94cdba9a0a7e91d`
`source_merge_request_id`	Integer (int64)	Example: `45`
`state`	String	Example: `opened`
`user`	APIEntitiesUserBasic	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Resource subscriptions

Use this API to subscribe to and unsubscribe from issues, merge requests, and labels in projects and groups.

All API calls to non-public information require authentication.

Subscribe to a group label

`POST /api/v4/groups/{id}/labels/{subscribable_id}/subscribe`

Subscribes the currently authenticated user to a specified group label. They will receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Unsubscribe to a group label

`POST /api/v4/groups/{id}/labels/{subscribable_id}/unsubscribe`

Unsubscribes the currently authenticated user from a specified group label. They will no longer receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The group ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGroupLabel
`400`	Bad Request	—
`404`	Not Found	—

Subscribe to an issue

`POST /api/v4/projects/{id}/issues/{subscribable_id}/subscribe`

Subscribes the currently authenticated user to a specified issue. They will receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Unsubscribe to an issue

`POST /api/v4/projects/{id}/issues/{subscribable_id}/unsubscribe`

Unsubscribes the currently authenticated user from a specified issue. They will no longer receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesIssue
`400`	Bad Request	—
`404`	Not Found	—

Subscribe to a project label

`POST /api/v4/projects/{id}/labels/{subscribable_id}/subscribe`

Subscribes the currently authenticated user to a specified project label. They will receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Unsubscribe to a project label

`POST /api/v4/projects/{id}/labels/{subscribable_id}/unsubscribe`

Unsubscribes the currently authenticated user from a specified project label. They will no longer receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectLabel
`400`	Bad Request	—
`404`	Not Found	—

Subscribe to a merge request

`POST /api/v4/projects/{id}/merge_requests/{subscribable_id}/subscribe`

Subscribes the currently authenticated user to a specified merge request. They will receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequest
`400`	Bad Request	—
`404`	Not Found	—

Unsubscribe to a merge request

`POST /api/v4/projects/{id}/merge_requests/{subscribable_id}/unsubscribe`

Unsubscribes the currently authenticated user from a specified merge request. They will no longer receive notifications on changes to this item.

Parameters

Name	Type	Description
`id` Path, required	String	The project ID
`subscribable_id` Path, required	String	The ID of a resource

Responses

Code	Description	Schema
`201`	Created	APIEntitiesMergeRequest
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiPipeline

Property	Type	Description
`archived`	Boolean	Example: `false`
`before_sha`	String	Example: `a91957a858320c0e17f3a0eca7cfacbff50ea29a`
`committed_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`detailed_status`	DetailedStatusEntity	—
`duration`	Integer	Time spent running in seconds Example: `127`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`queued_duration`	Integer	Time spent enqueued in seconds Example: `63`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `success`
`tag`	Boolean	Example: `false`

Property	Type	Description
`updated_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`user`	APIEntitiesUserBasic	—
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`
`yaml_errors`	String	Example: `widgets:build: needs 'widgets:test'`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesDiffRefs

Property	Type	Description
`base_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`head_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`
`start_sha`	String	Example: `b83d6e391c22777fca1ed3012fce84f633d7fed0`

APIEntitiesGroupLabel

Property	Type	Description
`archived`	Boolean	Example: `false`
`closed_issues_count`	Integer	Example: `0`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `bug`
`open_issues_count`	Integer	Example: `1`
`open_merge_requests_count`	Integer	Example: `1`
`subscribed`	Boolean	Example: `false`
`text_color`	String	Example: `#FFFFFF`

APIEntitiesIssuableReferences

Property	Type	Description
`full`	String	Example: `test&6`
`relative`	String	Example: `&6`
`short`	String	Example: `&6`

APIEntitiesIssuableTimeStats

Property	Type	Description
`human_time_estimate`	String	Example: `3h 30m`
`human_total_time_spent`	String	Example: `1h`
`time_estimate`	Integer	Example: `12600`
`total_time_spent`	Integer	Example: `3600`

APIEntitiesIssue

Property	Type	Description
`_links`	Object	—
`_links.award_emoji`	String	Example: `http://example.com/api/v4/projects/1/issues/2/award_emoji`
`_links.closed_as_duplicate_of`	String	Example: `http://example.com/api/v4/projects/1/issues/75`
`_links.notes`	String	Example: `http://example.com/api/v4/projects/1/issues/2/notes`
`_links.project`	String	Example: `http://example.com/api/v4/projects/1`
`_links.self`	String	Example: `http://example.com/api/v4/projects/1/issues/2`
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesUserBasic	—
`closed_at`	String (date-time)	Example: `2022-11-15T08:30:55.232Z`
`closed_by`	APIEntitiesUserBasic	—
`confidential`	Boolean	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`discussion_locked`	Boolean	—
`downvotes`	Integer	—
`due_date`	String (date)	Example: `2022-11-20`

Property	Type	Description
`has_tasks`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `github`
`issue_type`	String	Example: `issue`
`labels`	Array of strings	Example: `["bug"]`
`merge_requests_count`	Integer	—
`milestone`	APIEntitiesMilestone	—
`moved_to_id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `4`
`references`	APIEntitiesIssuableReferences	—
`service_desk_reply_to`	String	Example: `user@example.com`
`severity`	String	The severity of the issue Allowed values: `UNKNOWN`, `LOW`, `MEDIUM`, `HIGH`, `CRITICAL`
`start_date`	String (date)	Example: `2022-11-18`
`state`	String	Example: `closed`
`subscribed`	Boolean	Example: `false`
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`task_status`	String	Example: `2 of 4 tasks completed`
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`

Property	Type	Description
`type`	String	The type of the issue Allowed values: `ISSUE`, `INCIDENT`, `TEST_CASE`, `REQUIREMENT`, `TASK`, `TICKET` Example: `ISSUE`
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `http://example.com/example/example/issues/14`

APIEntitiesMergeRequest

Property	Type	Description
`allow_collaboration`	Boolean	—
`allow_maintainer_to_push`	Boolean	—
`assignee`	APIEntitiesUserBasic	—
`assignees`	APIEntitiesUserBasic	—
`author`	APIEntitiesMergeRequestAuthor	—
`blocking_discussions_resolved`	Boolean	—
`changes_count`	String	Example: `1`
`closed_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`closed_by`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2022-08-17T12:46:35.053Z`
`description`	String	Example: `Repellendus impedit et vel velit dignissimos.`
`description_html`	String	—
`detailed_merge_status`	String	Example: `mergeable`
`diff_refs`	APIEntitiesDiffRefs	—
`discussion_locked`	Boolean	—
`diverged_commits_count`	Integer	Example: `0`
`downvotes`	Integer	—
`draft`	Boolean	—
`first_contribution`	Boolean	—
`first_deployed_to_production_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`force_remove_source_branch`	Boolean	—

Property	Type	Description
`has_conflicts`	Boolean	—
`head_pipeline`	APIEntitiesCiPipeline	—
`id`	Integer (int64)	Example: `84`
`iid`	Integer	Example: `14`
`imported`	Boolean	—
`imported_from`	String	Example: `bitbucket`
`labels`	Array of strings	—
`latest_build_finished_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`latest_build_started_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_after`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merge_commit_sha`	String	Example: `1234abcd`
`merge_error`	String	Example: `Merge error`
`merge_status`	String	Example: `unchecked`
`merge_user`	APIEntitiesUserBasic	—
`merge_when_pipeline_succeeds`	Boolean	—
`merged_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`merged_by`	APIEntitiesUserBasic	—
`milestone`	APIEntitiesMilestone	—
`pipeline`	APIEntitiesCiPipelineBasic	—
`prepared_at`	String (date-time)	Example: `2022-01-31T15:10:45.080Z`
`project_id`	Integer (int64)	Example: `4`

Property	Type	Description
`rebase_in_progress`	Boolean	—
`reference`	String	Example: `!1`
`references`	APIEntitiesIssuableReferences	—
`reviewers`	APIEntitiesUserBasic	—
`sha`	String	Example: `1234abcd`
`should_remove_source_branch`	Boolean	—
`source_branch`	String	—
`source_project_id`	Integer (int64)	—
`squash`	Boolean	—
`squash_commit_sha`	String	Example: `1234abcd`
`squash_on_merge`	Boolean	—
`state`	String	Example: `closed`
`subscribed`	Boolean	—
`target_branch`	String	—
`target_project_id`	Integer (int64)	—
`task_completion_status`	APIEntitiesTaskCompletionStatus	—
`time_stats`	APIEntitiesIssuableTimeStats	—
`title`	String	Example: `Impedit et ut et dolores vero provident ullam est`
`title_html`	String	—
`updated_at`	String (date-time)	Example: `2022-11-14T17:22:01.470Z`
`upvotes`	Integer	—
`user`	Object	—

Property	Type	Description
`user.can_merge`	Boolean	—
`user_notes_count`	Integer	—
`web_url`	String	Example: `https://gitlab.example.com/my-group/my-project/-/merge_requests/1`
`work_in_progress`	Boolean	—

APIEntitiesMergeRequestAuthor

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bot`	Boolean	Example: `false`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesMilestone

Property	Type	Description
`created_at`	String	—
`description`	String	—
`due_date`	String	—
`expired`	Boolean	—
`group_id`	String	—
`id`	Integer (int64)	—
`iid`	Integer (int64)	—
`project_id`	Integer (int64)	—
`start_date`	String	—
`state`	String	—
`title`	String	—
`updated_at`	String	—
`web_url`	String	—

APIEntitiesProjectLabel

Property	Type	Description
`archived`	Boolean	Example: `false`
`closed_issues_count`	Integer	Example: `0`
`color`	String	Example: `#FF0000`
`description`	String	Example: `Bug reported by user`
`description_html`	String	Example: `<p>Bug reported by user</p>`
`id`	Integer (int64)	Example: `1`
`is_project_label`	Boolean	—
`name`	String	Example: `bug`
`open_issues_count`	Integer	Example: `1`
`open_merge_requests_count`	Integer	Example: `1`
`priority`	Integer	Example: `10`
`subscribed`	Boolean	Example: `false`
`text_color`	String	Example: `#FFFFFF`

APIEntitiesTaskCompletionStatus

Property	Type	Description
`completed_count`	Integer	Example: `3`
`count`	Integer	Example: `5`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

DetailedStatusEntity

Property	Type	Description
`action`	Object	—
`action.button_title`	String	Example: `Cancel this job`
`action.confirmation_message`	String	Example: `Are you sure?`
`action.icon`	String	Example: `cancel`
`action.method`	String	Example: `post`
`action.path`	String	Example: `/namespace1/project1/-/jobs/2/cancel`
`action.title`	String	Example: `Cancel`
`details_path`	String	Example: `/test-group/test-project/-/pipelines/287`
`favicon`	String	Example: `/assets/ci_favicons/favicon_status_success.png`
`group`	String	Example: `success`
`has_details`	Boolean	Example: `true`
`icon`	String	Example: `status_success`
`illustration`	Object	Example: `{"content":"This job depends on upstream jobs that need to succeed in order for this job to be triggered","image":"illustrations/empty-state/empty-job-not-triggered-md.svg","size":"","title":"This job has not been triggered yet"}`
`label`	String	Example: `passed`
`text`	String	Example: `passed`
`tooltip`	String	Example: `passed`

Runners

Use this API to manage the runners already registered to an instance, group, or project.

To create a runner, use the `POST /user/runners` endpoint instead.

List all runners in a group

`GET /api/v4/groups/{id}/runners`

Lists all runners available in a specified group and any ancestor groups, including any allowed instance runners.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the group
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`403`	Forbidden	—
`404`	Not Found	—

Reset runner registration token

`POST /api/v4/groups/{id}/runners/reset_registration_token`

FE version

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a group

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Group Not Found	—

List all runners for a project

`GET /api/v4/projects/{id}/runners`

List all runners available in the project, including from ancestor groups and any allowed shared runners.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`scope` Query	String	Scope of runners to return. Deprecated. Use `type` or `status` instead Allowed values: `specific`, `shared`, `instance_type`, `group_type`, `project_type`, `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
	Integer	

Name	Type	Description
`per_page` Query		Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`403`	No access granted	—
`404`	Not Found	—

Assign a runner to a project

`POST /api/v4/projects/{id}/runners`

Assigns an available project runner to a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`runner_id` Required	Integer	ID of the runner

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiRunner
`400`	Bad Request	—
`403`	Runner is locked	—
`404`	Runner not found	—

Reset runner registration token

`POST /api/v4/projects/{id}/runners/reset_registration_token`

FE version

Parameters

Name	Type	Description
`id` Path, required	String	The ID of a project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Project Not Found	—

Unassign a runner from a project

`DELETE /api/v4/projects/{id}/runners/{runner_id}`

Unassigns a specified project runner from a project. You cannot unassign a runner from the owner project. Use the delete a runner operation instead.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`runner_id` Path, required	Integer	ID of the runner

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Bad Request	—
`403`	You cannot unassign a runner from the owner project. Delete the runner instead	—
`404`	Runner not found	—
`412`	Precondition Failed	—

List all available runners

`GET /api/v4/runners`

Lists all runners available to the user. For group runners, you must have the Owner role in the owner namespace.

Parameters

Name	Type	Description
`scope` Query	String	Scope of runners to return. Deprecated. Use `type` or `status` instead Allowed values: `specific`, `shared`, `instance_type`, `group_type`, `project_type`, `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`401`	Unauthorized	—

List all runners

`GET /api/v4/runners/all`

Lists all runners in the GitLab instance (project and shared). You must have either administrator access or auditor access.

Parameters

Name	Type	Description
`scope` Query	String	Scope of runners to return. Deprecated. Use `type` or `status` instead Allowed values: `specific`, `shared`, `instance_type`, `group_type`, `project_type`, `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`type` Query	String	Type of runners to return Allowed values: `instance_type`, `group_type`, `project_type`
`paused` Query	Boolean	If `true`, returns only runners that are ignoring new jobs. If `false`, returns only runners that are accepting new jobs
`status` Query	String	Status of runners to return Allowed values: `active`, `paused`, `online`, `offline`, `never_contacted`, `stale`
`tag_list` Query	Array of strings	List of runner tags Example: `["macos","shell"]`
`version_prefix` Query	String	Version prefix of runners to return Pattern: `(?<=^|\n(?!$))[\d+.]+` Example: `15.1.`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Scope contains invalid value	—
`401`	Unauthorized	—

Reset the runner registration token for the instance

`POST /api/v4/runners/reset_registration_token`

Resets the runner registration token for the GitLab instance.

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`403`	Forbidden	—

Retrieve details on a runner

`GET /api/v4/runners/{id}`

Retrieves details of a runner. Instance runner details are available to all authenticated users through this endpoint. For groups and projects, you must have the Maintainer or Owner role for the associated project or group.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner
`include_projects` Query	Boolean	If `true`, includes projects in the response. If `false`, improves performance for runners with many projects Default: `true`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunnerDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

Update a runner

`PUT /api/v4/runners/{id}`

Updates a specified runner.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner

Request body (`application/json`)

Property	Type	Description
`access_level`	String	Access level of the runner. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided Allowed values: `not_protected`, `ref_protected`
`active`	Boolean	If `true`, the runner is allowed to receive jobs. Deprecated. Use `paused` instead. Mutually exclusive with `paused`. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`description`	String	Description of the runner. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`locked`	Boolean	If `true`, the runner is locked. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`maintenance_note`	String	Free-form maintenance notes for the runner. Limited to 1024 characters. At least one of

Property	Type	Description
		`description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`maximum_timeout`	Integer	Maximum time, in seconds, that runners can spend running a job. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`paused`	Boolean	If `true`, the runner ignores new jobs. Mutually exclusive with `active`. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`run_untagged`	Boolean	If `true`, the runner can execute untagged jobs. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_timeout`, `maintenance_note` must be provided
`tag_list`	Array of strings	List of tags for the runner. At least one of `description`, `active`, `paused`, `tag_list`, `run_untagged`, `locked`, `access_level`, `maximum_`

Property	Type	Description
		`timeout`, `maintenance_note` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunnerDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

Delete a runner

`DELETE /api/v4/runners/{id}`

Deletes a specified runner.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunner
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Runner associated with more than one project	—
`404`	Runner not found	—
`412`	Precondition Failed	—

List all jobs processed by a runner

`GET /api/v4/runners/{id}/jobs`

Lists all jobs that are being processed or were processed by a specified runner. The list of jobs is limited to projects where the user has the Reporter, Developer, Maintainer, or Owner role.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner
`system_id` Query	String	System ID associated with the runner manager
`status` Query	String	Status of the job Allowed values: `created`, `waiting_for_resource`, `preparing`, `waiting_for_callback`, `pending`, `running`, `success`, `failed`, `canceling`, `canceled`, `skipped`, `manual`, `scheduled`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`
`sort` Query	String	Sort results in ascending or descending order. Specify `order_by` as well, including for `id` Allowed values: `asc`, `desc` Default: `desc`
`cursor` Query	String	Cursor for obtaining the next set of records
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiJobBasicWithProject
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

List all managers for a runner

`GET /api/v4/runners/{id}/managers`

List all managers for a specified runner.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiRunnerManager
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Get projects associated with a runner

`GET /api/v4/runners/{id}/projects`

Get a paginated list of all projects associated with the specified runner. Access is restricted based on user permissions.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesBasicProjectDetails
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	No access granted	—
`404`	Runner not found	—

Reset an authentication token for a runner

`POST /api/v4/runners/{id}/reset_authentication_token`

Resets the authentication token for a specified runner.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the runner

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiResetTokenResult
`400`	Bad Request	—
`403`	No access granted	—
`404`	Runner not found	—
`422`	Unprocessable Entity	—

Create a runner owned by currently authenticated user

`POST /api/v4/user/runners`

Create a new runner

Request body (`application/json`)

Property	Type	Description
`access_level`	String	Access level of the runner Allowed values: `not_protected`, `ref_protected`
`description`	String	Description of the runner
`group_id` Required	Integer	ID of the group to create the runner in. Required if `runner_type` is `group_type` Example: `1`
`locked`	Boolean	Specifies if the runner should be locked for the current project Default: `false`
`maintenance_note`	String	Free-form maintenance notes for the runner. Limited to 1024 characters
`maximum_timeout`	Integer	Maximum time, in seconds, that runners can spend running a job
`paused`	Boolean	If `true`, the runner ignores new jobs
`project_id` Required	Integer	ID of the project to create the runner in. Required if `runner_type` is `project_type` Example: `1`
`run_untagged`	Boolean	Specifies if the runner should handle untagged jobs Default: `true`
`runner_type` Required	String	Scope of the runner Allowed values: `instance_type`, `group_type`, `project_type` Minimum length: `1`
`tag_list`	Array of strings	

Property	Type	Description
		Comma-separated list of runner tags

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiRunnerRegistrationDetails
`400`	Bad Request	—
`403`	Forbidden	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicGroupDetails

Property	Type	Description
`id`	Integer (int64)	—
`name`	String	Example: `Diaspora`
`web_url`	String	Example: `http://gitlab.example.com/groups/diaspora`

APIEntitiesBasicProjectDetails

Property	Type	Description
`avatar_url`	String	Example: `http://example.com/uploads/project/avatar/3/uploads/avatar.png`
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`custom_attributes`	APIEntitiesCustomAttribute	—
`default_branch`	String	Example: `main`
`description`	String	Example: `desc`
`forks_count`	Integer	Example: `1`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/gitlab/gitlab.git`
`id`	Integer (int64)	Example: `1`
`last_activity_at`	String (date-time)	Example: `2013-09-30T13:46:02Z`
`license`	APIEntitiesLicenseBasic	—
`license_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/LICENCE`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`namespace`	APIEntitiesNamespaceBasic	—
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`
`readme_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab/blob/master/README.md`

Property	Type	Description
`repository_storage`	String	Example: `default`
`ssh_url_to_repo`	String	Example: `git@gitlab.example.com:gitlab/gitlab.git`
`star_count`	Integer	Example: `1`
`tag_list`	Array of strings	Example: `["tag"]`
`topics`	Array of strings	Example: `["topic"]`
`visibility`	String	Example: `public`
`web_url`	String	Example: `https://gitlab.example.com/gitlab/gitlab`

APIEntitiesCiJobBasicWithProject

Property	Type	Description
`allow_failure`	Boolean	—
`commit`	APIEntitiesCommit	—
`coverage`	Number (float)	Example: `98.29`
`created_at`	String (date-time)	Example: `2015-12-24T15:51:21.880Z`
`duration`	Number (float)	Time spent running Example: `0.465`
`erased_at`	String (date-time)	Example: `2015-12-24T18:00:29.728Z`
`failure_reason`	String	Example: `script_failure`
`finished_at`	String (date-time)	Example: `2015-12-24T17:54:31.198Z`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `deploy_to_production`
`pipeline`	APIEntitiesCiPipelineBasic	—
`project`	APIEntitiesProjectIdentity	—
`queued_duration`	Number (float)	Time spent enqueued Example: `0.123`
`ref`	String	Example: `main`
`stage`	String	Example: `deploy`
`started_at`	String (date-time)	Example: `2015-12-24T17:54:30.733Z`
`status`	String	Example: `waiting_for_resource`
`tag`	Boolean	—
`user`	APIEntitiesUser	—
`web_url`	String	

Property	Type	Description
		Example: `https://example.com/foo/bar/-/jobs/1`

APIEntitiesCiPipelineBasic

Property	Type	Description
`created_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`id`	Integer (int64)	Example: `1`
`iid`	Integer	Example: `2`
`project_id`	Integer (int64)	Example: `3`
`ref`	String	Example: `feature-branch`
`sha`	String	Example: `0ec9e58fdfca6cdd6652c083c9edb53abc0bad52`
`source`	String	Example: `push`
`status`	String	Example: `success`
`updated_at`	String (date-time)	Example: `2022-10-21T16:49:48.000+02:00`
`web_url`	String	Example: `https://gitlab.example.com/gitlab-org/gitlab-foss/-/pipelines/61`

APIEntitiesCiResetTokenResult

Property	Type	Description
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCiRunner

Property	Type	Description
`active`	Boolean	Example: `true`
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`runner_type`	String	Allowed values: `instance_type`, `group_type`, `project_type` Example: `instance_type`
`status`	String	Example: `online`

APIEntitiesCiRunnerDetails

Property	Type	Description
`access_level`	String	—
`active`	Boolean	Example: `true`
`architecture`	String	—
`contacted_at`	String	—
`created_at`	String (date-time)	Example: `2025-05-03T00:00:00.000Z`
`created_by`	APIEntitiesUserBasic	—
`description`	String	Example: `test-1-20150125`
`groups`	APIEntitiesBasicGroupDetails	—
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`is_shared`	Boolean	Example: `true`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`locked`	String	—
`maintenance_note`	String	—
`maximum_timeout`	String	—
`name`	String	Example: `test`
`online`	Boolean	Example: `true`
`paused`	Boolean	Example: `false`
`platform`	String	—
`projects`	APIEntitiesBasicProjectDetails	—
`revision`	String	—
`run_untagged`	String	—
`runner_type`	String	Allowed values: `instance_type`, `group_type`,

Property	Type	Description
		`project_type` Example: `instance_type`
`status`	String	Example: `online`
`tag_list`	String	—
`token_expires_at`	String (date-time)	Example: `2025-01-01T00:00:00.000Z`
`version`	String	—

APIEntitiesCiRunnerManager

Property	Type	Description
`architecture`	String	Example: `amd64`
`contacted_at`	String	Example: `2023-10-24T01:27:06.549Z`
`created_at`	String	Example: `2023-10-24T01:27:06.549Z`
`id`	Integer (int64)	Example: `8`
`ip_address`	String	Example: `127.0.0.1`
`job_execution_status`	String	Allowed values: `active`, `idle` Example: `idle`
`platform`	String	Example: `linux`
`revision`	String	Example: `91a27b2a`
`status`	String	Example: `online`
`system_id`	String	Example: `runner-1`
`version`	String	Example: `16.11.0`

APIEntitiesCiRunnerRegistrationDetails

Property	Type	Description
`id`	String	—
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesLicenseBasic

Property	Type	Description
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`source_url`	String	—

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Search

Use this API to search across GitLab in an instance, a group, or a project, and to retrieve information about advanced search migrations.

Every call to this API requires authentication. Retrieving advanced search migrations requires administrator access.

Some scopes are available for basic search. When advanced search or exact code search is enabled, additional scopes become available. To use basic search instead, see specify a search type.

These endpoints support offset-based pagination.

Search on GitLab within a group

`GET /api/v4/groups/{id}/(-/)search`

This feature was introduced in GitLab 10.5.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`search` Query, required	String	Search query
`scope` Query, required	String	Scope of the search Allowed values: `projects`, `groups`, `issues`, `work_items`, `merge_requests`, `milestones`, `users` Minimum length: `1`
`state` Query	String	Filter results by state. Applies only to the `issues`, `work_items`, and `merge_requests` scopes Allowed values: `all`, `opened`, `closed`, `merged`
`confidential` Query	Boolean	If `true`, returns only confidential results. If `false`, returns only non-confidential results. Applies only to the `issues` and `work_items` scopes
`type` Query	Array of strings	Filter work items by type. Only applies to work_items scope. Available types: issue, task, incident, ticket
`include_archived` Query	Boolean	If `true`, includes archived projects in the search. Introduced in GitLab 18.7 Default: `false`
`fields` Query	Array of strings	Array of fields to search. Only “title” is supported (work_items / issues scope)
	Array of strings	

Name	Type	Description
`language` Query		Array of languages to filter blobs by language
`label_name` Query	Array of strings	Array of labels to filter work_ items / issues / merge_ requests by label
`source_branch` Query	String	Filter merge requests by source branch Maximum length: `255`
`target_branch` Query	String	Filter merge requests by target branch Maximum length: `255`
`not_source_branch` Query	String	Exclude merge requests by source branch Maximum length: `255`
`not_target_branch` Query	String	Exclude merge requests by target branch Maximum length: `255`
`author_username` Query	String	Filter merge requests by author username Maximum length: `255`
`not_author_username` Query	String	Exclude merge requests by author username Maximum length: `255`
`exclude_forks` Query	Boolean	Exclude forked projects from search results. Only supported for work_items / issues scope
`num_context_lines` Query	Integer	Number of context lines around each match. Only supported for blobs scope Maximum: `20` Minimum: `0`
`page` Query	Integer	Page number of results to retrieve

Name	Type	Description
		Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Search a project

`GET /api/v4/projects/{id}/(-/)search`

Searches for content in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`search` Query, required	String	Search query
`scope` Query, required	String	Scope of the search Allowed values: `issues`, `work_items`, `merge_requests`, `milestones`, `notes`, `wiki_blobs`, `commits`, `blobs`, `users` Minimum length: `1`
`ref` Query	String	Name of the repository branch or tag. If omitted, uses the default branch
`state` Query	String	Filter results by state. Applies only to the `issues`, `work_items`, and `merge_requests` scopes Allowed values: `all`, `opened`, `closed`, `merged`
`confidential` Query	Boolean	If `true`, returns only confidential results. If `false`, returns only non-confidential results. Applies only to the `issues` and `work_items` scopes
`type` Query	Array of strings	Filter work items by type. Only applies to work_items scope. Available types: issue, task, incident, ticket
`fields` Query	Array of strings	Array of fields to search. Only “title” is supported (work_items / issues scope)
	Array of strings	

Name	Type	Description
`language` Query		Array of languages to filter blobs by language
`label_name` Query	Array of strings	Array of labels to filter work_ items / issues / merge_ requests by label
`source_branch` Query	String	Filter merge requests by source branch Maximum length: `255`
`target_branch` Query	String	Filter merge requests by target branch Maximum length: `255`
`not_source_branch` Query	String	Exclude merge requests by source branch Maximum length: `255`
`not_target_branch` Query	String	Exclude merge requests by target branch Maximum length: `255`
`author_username` Query	String	Filter merge requests by author username Maximum length: `255`
`not_author_username` Query	String	Exclude merge requests by author username Maximum length: `255`
`num_context_lines` Query	Integer	Number of context lines around each match. Only supported for blobs scope Maximum: `20` Minimum: `0`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page

Name	Type	Description
		Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Search an instance

`GET /api/v4/search`

Searches for a term across the entire GitLab instance. The response depends on the requested scope.

Parameters

Name	Type	Description
`search` Query, required	String	Search query
`scope` Query, required	String	Scope of the search Allowed values: `projects`, `groups`, `issues`, `work_items`, `merge_requests`, `milestones`, `snippet_titles`, `users` Minimum length: `1`
`state` Query	String	Filter results by state. Applies only to the `issues`, `work_items`, and `merge_requests` scopes Allowed values: `all`, `opened`, `closed`, `merged`
`confidential` Query	Boolean	If `true`, returns only confidential results. If `false`, returns only non-confidential results. Applies only to the `issues` and `work_items` scopes
`type` Query	Array of strings	Filter work items by type. Only applies to work_items scope. Available types: issue, task, incident, ticket
`include_archived` Query	Boolean	If `true`, includes archived projects in the search. Introduced in GitLab 18.7 Default: `false`
`fields` Query	Array of strings	Array of fields to search. Only “title” is supported (work_items / issues scope)
`language` Query	Array of strings	Array of languages to filter blobs by language

Name	Type	Description
`label_name` Query	Array of strings	Array of labels to filter work_items / issues / merge_requests by label
`source_branch` Query	String	Filter merge requests by source branch Maximum length: `255`
`target_branch` Query	String	Filter merge requests by target branch Maximum length: `255`
`not_source_branch` Query	String	Exclude merge requests by source branch Maximum length: `255`
`not_target_branch` Query	String	Exclude merge requests by target branch Maximum length: `255`
`author_username` Query	String	Filter merge requests by author username Maximum length: `255`
`not_author_username` Query	String	Exclude merge requests by author username Maximum length: `255`
`exclude_forks` Query	Boolean	Exclude forked projects from search results. Only supported for work_items / issues scope
`num_context_lines` Query	Integer	Number of context lines around each match. Only supported for blobs scope Maximum: `20` Minimum: `0`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
	Integer	

Name	Type	Description
`per_page` Query		Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—

Secure files

Use this API to manage secure files for a project.

List all secure files for a project

`GET /api/v4/projects/{id}/secure_files`

Lists all secure files for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiSecureFile
`400`	Bad Request	—
`404`	Not Found	—

Create a secure file

`POST /api/v4/projects/{id}/secure_files`

Creates a secure file in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The secure file being uploaded
`name` Required	String	The name of the file being uploaded. The filename must be unique within the project

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiSecureFile
`400`	400 Bad Request	—
`404`	Not Found	—

Retrieve details of a secure file

`GET /api/v4/projects/{id}/secure_files/{secure_file_id}`

Retrieves details of a specified secure file in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`secure_file_id` Path, required	Integer	ID of the secure file

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCiSecureFile
`400`	Bad Request	—
`404`	404 Not found	—

Delete a secure file

`DELETE /api/v4/projects/{id}/secure_files/{secure_file_id}`

Deletes a specified secure file from a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`secure_file_id` Path, required	Integer	ID of the secure file

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	Bad Request	—
`404`	404 Not found	—

Download a secure file

`GET /api/v4/projects/{id}/secure_files/{secure_file_id}/download`

Downloads the contents of a specified secure file in a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`secure_file_id` Path, required	Integer	ID of the secure file

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`404`	404 Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiSecureFile

Property	Type	Description
`checksum`	String	Example: `16630b189ab34b2e3504f4758e1054d2e478deda510b2b08cc0ef38d12e80aac`
`checksum_algorithm`	String	Example: `sha256`
`created_at`	String (date-time)	Example: `2022-02-22T22:22:22.222Z`
`expires_at`	String (date-time)	Example: `2023-09-21T14:55:59.000Z`
`file_extension`	String	Example: `jks`
`id`	Integer (int64)	Example: `123`
`metadata`	Object	Example: `{"id":"759499105426963432432644053776584439 14"}`
`name`	String	Example: `upload-keystore.jks`

Service accounts

Use this API to interact with service accounts.

The number of service accounts you can create depends on your subscription and offering:

- On GitLab Premium and Ultimate, you can create an unlimited number of service accounts for all offerings.
- On GitLab Free, limits vary by offering:
  - For GitLab.com, you can create up to 100 service accounts for each top-level group. This includes service accounts created in subgroups or projects.
  - For GitLab Self-Managed, you can create up to 100 service accounts for the entire instance. This includes service accounts created for the instance, a group, or a project.

You can also interact with service accounts through the users API. To manage SSH keys for service accounts, use the user SSH and GPG keys API.

List all group service accounts

`GET /api/v4/groups/{id}/service_accounts`

Lists all service accounts in a specified group. Use the `page` and `per_page` pagination parameters to filter the results.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `username` Default: `id`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Group not found	—

Create a group service account

`POST /api/v4/groups/{id}/service_accounts`

Creates a service account in a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group

Request body (`application/json`)

Property	Type	Description
`email`	String	Email address of the user account. If omitted when creating a new service account, generates an email address prefixed with `service_account_group_`. Custom email addresses require confirmation, unless the group has a matching verified domain or email confirmation settings are turned off
`name`	String	Name of the user. If omitted when creating a new service account, uses `Service account user`
`username`	String	Username of the user account. If omitted when creating a new service account, generates a username prefixed with `service_account_group_`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Group not found	—

Get a single group service account

`GET /api/v4/groups/{id}/service_accounts/{user_id}`

Gets a specified service account in a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 User not found	—

Update a group service account

`PATCH /api/v4/groups/{id}/service_accounts/{user_id}`

Update a specified group service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the service account

Request body (`application/json`)

Property	Type	Description
`email`	String	Email address of the user account. If omitted when creating a new service account, generates an email address prefixed with `service_account_group_`. Custom email addresses require confirmation, unless the group has a matching verified domain or email confirmation settings are turned off
`name`	String	Name of the user. If omitted when creating a new service account, uses `Service account user`
`username`	String	Username of the user account. If omitted when creating a new service account, generates a username prefixed with `service_account_group_`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 User not found	—

Delete a group service account

`DELETE /api/v4/groups/{id}/service_accounts/{user_id}`

Deletes a specified group service account. Available only for group Owners and administrators.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the service account
`hard_delete` Query	Boolean	If `true`, contributions that would usually be moved to a ghost user are deleted instead, along with groups owned solely by this service account

Responses

Code	Description	Schema
`204`	Resource deleted	—
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Group not found	—

List all personal access tokens for a group service account

`GET /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens`

Lists all personal access tokens for a specified group service account

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`user_id` Path, required	Integer	ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	Forbidden	—
`404`	404 Group Not Found	—

Create a personal access token for a group service account

`POST /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens`

Creates a personal access token for a specified group service account. Available only for group Owners and administrators. This feature was introduced in GitLab 16.1.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`user_id` Path, required	Integer	ID of the service account

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	Array of approved scopes. For a list of possible values, see access token scopes

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a personal access token for a group service account

`DELETE /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}`

Revokes a specified personal access token for a group service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the personal access token
`user_id` Path, required	Integer	ID of the service account

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Rotate a personal access token for a group service account

`POST /api/v4/groups/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}/rotate`

Rotates a specified personal access token for a group service account. This revokes the existing token and creates a token with the same name, description, and scopes.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group
`token_id` Path, required	Integer	ID of the personal access token
`user_id` Path, required	Integer	ID of the service account

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	Date when the access token expires. Introduced in GitLab 17.9. If omitted, the new token expires one week after rotation when token expiry is enforced, and does not expire otherwise Example: `2021-01-31`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all project service accounts

`GET /api/v4/projects/{id}/service_accounts`

Lists all service accounts in a specified project. Use the `page` and `per_page` pagination parameters to filter the results.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Attribute to sort by Allowed values: `id`, `username` Default: `id`
`sort` Query	String	Order of sorting Allowed values: `asc`, `desc` Default: `desc`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Project not found	—

Create a project service account

`POST /api/v4/projects/{id}/service_accounts`

Creates a service account in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`email`	String	Custom email address for the user
`name`	String	Name of the user
`username`	String	Username of the user

Responses

Code	Description	Schema
`201`	Created	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Project not found	—

Get a single project service account

`GET /api/v4/projects/{id}/service_accounts/{user_id}`

Gets a specified service account in a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`user_id` Path, required	Integer	The ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 User not found	—

Update a project service account

`PATCH /api/v4/projects/{id}/service_accounts/{user_id}`

Update a specified project service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`user_id` Path, required	Integer	The ID of the service account

Request body (`application/json`)

Property	Type	Description
`email`	String	Custom email address for the user
`name`	String	Name of the user
`username`	String	Username of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 User not found	—

Delete a project service account

`DELETE /api/v4/projects/{id}/service_accounts/{user_id}`

Deletes a specified project service account. Available only for project Owners, Maintainers, and administrators.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`user_id` Path, required	Integer	The ID of the service account
`hard_delete` Query	Boolean	Whether to remove a user's contributions

Responses

Code	Description	Schema
`204`	No Content	—
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Project not found	—

List all personal access tokens for a project service account

`GET /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens`

Lists all personal access tokens for a specified project service account

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`revoked` Query	Boolean	If `true`, only returns revoked tokens. If `false`, only returns tokens that are not revoked Example: `false`
`state` Query	String	Return tokens with the specified state Allowed values: `active`, `inactive` Example: `active`
`created_before` Query	String (date-time)	Return tokens created on or before the specified time Example: `2022-01-01T00:00:00Z`
`created_after` Query	String (date-time)	Return tokens created on or after the specified time Example: `2021-01-01T00:00:00Z`
`last_used_before` Query	String (date-time)	Return tokens last used on or before the specified time Example: `2021-01-01T00:00:00Z`
`last_used_after` Query	String (date-time)	Return tokens last used on or after the specified time Example: `2022-01-01T00:00:00Z`
`expires_before` Query	String (date)	Return tokens that expire before the specified date Example: `2022-01-01`
`expires_after` Query	String (date)	Return tokens that expire on or after the specified date Example: `2021-01-01`
	String	

Name	Type	Description
`search` Query		Return tokens that include the specified value in their name Example: `token`
`sort` Query	String	Sort results by the specified field and direction. Possible values are `created_asc`, `created_desc`, `expires_asc`, `expires_desc`, `last_used_asc`, `last_used_desc`, `name_asc`, and `name_desc` Example: `created_desc`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`user_id` Path, required	Integer	The ID of the service account

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalAccessToken
`400`	Bad Request	—
`401`	401 Unauthorized	—
`403`	Forbidden	—
`404`	404 Project Not Found	—

Create a personal access token for a project service account

`POST /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens`

Creates a personal access token for a specified project service account. Available only for project Owners, Maintainers, and administrators.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`user_id` Path, required	Integer	The ID of the service account

Request body (`application/json`)

Property	Type	Description
`description`	String	Description of the access token. Limited to 255 characters Example: `A token used for k8s`
`expires_at`	String (date)	Expiration date of the access token in ISO format (YYYY-MM-DD). If undefined, the date is set to the maximum allowable lifetime limit Example: `2021-01-31`
`name` Required	String	Name of the access token Example: `My token`
`scopes` Required	Array of strings	The array of scopes of the personal access token

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

Revoke a personal access token for a project service account

`DELETE /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}`

Revokes a specified personal access token for a project service account.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`token_id` Path, required	Integer	The ID of the personal access token
`user_id` Path, required	Integer	The ID of the service account

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Rotate a personal access token for a project service account

`POST /api/v4/projects/{id}/service_accounts/{user_id}/personal_access_tokens/{token_id}/rotate`

Rotates a specified personal access token for a project service account. This revokes the existing token and creates a token with the same name, description, and scopes.

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project
`token_id` Path, required	Integer	The ID of the personal access token
`user_id` Path, required	Integer	The ID of the service account

Request body (`application/json`)

Property	Type	Description
`expires_at`	String (date)	The expiration date of the token Example: `2021-01-31`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalAccessTokenWithToken
`400`	Bad Request	—
`404`	Not Found	—

List all instance service accounts

`GET /api/v4/service_accounts`

Lists all instance service accounts. Use the `page` and `per_page` pagination parameters to filter the results.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`order_by` Query	String	Attribute to sort by Allowed values: `id`, `username` Default: `id`
`sort` Query	String	Order of sorting Allowed values: `asc`, `desc` Default: `desc`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Create an instance service account

`POST /api/v4/service_accounts`

Creates an instance service account.

Request body (`application/json`)

Property	Type	Description
`email`	String	Custom email address for the user
`external`	Boolean	Flag indicating the user is an external user
`name`	String	Name of the user
`username`	String	Username of the user

Responses

Code	Description	Schema
`201`	Created	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—

Update an instance service account

`PATCH /api/v4/service_accounts/{user_id}`

Updates a specified instance service account.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	The ID of the service account user

Request body (`application/json`)

Property	Type	Description
`email`	String	Custom email address for the user
`external`	Boolean	Flag indicating the user is an external user
`name`	String	Name of the user
`username`	String	Username of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesServiceAccount
`400`	400 Bad request	—
`401`	401 Unauthorized	—
`403`	403 Forbidden	—
`404`	404 Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesPersonalAccessToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`user_id`	Integer (int64)	Example: `3`

APIEntitiesPersonalAccessTokenGranularScope

Property	Type	Description
`access`	String	Example: `personal_projects`
`group_id`	Integer (int64)	Example: `5`
`permissions`	Array	Example: `["read_job"]`
`project_id`	Integer (int64)	Example: `3`

APIEntitiesPersonalAccessTokenWithToken

Property	Type	Description
`active`	Boolean	—
`created_at`	String (date-time)	—
`description`	String	Example: `Token to manage api`
`expires_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`granular`	Boolean	—
`granular_scopes`	Array of APIEntitiesPersonalAccessTokenGranularScope	—
`id`	Integer (int64)	Example: `2`
`last_used_at`	String (date-time)	Example: `2020-08-31T15:53:00.073Z`
`last_used_ips`	Array of strings	The five most recent unique IP addresses that have authenticated with this token. When the limit is reached, the oldest IP address is removed. The list updates once per minute per token Example: `["127.0.0.1","127.0.0.2","127.0.0.3"]`
`name`	String	Example: `John Doe`
`revoked`	Boolean	—
`scopes`	Array	Example: `["api"]`
`token`	String	—
`user_id`	Integer (int64)	Example: `3`

APIEntitiesServiceAccount

Property	Type	Description
`email`	String	Example: `service_account@example.com`
`external`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`unconfirmed_email`	String	Example: `updated_service_account@example.com`
`username`	String	Example: `admin`

Sidekiq

Use this API to retrieve information about the current state of Sidekiq, including its jobs, queues, and processes, and to delete jobs from a queue.

Delete jobs from a Sidekiq queue

`DELETE /api/v4/admin/sidekiq/queues/{queue_name}`

Deletes jobs from a Sidekiq queue that match the specified metadata.

Parameters

Name	Type	Description
`queue_name` Path, required	String	Name of the queue to delete jobs from
`organization_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`user` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`,

Name	Type	Description
		`artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`user_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`gl_user_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_`

Name	Type	Description
		`namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`scoped_user` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`,

Name	Type	Description
		`client_version`, `worker_class` must be provided
`scoped_user_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`project` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_coun

Name	Type	Description
		`t`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`root_namespace` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`gl_root_namespace_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`,

Name	Type	Description
		`caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`client_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided

Name	Type	Description
`caller_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`remote_ip` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`,

Name	Type	Description
		`bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`job_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`pipeline_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_`

Name	Type	Description
		`class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`related_class` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
	String	Metadata key to match. At least one of `organization_id`,

Name	Type	Description
`feature_category` Query		`user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`artifact_size` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_r`

Name	Type	Description
		`edis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`artifact_used_cdn` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`artifacts_dependencies_size` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_`

Name	Type	Description
		`used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`artifacts_dependencies_count` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`root_caller_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_`

Name	Type	Description
		`id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`merge_action_status` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`,

Name	Type	Description
		`client_type`, `client_name`, `client_version`, `worker_class` must be provided
`bulk_import_entity_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`sidekiq_destination_shard_redis` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`,

Name	Type	Description
		`artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`kubernetes_agent_id` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`mvcc_manifest` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_`

Name	Type	Description
		`namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`client_type` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`,

Name	Type	Description
		`client_version`, `worker_class` must be provided
`client_name` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`client_version` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_coun

Name	Type	Description
		`t`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided
`worker_class` Query	String	Metadata key to match. At least one of `organization_id`, `user`, `user_id`, `gl_user_id`, `scoped_user`, `scoped_user_id`, `project`, `root_namespace`, `gl_root_namespace_id`, `client_id`, `caller_id`, `remote_ip`, `job_id`, `pipeline_id`, `related_class`, `feature_category`, `artifact_size`, `artifact_used_cdn`, `artifacts_dependencies_size`, `artifacts_dependencies_count`, `root_caller_id`, `merge_action_status`, `bulk_import_entity_id`, `sidekiq_destination_shard_redis`, `kubernetes_agent_id`, `mvcc_manifest`, `client_type`, `client_name`, `client_version`, `worker_class` must be provided

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List all Sidekiq metrics

`GET /api/v4/sidekiq/compound_metrics`

Lists all Sidekiq metrics. Includes queue, process, and job completion metrics.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—

Retrieve job completion metrics

`GET /api/v4/sidekiq/job_stats`

Retrieves statistics on the completion status of all Sidekiq jobs.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—

List all Sidekiq processes

`GET /api/v4/sidekiq/process_metrics`

Lists all details about all registered Sidekiq worker processes, including hostname, process ID, queues, and concurrency settings.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—

List all job queue metrics

`GET /api/v4/sidekiq/queue_metrics`

Lists all details about all Sidekiq job queues, including backlog size and latency.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—

Snippets

Use this API to manage personal and project snippets.

To move snippets between storages, use the snippet repository storage moves API.

List all snippets for a project

`GET /api/v4/projects/{id}/snippets`

Lists all snippets for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProject Snippet
`400`	Bad Request	—
`404`	Not found	—

Create a project snippet

`POST /api/v4/projects/{id}/snippets`

Creates a project snippet. The user must have permission to create snippets.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`content`	String	The content of a snippet. Mutually exclusive with `files`. Exactly one of `files`, `content` must be provided
`description`	String	Description of the snippet
`file_name` Required	String	The name of a snippet file
`files`	Array of objects	Array of files for the snippet. Mutually exclusive with `content`. Exactly one of `files`, `content` must be provided
`files[].content` Required	String	Content of the snippet file
`files[].file_path` Required	String	Path of the snippet file
`title` Required	String	Title of the snippet
`visibility` Required	String	Visibility level for the snippet. On GitLab.com, the `internal` value is not available Allowed values: `private`, `internal`, `public` Minimum length: `1`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectSnippet
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Retrieve a project snippet

`GET /api/v4/projects/{id}/snippets/{snippet_id}`

Retrieves a specified project snippet.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectSnippet
`400`	Bad Request	—
`404`	Not found	—

Update a project snippet

`PUT /api/v4/projects/{id}/snippets/{snippet_id}`

Updates a specified project snippet. The user must have permission to modify snippets. Updates to snippets with multiple files must use the `files` attribute.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID of the snippet

Request body (`application/json`)

Property	Type	Description
`content`	String	Content of the snippet. Deprecated. Use `files` instead. Mutually exclusive with `files`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`description`	String	Description of the snippet. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`file_name`	String	Name of the snippet file. Deprecated. Use `files` instead. Mutually exclusive with `files`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`files`	Array of objects	Array of files to update for the snippet. Required when the snippet has multiple files. Mutually exclusive with `content`, `file_name`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`files[].action` Required	String	The type of action to perform on the file, must be one of: create, update, delete, move Allowed values: `create`, `update`, `delete`, `move` Minimum length: `1`
`files[].content`	String	Content of the snippet file
`files[].file_path`	String	Path of the snippet file

Property	Type	Description
`files[].previous_path`	String	Previous path of the snippet file
`title`	String	Title of the snippet. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`visibility`	String	Visibility level for the snippet. On GitLab.com, the `internal` value is not available. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided Allowed values: `private`, `internal`, `public`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectSnippet
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a project snippet

`DELETE /api/v4/projects/{id}/snippets/{snippet_id}`

Deletes a specified project snippet.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Validation error	—
`404`	Not found	—

Retrieve snippet repository file content

`GET /api/v4/projects/{id}/snippets/{snippet_id}/files/{ref}/{file_path}/raw`

Retrieves the raw file content from a snippet repository as plain text.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`ref` Path, required	String	The name of branch, tag or commit
`file_path` Path, required	String	The URL-encoded path to the file, like lib%2Fclass%2Erb
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectSnippet
`400`	Bad Request	—
`404`	Not found	—

Retrieve a raw project snippet

`GET /api/v4/projects/{id}/snippets/{snippet_id}/raw`

Retrieves the raw contents of a specified project snippet as plain text

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectSnippet
`400`	Bad Request	—
`404`	Not found	—

Retrieve user agent details for a project snippet

`GET /api/v4/projects/{id}/snippets/{snippet_id}/user_agent_detail`

Retrieves user agent details for a specified snippet. Available only for administrators.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`snippet_id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserAgentDetail
`400`	Bad Request	—
`404`	Not found	—

List all snippets for current user

`GET /api/v4/snippets`

Lists all snippets for the currently authenticated user.

Parameters

Name	Type	Description
`created_after` Query	String (date-time)	Return snippets created on or after the specified time
`created_before` Query	String (date-time)	Return snippets created on or before the specified time
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSnippet
`400`	Bad Request	—
`404`	Not found	—

Create a snippet

`POST /api/v4/snippets`

Creates a snippet.

Request body (`application/json`)

Property	Type	Description
`content`	String	The content of a snippet. Mutually exclusive with `files`. Exactly one of `files`, `content` must be provided
`description`	String	Description of the snippet
`file_name` Required	String	The name of a snippet file
`files`	Array of objects	Array of files for the snippet. Mutually exclusive with `content`. Exactly one of `files`, `content` must be provided
`files[].content` Required	String	Content of the snippet file
`files[].file_path` Required	String	Path of the snippet file
`title` Required	String	Title of the snippet
`visibility`	String	Visibility level for the snippet. On GitLab.com, the `internal` value is not available Allowed values: `private`, `internal`, `public` Default: `internal`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesPersonalSnippet
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

List all snippets

`GET /api/v4/snippets/all`

Lists all snippets available to the currently authenticated user. Users with Administrator or Auditor access levels can see all snippets (both personal and project). This feature was introduced in GitLab 16.3.

Parameters

Name	Type	Description
`created_after` Query	String (date-time)	Return snippets created on or after the specified time
`created_before` Query	String (date-time)	Return snippets created on or before the specified time
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSnippet
`400`	Bad Request	—
`404`	Not found	—

List all public snippets

`GET /api/v4/snippets/public`

Lists all public snippets accessible to the currently authenticated user.

Parameters

Name	Type	Description
`created_after` Query	String (date-time)	Return snippets created on or after the specified time
`created_before` Query	String (date-time)	Return snippets created on or before the specified time
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesPersonal Snippet
`400`	Bad Request	—
`404`	Not found	—

Retrieve a snippet

`GET /api/v4/snippets/{id}`

Retrieves a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalSnippet
`400`	Bad Request	—
`404`	Not found	—

Update snippet

`PUT /api/v4/snippets/{id}`

Updates a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the snippet

Request body (`application/json`)

Property	Type	Description
`content`	String	The content of a snippet. Mutually exclusive with `files`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`description`	String	Description of the snippet. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`file_name`	String	The name of a snippet file. Mutually exclusive with `files`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`files`	Array of objects	Array of files to update for the snippet. Required when the snippet has multiple files. Mutually exclusive with `content`, `file_name`. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`files[].action` Required	String	The type of action to perform on the file, must be one of: create, update, delete, move Allowed values: `create`, `update`, `delete`, `move` Minimum length: `1`
`files[].content`	String	Content of the snippet file
`files[].file_path`	String	Path of the snippet file
`files[].previous_path`	String	Previous path of the snippet file

Property	Type	Description
`title`	String	Title of the snippet. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided
`visibility`	String	Visibility level for the snippet. On GitLab.com, the `internal` value is not available. At least one of `content`, `description`, `files`, `file_name`, `title`, `visibility` must be provided Allowed values: `private`, `internal`, `public`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalSnippet
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete snippet

`DELETE /api/v4/snippets/{id}`

Deletes a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesPersonalSnippet
`400`	Validation error	—
`404`	Not found	—

Retrieve snippet file content

`GET /api/v4/snippets/{id}/files/{ref}/{file_path}/raw`

Retrieves the raw file content from a snippet as plain text.

Parameters

Name	Type	Description
`ref` Path, required	String	The name of branch, tag or commit
`file_path` Path, required	String	The URL-encoded path to the file, like lib%2Fclass%2Erb
`id` Path, required	String	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not found	—

Retrieve a raw snippet

`GET /api/v4/snippets/{id}/raw`

Retrieves the raw contents of a specified snippet as plain text

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not found	—

Retrieve user agent details for a snippet

`GET /api/v4/snippets/{id}/user_agent_detail`

Retrieves user agent details for a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the snippet

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserAgentDetail
`400`	Bad Request	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesPersonalSnippet

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`description`	String	Example: `Ruby test snippet`
`file_name`	String	Example: `add.rb`
`files`	Array of objects	Example: `[{"path":"file.txt","raw_url":"https://gitlab.example.com/.../raw"}]`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/snippets/65.git`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `none`
`project_id`	Integer (int64)	Example: `1`
`raw_url`	String	Example: `http://example.com/example/example/snippets/1/raw`
`ssh_url_to_repo`	String	Example: `ssh://user@gitlab.example.com/snippets/65.git`
`title`	String	Example: `test`
`updated_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`visibility`	String	Example: `public`
`web_url`	String	Example: `http://example.com/example/example/snippets/1`

APIEntitiesProjectSnippet

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`description`	String	Example: `Ruby test snippet`
`file_name`	String	Example: `add.rb`
`files`	Array of objects	Example: `[{"path":"file.txt","raw_url":"https://gitlab.example.com/.../raw"}]`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/snippets/65.git`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `none`
`project_id`	Integer (int64)	Example: `1`
`raw_url`	String	Example: `http://example.com/example/example/snippets/1/raw`
`ssh_url_to_repo`	String	Example: `ssh://user@gitlab.example.com/snippets/65.git`
`title`	String	Example: `test`
`updated_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`visibility`	String	Example: `public`
`web_url`	String	Example: `http://example.com/example/example/snippets/1`

APIEntitiesSnippet

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`description`	String	Example: `Ruby test snippet`
`file_name`	String	Example: `add.rb`
`files`	Array of objects	Example: `[{"path":"file.txt","raw_url":"https://gitlab.example.com/.../raw"}]`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/snippets/65.git`
`id`	Integer (int64)	Example: `1`
`imported`	Boolean	Example: `false`
`imported_from`	String	Example: `none`
`project_id`	Integer (int64)	Example: `1`
`raw_url`	String	Example: `http://example.com/example/example/snippets/1/raw`
`ssh_url_to_repo`	String	Example: `ssh://user@gitlab.example.com/snippets/65.git`
`title`	String	Example: `test`
`updated_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`visibility`	String	Example: `public`
`web_url`	String	Example: `http://example.com/example/example/snippets/1`

APIEntitiesUserAgentDetail

Property	Type	Description
`akismet_submitted`	Boolean	Example: `false`
`ip_address`	String	Example: `127.0.0.1`
`user_agent`	String	Example: `AppleWebKit/537.36`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustom Attribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Storage moves

Use this API to schedule and track repository storage moves for groups, projects, and snippets. Project moves cover wiki and design repositories, and group moves cover group wikis. Moving repositories can help you migrate to Gitaly Cluster (Praefect).

To ensure data integrity, GitLab places the group, project, or snippet in a temporary read-only state for the duration of the move. During this time, users receive this message if they try to push new commits:

```
The repository is temporarily read-only. Please try again later.
```

This API requires you to authenticate yourself as an administrator.

States

As GitLab processes a storage move, it transitions through different states. Values of `state` are:

- `initial`: The record has been created, but the background job has not yet been scheduled.
- `scheduled`: The background job has been scheduled.
- `started`: The repositories are being copied to the destination storage.
- `replicated`: The repositories have been moved.
- `failed`: The repositories failed to copy, or the checksums did not match.
- `finished`: The repositories have been moved, and the repositories on the source storage have been deleted.
- `cleanup failed`: The repositories have been moved, but the repositories on the source storage could not be deleted.

List all project repository storage moves

`GET /api/v4/project_repository_storage_moves`

Lists all project repository storage moves.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProjectsRepositoryStorageMove
`400`	Bad Request	—

Create repository storage moves for all projects on a storage shard

`POST /api/v4/project_repository_storage_moves`

Creates repository storage moves for each project repository stored on the source storage shard. This endpoint migrates all projects at once.

Request body (`application/json`)

Property	Type	Description
`destination_storage_name`	String	Name of the destination storage shard. If not provided, the storage is selected automatically based on storage weights
`source_storage_name` Required	String	Name of the source storage shard Minimum length: `1`

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—

Retrieve a project repository storage move

`GET /api/v4/project_repository_storage_moves/{repository_storage_move_id}`

Retrieves a specified project repository storage move.

Parameters

Name	Type	Description
`repository_storage_move_id` Path, required	Integer	ID of the project repository storage move

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

List all repository storage moves for a project

`GET /api/v4/projects/{id}/repository_storage_moves`

Lists all repository storage moves for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesProjectsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

Create a repository storage move for a project

`POST /api/v4/projects/{id}/repository_storage_moves`

Creates a repository storage move for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`destination_storage_name`	String	Name of the destination storage shard. If not provided, the storage is selected automatically based on storage weights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesProjectsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a repository storage move for a project

`GET /api/v4/projects/{id}/repository_storage_moves/{repository_storage_move_id}`

Retrieves a specified repository storage move for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`repository_storage_move_id` Path, required	Integer	ID of the project repository storage move

Responses

Code	Description	Schema
`200`	OK	APIEntitiesProjectsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

List all snippet repository storage moves

`GET /api/v4/snippet_repository_storage_moves`

Lists all snippet repository storage moves. By default, `GET` requests return 20 results at a time because the API results are paginated.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSnippetsRepositoryStorageMove
`400`	Bad Request	—

Schedule repository storage moves for all snippets on a storage shard

`POST /api/v4/snippet_repository_storage_moves`

Schedules repository storage moves for each snippet repository stored on the source storage shard. This endpoint migrates all snippets at once.

Request body (`application/json`)

Property	Type	Description
`destination_storage_name`	String	Name of the destination storage shard. If not provided, the storage is selected automatically based on storage weights
`source_storage_name` Required	String	Name of the source storage shard Minimum length: `1`

Responses

Code	Description	Schema
`202`	Accepted	—
`400`	Bad Request	—

Retrieve a snippet repository storage move

`GET /api/v4/snippet_repository_storage_moves/{repository_storage_move_id}`

Retrieves a specified snippet repository storage move.

Parameters

Name	Type	Description
`repository_storage_move_id` Path, required	Integer	ID of the snippet repository storage move

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSnippetsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

List all repository storage moves for a snippet

`GET /api/v4/snippets/{id}/repository_storage_moves`

Lists all repository storage moves for a specified snippet. By default, `GET` requests return 20 results at a time because the API results are paginated.

Parameters

Name	Type	Description
`id` Path, required	String	ID of the snippet
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesSnippets RepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

Schedule a repository storage move for a snippet

`POST /api/v4/snippets/{id}/repository_storage_moves`

Schedules a repository storage move for a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID of the snippet

Request body (`application/json`)

Property	Type	Description
`destination_storage_name`	String	Name of the destination storage shard. If not provided, the storage is selected automatically based on storage weights

Responses

Code	Description	Schema
`201`	Created	APIEntitiesSnippetsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a repository storage move for a snippet

`GET /api/v4/snippets/{id}/repository_storage_moves/{repository_storage_move_id}`

Retrieves a repository storage move for a specified snippet.

Parameters

Name	Type	Description
`id` Path, required	String	ID of the snippet
`repository_storage_move_id` Path, required	Integer	ID of the snippet repository storage move

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSnippetsRepositoryStorageMove
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesBasicSnippet

Property	Type	Description
`author`	APIEntitiesUserBasic	—
`created_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`description`	String	Example: `Ruby test snippet`
`http_url_to_repo`	String	Example: `https://gitlab.example.com/snippets/65.git`
`id`	Integer (int64)	Example: `1`
`project_id`	Integer (int64)	Example: `1`
`raw_url`	String	Example: `http://example.com/example/example/snippets/1/raw`
`ssh_url_to_repo`	String	Example: `ssh://user@gitlab.example.com/snippets/65.git`
`title`	String	Example: `test`
`updated_at`	String (date-time)	Example: `2012-06-28T10:52:04Z`
`visibility`	String	Example: `public`
`web_url`	String	Example: `http://example.com/example/example/snippets/1`

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesProjectsRepositoryStorageMove

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.234Z`
`destination_storage_name`	String	Example: `storage1`
`error_message`	String	Example: `Failed to move repository`
`id`	Integer (int64)	Example: `1`
`project`	APIEntitiesProjectIdentity	—
`source_storage_name`	String	Example: `default`
`state`	String	Example: `scheduled`

APIEntitiesSnippetsRepositoryStorageMove

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.234Z`
`destination_storage_name`	String	Example: `storage1`
`error_message`	String	Example: `Failed to move repository`
`id`	Integer (int64)	Example: `1`
`snippet`	APIEntitiesBasicSnippet	—
`source_storage_name`	String	Example: `default`
`state`	String	Example: `scheduled`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Submodules

Use this API to update Git submodules.

Update a submodule reference

`PUT /api/v4/projects/{id}/repository/submodules/{submodule}`

Updates a reference for a specified submodule.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project Example: `gitlab-org/gitlab`
`submodule` Path, required	String	URL-encoded full path to the submodule, for example `lib%2Fclass%2Erb` Example: `gitlab-org/gitlab-shell`

Request body (`application/json`)

Property	Type	Description
`branch` Required	String	Name of the branch to commit into Example: `main`
`commit_message`	String	Commit message. If omitted, a default message is used Example: `Commit message`
`commit_sha` Required	String	Full commit SHA to update the submodule to Example: `ed899a2f4b50b4370feeea94676502b42383c746`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesCommitDetail
`400`	The repository is empty	—
`401`	401 Unauthorized	—
`404`	404 Project Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCommitDetail

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by": ["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`last_pipeline`	Object	—
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`project_id`	Integer (int64)	Example: `1`
`short_id`	String	Example: `2695effb`
`stats`	APIEntitiesCommitStats	—
`status`	String	Example: `success`
`title`	String	Example: `Initial commit`

Property	Type	Description
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesCommitStats

Property	Type	Description
`additions`	Integer	Example: `1`
`deletions`	Integer	Example: `0`
`total`	Integer	Example: `1`

Suggestions

Use this API to manage code suggestions.

Suggestions provide a way to propose specific changes that can be directly applied to the code. You can programmatically create and apply code suggestions in merge request discussions with this API. Every API call to suggestions must be authenticated.

Apply multiple suggestions to a merge request

`PUT /api/v4/suggestions/batch_apply`

Applies multiple suggested patches in a merge request. You must have the Developer, Maintainer, or Owner role.

Request body (`application/json`)

Property	Type	Description
`commit_message`	String	A custom commit message to use instead of the default generated message or the project's default message
`ids` Required	Array of integers	An array of the suggestion IDs

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSuggestion
`400`	Bad Request	—

Apply a suggestion to a merge request

`PUT /api/v4/suggestions/{id}/apply`

Applies a suggested patch in a merge request. You must have the Developer, Maintainer, or Owner role.

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the suggestion

Request body (`application/json`)

Property	Type	Description
`commit_message`	String	A custom commit message to use instead of the default generated message or the project's default message

Responses

Code	Description	Schema
`200`	OK	APIEntitiesSuggestion
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesSuggestion

Property	Type	Description
`appliable`	Boolean	Example: `true`
`applied`	Boolean	Example: `false`
`from_content`	String	Example: `Original content`
`from_line`	Integer	Example: `1`
`id`	Integer (int64)	Example: `1`
`to_content`	String	Example: `New content`
`to_line`	Integer	Example: `1`

Tags

Use this API to manage Git tags. This API also returns X.509 signature information from signed tags.

List all project repository tags

`GET /api/v4/projects/{id}/repository/tags`

Lists all repository tags from a project, sorted by update date and time in descending order.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc` Default: `desc`
`order_by` Query	String	Sort results by the specified field. Sorting by version uses the semantic version number Allowed values: `name`, `updated`, `version` Default: `updated`
`search` Query	String	Return only tags matching the search criteria. Use `^term` to find tags that begin with `term`, and `term$` to find tags that end with `term`. No other regular expressions are supported
`page_token` Query	String	Name of the tag to start the pagination from. Used for keyset pagination
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesTag
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—
`422`	Unprocessable entity	—
`503`	Service unavailable	—

Create a tag

`POST /api/v4/projects/{id}/repository/tags`

Creates a tag in the repository that points to a specified reference.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project

Request body (`application/json`)

Property	Type	Description
`message`	String	Message for the annotated tag. If specified, creates an annotated tag Example: `Release 1.0.0`
`ref` Required	String	Commit SHA or name of the branch or tag to create the tag from Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`tag_name` Required	String	Name of the tag Example: `v.1.0.0`

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTag
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—
`429`	Too many requests	—

Retrieve a single repository tag

`GET /api/v4/projects/{id}/repository/tags/{tag_name}`

Retrieves a repository tag with a specified name. This endpoint can be accessed without authentication if the repository is publicly accessible.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Name of the tag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTag
`400`	Bad Request	—
`403`	Unauthenticated	—
`404`	Not found	—

Delete a tag

`DELETE /api/v4/projects/{id}/repository/tags/{tag_name}`

Deletes a specified repository tag.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Name of the tag

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad request	—
`403`	Unauthenticated	—
`404`	Not found	—
`412`	Precondition failed	—

Retrieve X.509 signature of a tag

`GET /api/v4/projects/{id}/repository/tags/{tag_name}/signature`

Retrieves the X.509 signature from a signed tag. Unsigned tags return a `404 Not Found` response.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`tag_name` Path, required	String	Name of the tag

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTagSignature
`400`	Bad Request	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCommit

Property	Type	Description
`author_email`	String	Example: `john@example.com`
`author_name`	String	Example: `John Smith`
`authored_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committed_date`	String (date-time)	Example: `2012-05-28T04:42:42-07:00`
`committer_email`	String	Example: `jack@example.com`
`committer_name`	String	Example: `Jack Smith`
`created_at`	String (date-time)	Example: `2017-07-26T11:08:53.000+02:00`
`extended_trailers`	Object	Example: `{"Signed-off-by":["John Doe \u003cjohndoe@gitlab.com\u003e","Jane Doe \u003cjanedoe@gitlab.com\u003e"]}`
`id`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`
`message`	String	Example: `Initial commit`
`parent_ids`	Array of strings	Example: `["2a4b78934375d7f53875269ffd4f45fd83a84ebe"]`
`short_id`	String	Example: `2695effb`
`title`	String	Example: `Initial commit`
`trailers`	Object	Example: `{"Merged-By":"Jane Doe janedoe@gitlab.com"}`
`web_url`	String	Example: `https://gitlab.example.com/janedoe/gitlab-`

Property	Type	Description
		`foss/-/commit/ed899a2f4b50b4370feeea94676502b42383c746`

APIEntitiesTag

Property	Type	Description
`commit`	APIEntitiesCommit	—
`created_at`	String (date-time)	Example: `2023-10-12T02:16:52.000Z`
`message`	String	Example: `Release v1.0.0`
`name`	String	Example: `v1.0.0`
`protected`	Boolean	Example: `true`
`release`	APIEntitiesTagRelease	—
`target`	String	Example: `2695effb5807a22ff3d138d593fd856244e155e7`

APIEntitiesTagRelease

Property	Type	Description
`description`	String	Example: `Amazing release. Wow`
`tag_name`	String	Example: `1.0.0`

APIEntitiesTagSignature

Property	Type	Description
`signature`	Object	—
`signature_type`	String	Example: `PGP`

Templates

Use this API to retrieve the file templates available to an entire instance:

- Dockerfile templates

- `.gitignore` templates
- CI/CD configuration templates
- Open source license templates

Custom CI/CD templates are not available from these endpoints. For templates specific to a project, use the project templates API.

Users with the Guest role can't access these templates. For more information, see project and group visibility.

List all Dockerfile templates

`GET /api/v4/templates/dockerfiles`

Lists all Dockerfile templates.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplatesList
`400`	Bad Request	—

Retrieve a Dockerfile template

`GET /api/v4/templates/dockerfiles/{name}`

Retrieves a specified Dockerfile template.

Parameters

Name	Type	Description
`name` Path, required	String	The name of the Dockerfile template

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplate
`400`	Bad Request	—
`404`	Not Found	—

List all .gitignore templates

`GET /api/v4/templates/gitignores`

Lists all .gitignore templates.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplatesList
`400`	Bad Request	—

Retrieve a .gitignore template

`GET /api/v4/templates/gitignores/{name}`

Retrieves a specified .gitignore template.

Parameters

Name	Type	Description
`name` Path, required	String	The name of the .gitignore template

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplate
`400`	Bad Request	—
`404`	Not Found	—

List all GitLab CI/CD YAML templates

`GET /api/v4/templates/gitlab_ci_ymls`

Lists all GitLab CI/CD YAML templates.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplatesList
`400`	Bad Request	—

Retrieve a GitLab CI/CD YAML template

`GET /api/v4/templates/gitlab_ci_ymls/{name}`

Retrieves a specified GitLab CI/CD YAML template.

Parameters

Name	Type	Description
`name` Path, required	String	The name of the GitLab CI/CD YAML template

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTemplate
`400`	Bad Request	—
`404`	Not Found	—

List all license templates

`GET /api/v4/templates/licenses`

Lists all available license templates.

Parameters

Name	Type	Description
`popular` Query	Boolean	If `true`, returns only popular licenses. If `false`, returns only licenses that aren't popular
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesLicense
`400`	Bad Request	—

Retrieve a license template

`GET /api/v4/templates/licenses/{name}`

Retrieves a specified license template. You can pass parameters to replace the license placeholder.

Parameters

Name	Type	Description
`name` Path, required	String	Name of the license template
`project` Query	String	Name of the copyrighted project
`fullname` Query	String	Full name of the copyright holder

Responses

Code	Description	Schema
`200`	OK	APIEntitiesLicense
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesLicense

Property	Type	Description
`conditions`	Array of strings	Example: `["include-copyright"]`
`content`	String	Example: `GNU GENERAL PUBLIC LICENSE`
`description`	String	Example: `A simple license`
`html_url`	String	Example: `http://choosealicense.com/licenses/gpl-3.0`
`key`	String	Example: `gpl-3.0`
`limitations`	Array of strings	Example: `["liability"]`
`name`	String	Example: `GNU General Public License v3.0`
`nickname`	String	Example: `GNU GPLv3`
`permissions`	Array of strings	Example: `["commercial-use"]`
`popular`	Boolean	—
`source_url`	String	—

APIEntitiesTemplate

Property	Type	Description
`content`	String	Example: `# Ruby gitignore template`
`name`	String	Example: `Ruby`

APIEntitiesTemplatesList

Property	Type	Description
`key`	String	Example: `mit`
`name`	String	Example: `MIT License`

Terraform

Use this API to interact with the Terraform CLI.

> This API is used by the Terraform CLI and is generally not meant for manual consumption. Undocumented authentication methods might be removed in the future.

Retrieve latest version for a module

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}`

Retrieves latest version for a specified module.

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTerraformModuleVersion
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Retrieve download URL for latest module version

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}/download`

Retrieves download URL for latest module version.

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider

Responses

Code	Description	Schema
`302`	Found	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

List all available versions for a module

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}/versions`

Lists all available versions for a specified module.

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesTerraformModuleVersions
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Get details about specific version of a module

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}/{module_version}`

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider
`module_version` Path, required	String	Version of the module

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTerraformModuleVersion
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Get download location for specific version of a module

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}/{module_version}/download`

Download specific version of a module

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider
`module_version` Path, required	String	Version of the module

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Download specific version of a module

`GET /api/v4/packages/terraform/modules/v1/{module_namespace}/{module_name}/{module_system}/{module_version}/file`

Parameters

Name	Type	Description
`module_namespace` Path, required	String	Group's ID or slug
`module_name` Path, required	String	Name of the module
`module_system` Path, required	String	Name of the module system, or provider
`module_version` Path, required	String	Version of the module

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Download the latest version of a module

`GET /api/v4/projects/{id}/packages/terraform/modules/{module_name}/{module_system}`

Downloads the latest version of a specified module. This feature was introduced in GitLab 16.7.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`module_name` Path, required	String	Module name Example: `infra-registry`
`module_system` Path, required	String	Name of the module system, or provider Example: `aws`
`terraform-get` Query	String	Terraform get redirection flag Allowed values: `1`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Download a specific version of a module

`GET /api/v4/projects/{id}/packages/terraform/modules/{module_name}/{module_system}/{module_version}`

This feature was introduced in GitLab 16.7

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`module_name` Path, required	String	Module name Example: `infra-registry`
`module_system` Path, required	String	Name of the module system, or provider Example: `aws`
`module_version` Path, required	String	Module version
`terraform-get` Query	String	Terraform get redirection flag Allowed values: `1`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Upload Terraform Module package file

`PUT /api/v4/projects/{id}/packages/terraform/modules/{module_name}/{module_system}/{module_version}/file`

This feature was introduced in GitLab 13.11

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`module_name` Path, required	String	Module name Example: `infra-registry`
`module_system` Path, required	String	Name of the module system, or provider Example: `aws`
`module_version` Path, required	String	Module version

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`201`	Created	—
`400`	Invalid file	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Workhorse authorize Terraform Module package file

`PUT /api/v4/projects/{id}/packages/terraform/modules/{module_name}/{module_system}/{module_version}/file/authorize`

This feature was introduced in GitLab 13.11

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`module_name` Path, required	String	Module name Example: `infra-registry`
`module_system` Path, required	String	Name of the module system, or provider Example: `aws`
`module_version` Path, required	String	Module version

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve a Terraform state

`GET /api/v4/projects/{id}/terraform/state/{name}`

Retrieves a Terraform state by name for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state
`ID` Query	String	Terraform state lock ID Maximum length: `255`

Responses

Code	Description	Schema
`200`	OK	—
`204`	Empty state	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Validation failure	—

Create or update a Terraform state

`POST /api/v4/projects/{id}/terraform/state/{name}`

Creates or updates a Terraform state for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state

Request body (`multipart/form-data`)

Property	Type	Description
`file` Required	String (binary)	The package file to be published (generated by Multipart middleware)

Responses

Code	Description	Schema
`200`	OK	—
`204`	No data provided	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not Found	—
`413`	Request Entity Too Large	—
`422`	Validation failure	—

Delete a Terraform state

`DELETE /api/v4/projects/{id}/terraform/state/{name}`

Deletes a Terraform state for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`422`	Validation failure	—

Authorize Terraform state upload

`POST /api/v4/projects/{id}/terraform/state/{name}/authorize`

Authorizes uploading a Terraform state for a specified project. This feature was introduced in GitLab 18.5.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Lock a Terraform state

`POST /api/v4/projects/{id}/terraform/state/{name}/lock`

Locks a Terraform state for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state

Request body (`application/json`)

Property	Type	Description
`Created` Required	String	Terraform state lock timestamp
`ID` Required	String	Terraform state lock ID Maximum length: `255`
`Info` Required	String	Terraform info
`Operation` Required	String	Terraform operation
`Path` Required	String	Terraform path
`Version` Required	String	Terraform version
`Who` Required	String	Terraform state lock owner

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`422`	Validation failure	—

Unlock a Terraform state

`DELETE /api/v4/projects/{id}/terraform/state/{name}/lock`

Unlocks a Terraform state for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state
`ID` Query	String	Terraform state lock ID Maximum length: `255`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—
`409`	Conflict	—
`422`	Validation failure	—

Retrieve a Terraform state version

`GET /api/v4/projects/{id}/terraform/state/{name}/versions/{serial}`

Retrieves a specified Terraform state version.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state
`serial` Path, required	Integer	Version number of the Terraform state

Responses

Code	Description	Schema
`200`	OK	String (binary) (`application/octet-stream`)
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Delete a Terraform state version

`DELETE /api/v4/projects/{id}/terraform/state/{name}/versions/{serial}`

Deletes a specified Terraform state version.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the project
`name` Path, required	String	Name of the Terraform state
`serial` Path, required	Integer	Version number of the Terraform state

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`403`	Forbidden	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesTerraformModuleVersion

Property	Type	Description
`name`	String	—
`provider`	String	—
`providers`	String	—
`root`	String	—
`source`	String	—
`submodules`	String	—
`version`	String	—
`versions`	String	—

APIEntitiesTerraformModuleVersions

Property	Type	Description
`modules`	String	—

To-dos

Use this API to list to-do items, mark them as done, and create a to-do item for an issue or a merge request.

Create a to-do item for an issuable

`POST /api/v4/projects/{id}/issues/{issue_iid}/todo`

Creates a to-do item for the current user on a specified issuable. If a to-do item already exists for the user on that issuable, returns a `304` status code.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`issue_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTodo
`400`	Bad Request	—
`404`	Not Found	—

Create a to-do item for an issuable

`POST /api/v4/projects/{id}/merge_requests/{merge_request_iid}/todo`

Creates a to-do item for the current user on a specified issuable. If a to-do item already exists for the user on that issuable, returns a `304` status code.

Parameters

Name	Type	Description
`id` Path, required	String	ID or URL-encoded path of the project
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTodo
`400`	Bad Request	—
`404`	Not Found	—

List all to-do items

`GET /api/v4/todos`

Lists all to-do items. When no filter is applied, it returns all pending to-do items for the current user. Different filters allow the user to refine the request.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`action` Query	String	Filter to-do items by action Allowed values: `assigned`, `review_requested`, `mentioned`, `build_failed`, `marked`, `approval_required`, `unmergeable`, `directly_addressed`, `member_access_requested`, `review_submitted`, `ssh_key_expired`, `ssh_key_expiring_soon`, `transfer_failed`, `merge_train_removed`
`author_id` Query	Integer	Filter to-do items by author ID
`project_id` Query	Integer	Filter to-do items by project ID
`group_id` Query	Integer	Filter to-do items by group ID
`state` Query	String	Filter to-do items by state Allowed values: `pending`, `done`
`type` Query	String	Filter to-do items by type Allowed values: `Commit`, `Issue`, `WorkItem`, `Merge Request`, `DesignManagement::Design`, `AlertManagement::Alert`,

Name	Type	Description
		`Namespace`, `Project`, `Key`, `WikiPage::Meta`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesTodo
`400`	Bad Request	—

Mark all to-do items as done

`POST /api/v4/todos/mark_as_done`

Marks all pending to-do items for the current user as done. It returns the HTTP status code `204` with an empty response.

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—

Mark a to-do item as done

`POST /api/v4/todos/{id}/mark_as_done`

Marks a to-do item for the current user as done.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the to-do item

Responses

Code	Description	Schema
`201`	Created	APIEntitiesTodo
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesNamespaceBasic

Property	Type	Description
`avatar_url`	String	Example: `https://example.com/avatar/12345`
`full_path`	String	Example: `group/my_project`
`id`	Integer (int64)	Example: `2`
`kind`	String	Example: `project`
`name`	String	Example: `project`
`parent_id`	Integer (int64)	Example: `1`
`path`	String	Example: `my_project`
`web_url`	String	Example: `https://example.com/group/my_project`

APIEntitiesProjectIdentity

Property	Type	Description
`created_at`	String (date-time)	Example: `2020-05-07T04:27:17.016Z`
`description`	String	Example: `desc`
`id`	Integer (int64)	Example: `1`
`name`	String	Example: `project1`
`name_with_namespace`	String	Example: `John Doe / project1`
`path`	String	Example: `project1`
`path_with_namespace`	String	Example: `namespace1/project1`

APIEntitiesTodo

Property	Type	Description
`action_name`	String	Example: `assigned`
`author`	APIEntitiesUserBasic	—
`body`	String	Example: `Task description`
`created_at`	String (date-time)	Example: `2016-06-17T07:52:35.225Z`
`group`	APIEntitiesNamespaceBasic	—
`id`	Integer (int64)	—
`project`	APIEntitiesProjectIdentity	—
`state`	String	Example: `pending`
`target`	Object	—
`target_type`	String	Example: `Issue`
`target_url`	String	Example: `http://example.com/foo/bar/-/issues/1`
`updated_at`	String (date-time)	Example: `2016-06-17T07:52:35.225Z`

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

Unleash

Use this API to serve feature flags to an Unleash-compatible client, and to register that client and receive its metrics. Generally these endpoints are used by an Unleash client and are not meant for manual consumption.

Get Unleash features

`GET /api/v4/feature_flags/unleash/{project_id}`

Parameters

Name	Type	Description
`project_id` Path, required	String	ID of the project
`instance_id` Query	String	Instance ID of Unleash Client
`app_name` Query	String	Application name of Unleash Client

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Get a list of features

`GET /api/v4/feature_flags/unleash/{project_id}/client/features`

Parameters

Name	Type	Description
`project_id` Path, required	String	ID of the project
`instance_id` Query	String	Instance ID of Unleash Client
`app_name` Query	String	Application name of Unleash Client

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Report Unleash client metrics

`POST /api/v4/feature_flags/unleash/{project_id}/client/metrics`

Parameters

Name	Type	Description
`project_id` Path, required	String	ID of the project

Request body (`application/json`)

Property	Type	Description
`app_name`	String	Application name of Unleash Client
`instance_id`	String	Instance ID of Unleash Client

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Register Unleash client

`POST /api/v4/feature_flags/unleash/{project_id}/client/register`

Parameters

Name	Type	Description
`project_id` Path, required	String	ID of the project

Request body (`application/json`)

Property	Type	Description
`app_name`	String	Application name of Unleash Client
`instance_id`	String	Instance ID of Unleash Client

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Get a list of features (v2 client support) (deprecated)

`GET /api/v4/feature_flags/unleash/{project_id}/features`

Deprecated in GitLab 15.6

Parameters

Name	Type	Description
`project_id` Path, required	String	ID of the project
`instance_id` Query	String	Instance ID of Unleash Client
`app_name` Query	String	Application name of Unleash Client

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Usage data

Use this API to track internal events and to retrieve the Service Ping payload that GitLab collects, including its non-SQL metrics and the queries behind it.

Track usage data event

`POST /api/v4/usage_data/increment_counter`

This feature was introduced in GitLab 13.4.

Request body (`application/json`)

Property	Type	Description
`event` Required	String	Event name that should be tracked Example: `i_quickactions_page`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Track usage data event for the current user

`POST /api/v4/usage_data/increment_unique_users`

Request body (`application/json`)

Property	Type	Description
`event` Required	String	Event name that should be tracked Example: `i_quickactions_page`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

List all non-SQL metrics

`GET /api/v4/usage_data/non_sql_metrics`

Lists all non-SQL metrics data used in the Service ping. Administrators only.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

List all Service Ping SQL queries

`GET /api/v4/usage_data/queries`

Lists all raw SQL queries used to compute Service Ping. Administrators only.

Responses

Code	Description	Schema
`200`	OK	—
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Retrieve Service Ping payload

`GET /api/v4/usage_data/service_ping`

Retrieves the Service Ping payload from the application cache as JSON. If no cached payload is available, returns an empty response. Requires a personal access token with the `read_service_ping` scope. Introduced in GitLab 16.9.

Responses

Code	Description	Schema
`200`	OK	—
`401`	401 Unauthorized	—
`403`	Forbidden	—
`404`	Not found	—

Track an internal GitLab event

`POST /api/v4/usage_data/track_event`

Tracks a GitLab internal event. This action increments Service Ping counters in Redis and is optionally sent to Snowplow. Introduced in GitLab 16.2.

Request body (`application/json`)

Property	Type	Description
`additional_properties`	Object	Additional properties to be tracked Example: `{"label":"login_button","value":1}`
`event` Required	String	Name of the event to track Example: `i_quickactions_page`
`namespace_id`	Integer	ID of the namespace associated with the event Example: `1234`
`project_id`	Integer	ID of the project. Mutually exclusive with `project_path` Example: `1234`
`project_path`	String	Path of the project. Used to find `project_id` when it is not provided. Mutually exclusive with `project_id` Example: `namespace/project`
`send_to_snowplow`	Boolean	If `true`, sends the tracked event to Snowplow Example: `true`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	Unauthorized	—
`404`	Not found	—

Track multiple internal GitLab events

`POST /api/v4/usage_data/track_events`

Tracks one or more GitLab internal events in a single request. Each event increments Service Ping counters in Redis and is optionally sent to Snowplow. This feature was introduced in GitLab 17.3.

Request body (`application/json`)

Property	Type	Description
`events` Required	Array of objects	Array of internal events to track. Maximum of 50 events

Responses

Code	Description	Schema
`200`	OK	—
`400`	Validation error	—
`401`	Unauthorized	—

Users

Use this API to interact with user accounts on GitLab. These endpoints can help manage your account, manage accounts for other users, moderate user accounts, perform follower actions, and interact with email addresses for user accounts.

Retrieve current user details

`GET /api/v4/user`

Retrieves details about your user account.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserPublic
`401`	Unauthorized	—
`403`	Forbidden	—

List all activity for a user

`GET /api/v4/user/activities`

Lists all activity for a specified user.

Parameters

Name	Type	Description
`from` Query	String (date-time)	Date string in the format YEAR-MONTH-DAY Default: `2024-08-14T17:26:19.883Z` Example: `2024-08-14T17:26:19.883Z`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserActivity
`400`	Bad Request	—

List all email addresses for a user

`GET /api/v4/user/emails`

Lists all email addresses for the currently authenticated user.

Parameters

Name	Type	Description
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesEmail
`400`	Bad Request	—

Add an email address

`POST /api/v4/user/emails`

Adds an email address for the currently authenticated user.

Request body (`application/json`)

Property	Type	Description
`email` Required	String	Email address to add to the account

Responses

Code	Description	Schema
`201`	Created	APIEntitiesEmail
`400`	Bad Request	—

Retrieve details on an email address

`GET /api/v4/user/emails/{email_id}`

Retrieves details on a specified email address for the currently authenticated user.

Parameters

Name	Type	Description
`email_id` Path, required	Integer	ID of the email

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEmail
`400`	Bad Request	—
`404`	Not Found	—

Delete an email address

`DELETE /api/v4/user/emails/{email_id}`

Deletes the specified email address. You cannot delete a primary email address. Any future emails sent to the deleted email address are sent to the primary email address instead.

Parameters

Name	Type	Description
`email_id` Path, required	Integer	ID of the email

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve your user preferences

`GET /api/v4/user/preferences`

Retrieves your user preferences.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserPreferences
`401`	Unauthorized	—
`403`	Forbidden	—

Update your user preferences

`PUT /api/v4/user/preferences`

Updates your user preferences.

Request body (`application/json`)

Property	Type	Description
`pass_user_identities_to_ci_jwt`	Boolean	If `true`, passes the user's external identities to CI jobs as part of a JSON web token. This attribute is internal to GitLab and must not be passed to third-party services. For more information, see Token Payload. At least one of `view_diffs_file_by_file`, `show_whitespace_in_diffs`, `pass_user_identities_to_ci_jwt`, `policy_advanced_editor` must be provided
`policy_advanced_editor`	Boolean	If `true`, enables the advanced editor. At least one of `view_diffs_file_by_file`, `show_whitespace_in_diffs`, `pass_user_identities_to_ci_jwt`, `policy_advanced_editor` must be provided
`show_whitespace_in_diffs`	Boolean	If `true`, the user sees whitespace changes in diffs. At least one of `view_diffs_file_by_file`, `show_whitespace_in_diffs`, `pass_user_identities_to_ci_jwt`, `policy_advanced_editor` must be provided
`view_diffs_file_by_file`	Boolean	If `true`, the user sees only one file diff per page. At least one of `view_diffs_file_by_file`, `show_whitespace_in_diffs`, `pass_user_identities_to_ci_jwt`, `policy_advanced_editor` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserPreferences
`400`	Bad Request	—

Create a runner owned by currently authenticated user

`POST /api/v4/user/runners`

Create a new runner

Request body (`application/json`)

Property	Type	Description
`access_level`	String	Access level of the runner Allowed values: `not_protected`, `ref_protected`
`description`	String	Description of the runner
`group_id` Required	Integer	ID of the group to create the runner in. Required if `runner_type` is `group_type` Example: `1`
`locked`	Boolean	Specifies if the runner should be locked for the current project Default: `false`
`maintenance_note`	String	Free-form maintenance notes for the runner. Limited to 1024 characters
`maximum_timeout`	Integer	Maximum time, in seconds, that runners can spend running a job
`paused`	Boolean	If `true`, the runner ignores new jobs
`project_id` Required	Integer	ID of the project to create the runner in. Required if `runner_type` is `project_type` Example: `1`
`run_untagged`	Boolean	Specifies if the runner should handle untagged jobs Default: `true`
`runner_type` Required	String	Scope of the runner Allowed values: `instance_type`, `group_type`, `project_type` Minimum length: `1`
`tag_list`	Array of strings	

Property	Type	Description
		Comma-separated list of runner tags

Responses

Code	Description	Schema
`201`	Created	APIEntitiesCiRunnerRegistrationDetails
`400`	Bad Request	—
`403`	Forbidden	—

Retrieve your user status

`GET /api/v4/user/status`

Retrieves your user status.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserStatus
`401`	Unauthorized	—
`403`	Forbidden	—

Set a user status

`PUT /api/v4/user/status`

Sets the status of the currently authenticated user. Any parameters that are not passed are nullified. This operation is similar but distinct from the Update a user status operation.

Request body (`application/json`)

Property	Type	Description
`availability`	String	Availability of the user. Possible values are `busy` and `not_set`
`clear_status_after`	String	Time after which the status is cleared automatically Allowed values: `30_minutes`, `3_hours`, `8_hours`, `1_day`, `3_days`, `7_days`, `30_days`
`emoji`	String	Name of the emoji to use as a status. If omitted, `speech_balloon` is used. Must be one of the names in the Gemojione index
`message`	String	Message to set as a status. Can contain emoji codes. Maximum of 100 characters

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserStatus
`400`	Bad Request	—

Update a user status

`PATCH /api/v4/user/status`

Sets the status of the currently authenticated user. Any parameters that are not passed are ignored. This operation is similar but distinct from the Set a user status operation.

Request body (`application/json`)

Property	Type	Description
`availability`	String	Availability of the user. Possible values are `busy` and `not_set`
`clear_status_after`	String	Time after which the status is cleared automatically Allowed values: `30_minutes`, `3_hours`, `8_hours`, `1_day`, `3_days`, `7_days`, `30_days`
`emoji`	String	Name of the emoji to use as a status. If omitted, `speech_balloon` is used. Must be one of the names in the Gemojione index
`message`	String	Message to set as a status. Can contain emoji codes. Maximum of 100 characters

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserStatus
`400`	Bad Request	—

Retrieve Support PIN

`GET /api/v4/user/support_pin`

Retrieves a Support PIN for the currently authenticated user. GitLab Support may ask for this PIN to validate your identity.

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserSupportPin
`401`	Unauthorized	—
`403`	Forbidden	—
`404`	Not Found	—

Create a Support PIN

`POST /api/v4/user/support_pin`

Creates a Support PIN. The PIN expires seven days after creation. GitLab Support may ask for this PIN to validate your identity.

Responses

Code	Description	Schema
`201`	Created	APIEntitiesUserSupportPin
`400`	Bad Request	—

Update a user’s credit_card_validation (deprecated)

`PUT /api/v4/user/{user_id}/credit_card_validation`

Deprecated in 17.7

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user

Request body (`application/json`)

Property	Type	Description
`credit_card_expiration_month` Required	Integer	Month the credit card expires
`credit_card_expiration_year` Required	Integer	Year the credit card expires
`credit_card_holder_name` Required	String	Name of the credit card holder
`credit_card_mask_number` Required	String	Last 4 digits of the credit card number
`credit_card_type` Required	String	Name of the credit card network
`credit_card_validated_at` Required	String (date-time)	Time when the user's credit card was validated
`stripe_card_fingerprint`	String	Stripe credit card fingerprint
`stripe_payment_method_xid`	String	ID of the Stripe payment method
`stripe_setup_intent_xid`	String	ID of the Stripe setup intent
`zuora_payment_method_xid`	String	ID of the Zuora payment method

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserCreditCardValidations
`400`	Bad Request	—
`404`	Not Found	—

Return the user specific counts

`GET /api/v4/user_counts`

Assigned open issues, assigned MRs and pending todos count

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserCounts
`401`	Unauthorized	—

List all users

`GET /api/v4/users`

Lists all users on the instance. This endpoint supports keyset pagination. In GitLab 17.0 and later, keyset pagination is used by default.

Parameters

Name	Type	Description
`username` Query	String	Return a single user with the specified username
`extern_uid` Query	String	Return a single user with the specified external authentication provider UID. Use with `provider`. Administrators only. All or none of `extern_uid`, `provider` must be provided
`public_email` Query	String	Return a single user with the specified public email
`provider` Query	String	Name of the external authentication provider. All or none of `extern_uid`, `provider` must be provided
`search` Query	String	Search for users by name, username, or public email
`active` Query	Boolean	If `true`, returns only active users Default: `false`
`humans` Query	Boolean	If `true`, returns only regular users that are not bot or internal users Default: `false`
`external` Query	Boolean	If `true`, returns only external users Default: `false`
`blocked` Query	Boolean	If `true`, returns only blocked users Default: `false`
`created_after` Query	String (date-time)	Return users created on or after the specified time
`created_before` Query	String (date-time)	Return users created on or before the specified time

Name	Type	Description
`without_projects` Query	Boolean	If `true`, returns only users that do not have any projects Default: `false`
`without_project_bots` Query	Boolean	If `true`, excludes project bot users from the results Default: `false`
`admins` Query	Boolean	If `true`, returns only administrators Default: `false`
`two_factor` Query	String	Filter users by Two-factor authentication
`exclude_active` Query	Boolean	If `true`, returns only users that are not active Default: `false`
`exclude_external` Query	Boolean	If `true`, returns only users that are not external Default: `false`
`exclude_humans` Query	Boolean	If `true`, returns only bot or internal users Default: `false`
`exclude_internal` Query	Boolean	If `true`, returns only users that are not internal Default: `false`
`order_by` Query	String	Sort results by the specified field Allowed values: `id`, `name`, `username`, `created_at`, `updated_at`
`sort` Query	String	Sort results in ascending or descending order Allowed values: `asc`, `desc`
`page` Query	Integer	Page number of results to retrieve

Name	Type	Description
		Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—

Create a user

`POST /api/v4/users`

Creates a user. Administrators only.

Request body (`multipart/form-data`)

Property	Type	Description
`admin`	Boolean	Flag indicating the user is an administrator
`avatar`	String (binary)	Avatar image for user
`bio`	String	The biography of the user
`can_create_group`	Boolean	Flag indicating the user can create groups
`color_scheme_id`	Integer	The color scheme for the file viewer
`commit_email`	String	The commit email, _private for private commit email
`discord`	String	The Discord user ID
`email` Required	String	Email address to add to the user account
`extern_uid`	String	The external authentication provider UID. All or none of `extern_uid`, `provider` must be provided
`external`	Boolean	Flag indicating the user is an external user
`force_random_password`	Boolean	Flag indicating a random password will be set. At least one of `password`, `reset_password`, `force_random_password` must be provided
`github`	String	The GitHub username
`linkedin`	String	The LinkedIn username
`location`	String	The location of the user
`name` Required	String	The name of the user
`note`	String	Admin note for this user

Property	Type	Description
`organization`	String	The organization of the user. Empty string or nil clears the field
`password`	String	The password of the new user. At least one of `password`, `reset_password`, `force_random_password` must be provided
`policy_advanced_editor`	Boolean	Flag indicating that advanced editor is enabled
`private_profile`	Boolean	Flag indicating the user has a private profile
`projects_limit`	Integer	The number of projects a user can create
`pronouns`	String	The pronouns of the user
`provider`	String	Name of the external authentication provider. All or none of `extern_uid`, `provider` must be provided
`public_email`	String	The public email of the user
`reset_password`	Boolean	Flag indicating the user will be sent a password reset token. At least one of `password`, `reset_password`, `force_random_password` must be provided
`skip_confirmation`	Boolean	Flag indicating the account is confirmed
`theme_id`	Integer	The GitLab theme for the user
`twitter`	String	The Twitter username
`username` Required	String	The username of the user

Property	Type	Description
`view_diffs_file_by_file`	Boolean	If `true`, the user sees only one file diff per page
`website_url`	String	The website of the user

Responses

Code	Description	Schema
`201`	Created	APIEntitiesUserWithAdmin
`400`	Bad Request	—

Retrieve a user as a regular user

`GET /api/v4/users/{id}`

Retrieves a specified user as a regular user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`with_custom_attributes` Query	Boolean	If `true`, includes custom attributes in the response. Administrators only Default: `false`
`custom_attributes` Query	Object	Filter by custom attributes that match all of the specified key and value pairs. Administrators only

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserProfile
`400`	Bad Request	—
`404`	Not Found	—

Update a user

`PUT /api/v4/users/{id}`

Updates the details of a specified user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Request body (`multipart/form-data`)

Property	Type	Description
`admin`	Boolean	Flag indicating the user is an administrator
`avatar`	String (binary)	Avatar image for user
`bio`	String	The biography of the user
`can_create_group`	Boolean	Flag indicating the user can create groups
`color_scheme_id`	Integer	The color scheme for the file viewer
`commit_email`	String	The commit email, _private for private commit email
`discord`	String	The Discord user ID
`email`	String	Email address to add to the user account
`extern_uid`	String	The external authentication provider UID. All or none of `extern_uid`, `provider` must be provided
`external`	Boolean	Flag indicating the user is an external user
`github`	String	The GitHub username
`linkedin`	String	The LinkedIn username
`location`	String	The location of the user
`name`	String	The name of the user
`note`	String	Admin note for this user
`organization`	String	The organization of the user. Empty string or nil clears the field
`password`	String	The password of the new user
`policy_advanced_editor`	Boolean	Flag indicating that advanced editor is enabled

Property	Type	Description
`private_profile`	Boolean	Flag indicating the user has a private profile
`projects_limit`	Integer	The number of projects a user can create
`pronouns`	String	The pronouns of the user
`provider`	String	Name of the external authentication provider. All or none of `extern_uid`, `provider` must be provided
`public_email`	String	The public email of the user
`skip_reconfirmation`	Boolean	Flag indicating the account skips the confirmation by email
`theme_id`	Integer	The GitLab theme for the user
`twitter`	String	The Twitter username
`username`	String	The username of the user
`view_diffs_file_by_file`	Boolean	If `true`, the user sees only one file diff per page
`website_url`	String	The website of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserWithAdmin
`400`	Bad Request	—
`404`	Not Found	—

Delete a user

`DELETE /api/v4/users/{id}`

Deletes a user. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`hard_delete` Query	Boolean	If `true`, contributions that would usually be moved to a ghost user are deleted instead, along with groups owned solely by this user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEmail
`400`	Bad Request	—
`404`	Not Found	—

Reactivate a user

`POST /api/v4/users/{id}/activate`

Reactivates a specified user account that was previously deactivated. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Approve access to a user

`POST /api/v4/users/{id}/approve`

Approves access to a specified user account that is pending approval. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve association counts for a user

`GET /api/v4/users/{id}/associations_count`

Retrieves the number of groups, projects, issues, and merge requests associated with a specified user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserAssociationsCount
`400`	Bad Request	—
`404`	Not Found	—

Ban a user

`POST /api/v4/users/{id}/ban`

Bans a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Block access to a user

`POST /api/v4/users/{id}/block`

Blocks a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Deactivate a user

`POST /api/v4/users/{id}/deactivate`

Deactivates a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Disable two-factor authentication for a user

`PATCH /api/v4/users/{id}/disable_two_factor`

Disables two-factor authentication for a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserWithAdmin
`400`	Bad Request	—
`404`	Not Found	—

List all email addresses for a user

`GET /api/v4/users/{id}/emails`

Lists all email addresses for a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesEmail
`400`	Bad Request	—
`404`	Not Found	—

Add an email address for a user

`POST /api/v4/users/{id}/emails`

Adds an email address for a specified user account. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Request body (`application/json`)

Property	Type	Description
`email` Required	String	Email address to add to the user account
`skip_confirmation`	Boolean	If `true`, skips email confirmation and assumes the email is verified

Responses

Code	Description	Schema
`201`	Created	APIEntitiesEmail
`400`	Bad Request	—
`404`	Not Found	—

Delete an email address for a user

`DELETE /api/v4/users/{id}/emails/{email_id}`

Deletes a specified email address for a user account. You cannot delete a primary email address. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`email_id` Path, required	Integer	ID of the email

Responses

Code	Description	Schema
`200`	OK	APIEntitiesEmail
`400`	Bad Request	—
`404`	Not Found	—

Follow a user

`POST /api/v4/users/{id}/follow`

Follows a specified user account.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	APIEntitiesUser
`400`	Bad Request	—
`404`	Not Found	—

List all accounts that follow a user

`GET /api/v4/users/{id}/followers`

Lists all user accounts that follow a specified user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`404`	Not Found	—

List all accounts followed by a user

`GET /api/v4/users/{id}/following`

Lists all user accounts being followed by a specified user.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesUserBasic
`400`	Bad Request	—
`404`	Not Found	—

Delete authentication identity from a user

`DELETE /api/v4/users/{id}/identities/{provider}`

Deletes a user authentication identity using the provider name associated with that identity. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user
`provider` Path, required	String	Name of the external authentication provider

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserWithAdmin
`400`	Bad Request	—
`404`	Not Found	—

Reject access to a user

`POST /api/v4/users/{id}/reject`

Rejects access to a specified user account that is pending approval. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Retrieve a Support PIN for a user

`GET /api/v4/users/{id}/support_pin`

Retrieves a Support PIN for a specified user. GitLab Support may ask for this PIN to validate your identity. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserSupportPin
`400`	Bad Request	—
`404`	Not Found	—

Revoke a Support PIN for a user

`POST /api/v4/users/{id}/support_pin/revoke`

Revokes a Support PIN for a specified user. This immediately expires and removes the PIN. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Unban a user

`POST /api/v4/users/{id}/unban`

Unbans a specified user account that was previously banned. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Unblock access to a user

`POST /api/v4/users/{id}/unblock`

Unblocks a specified user account that was previously blocked. Administrators only.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Unfollow a user

`POST /api/v4/users/{id}/unfollow`

Unfollows a specified user account.

Parameters

Name	Type	Description
`id` Path, required	Integer	ID of the user

Responses

Code	Description	Schema
`201`	Created	APIEntitiesUser
`400`	Bad Request	—
`404`	Not Found	—

List all project and group memberships for a user

`GET /api/v4/users/{user_id}/memberships`

Lists all project and group memberships for a specified user. Administrators only.

Parameters

Name	Type	Description
`user_id` Path, required	Integer	ID of the user
`type` Query	String	Filter memberships by type Allowed values: `Project`, `Namespace`
`page` Query	Integer	Page number of results to retrieve Default: `1` Example: `1`
`per_page` Query	Integer	Number of items to list per page Default: `20` Example: `20`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntities Membership
`400`	Bad Request	—
`404`	Not Found	—

Retrieve the status of a user

`GET /api/v4/users/{user_id}/status`

Retrieves the status of a user. You can access this endpoint without authentication.

Parameters

Name	Type	Description
`user_id` Path, required	String	ID or username of the user

Responses

Code	Description	Schema
`200`	OK	APIEntitiesUserStatus
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesCiRunnerRegistrationDetails

Property	Type	Description
`id`	String	—
`token`	String	—
`token_expires_at`	String	—

APIEntitiesCustomAttribute

Property	Type	Description
`key`	String	Example: `foo`
`value`	String	Example: `bar`

APIEntitiesEmail

Property	Type	Description
`confirmed_at`	String	—
`email`	String	—
`id`	String	—

APIEntitiesIdentity

Property	Type	Description
`extern_uid`	String	—
`provider`	String	—

APIEntitiesMembership

Property	Type	Description
`access_level`	String	—
`source_id`	String	—
`source_name`	String	—
`source_type`	String	—

APIEntitiesUser

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`

Property	Type	Description
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserActivity

Property	Type	Description
`last_activity_at`	String	—
`last_activity_on`	String	—
`username`	String	—

APIEntitiesUserAssociationsCount

Property	Type	Description
`groups_count`	String	—
`issues_count`	String	—
`merge_requests_count`	String	—
`projects_count`	String	—

APIEntitiesUserBasic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`id`	Integer (int64)	Example: `1`
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`

APIEntitiesUserCounts

Property	Type	Description
`assigned_issues`	Integer	Example: `10`
`assigned_merge_requests`	Integer	Example: `10`
`merge_requests`	Integer	Example: `10`
`review_requested_merge_requests`	Integer	Example: `10`
`todos`	Integer	Example: `10`

APIEntitiesUserCreditCardValidations

Property	Type	Description
`credit_card_validated_at`	String	—
`user_id`	String	—

APIEntitiesUserPreferences

Property	Type	Description
`id`	String	—
`pass_user_identities_to_ci_jwt`	String	—
`policy_advanced_editor`	String	—
`show_whitespace_in_diffs`	String	—
`user_id`	String	—
`view_diffs_file_by_file`	String	—

APIEntitiesUserProfile

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bio_html`	String	Example: `My <em>bio</em>.`
`bot`	Boolean	—
`created_at`	String	—
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`is_followed`	String	—
`job_title`	String	—
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`

Property	Type	Description
`state`	String	Example: `active`
`twitter`	String	—
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserPublic

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`can_create_group`	Boolean	Example: `true`
`can_create_project`	Boolean	Example: `true`
`color_scheme_id`	Integer	Example: `1`
`commit_email`	String	—
`confirmed_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`created_at`	String	—
`current_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`email`	String	Example: `john@example.com`
`external`	Boolean	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`identities`	APIEntitiesIdentity	—
`is_followed`	String	—

Property	Type	Description
`job_title`	String	—
`last_activity_on`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`last_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`organization`	String	—
`preferred_language`	String	Example: `en`
`private_profile`	Boolean	Example: `false`
`projects_limit`	Integer	Example: `10`
`pronouns`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`theme_id`	Integer	Example: `2`
`twitter`	String	—
`two_factor_enabled`	Boolean	Example: `true`
`username`	String	Example: `admin`
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

APIEntitiesUserStatus

Property	Type	Description
`availability`	String	—
`clear_status_at`	String	—
`emoji`	String	—
`message`	String	—
`message_html`	String	—

APIEntitiesUserSupportPin

Property	Type	Description
`expires_at`	String (date-time)	The expiration time of the PIN
`pin`	String	The security PIN

APIEntitiesUserWithAdmin

Property	Type	Description
`avatar_path`	String	Example: `/user/avatar/28/The-Big-Lebowski-400-400.png`
`avatar_url`	String	Example: `https://gravatar.com/avatar/1`
`bio`	String	—
`bot`	Boolean	—
`can_create_group`	Boolean	Example: `true`
`can_create_project`	Boolean	Example: `true`
`color_scheme_id`	Integer	Example: `1`
`commit_email`	String	—
`confirmed_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`created_at`	String	—
`created_by`	APIEntitiesUserBasic	—
`current_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`custom_attributes`	Array of APIEntitiesCustomAttribute	—
`discord`	String	—
`email`	String	Example: `john@example.com`
`external`	Boolean	—
`followers`	String	—
`following`	String	—
`github`	String	—
`id`	Integer (int64)	Example: `1`
`identities`	APIEntitiesIdentity	—

Property	Type	Description
`is_admin`	String	—
`is_followed`	String	—
`job_title`	String	—
`last_activity_on`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`last_sign_in_at`	String (date-time)	Example: `2015-09-03T07:24:01.670Z`
`linkedin`	String	—
`local_time`	String	—
`location`	String	—
`locked`	Boolean	—
`name`	String	Example: `Administrator`
`namespace_id`	String	—
`note`	String	—
`organization`	String	—
`preferred_language`	String	Example: `en`
`private_profile`	Boolean	Example: `false`
`projects_limit`	Integer	Example: `10`
`pronouns`	String	—
`provisioned_by_project_id`	String	—
`public_email`	String	Example: `john@example.com`
`state`	String	Example: `active`
`theme_id`	Integer	Example: `2`
`twitter`	String	—
`two_factor_enabled`	Boolean	Example: `true`
`username`	String	Example: `admin`

Property	Type	Description
`web_url`	String	Example: `https://gitlab.example.com/root`
`website_url`	String	—
`work_information`	String	—

VSCode

Use this API to synchronize Visual Studio Code settings, so that a user's editor preferences follow them between VS Code and the Web IDE.

Deletes all user vscode setting resources

`DELETE /api/v4/vscode/settings_sync/v1/collection`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get the settings manifest for Settings Sync

`GET /api/v4/vscode/settings_sync/v1/manifest`

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeManifest
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get a list of references to one or more vscode setting resources

`GET /api/v4/vscode/settings_sync/v1/resource/{resource_name}`

Parameters

Name	Type	Description
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeSettingReference
`400`	400 bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Creates or updates a specific setting

`POST /api/v4/vscode/settings_sync/v1/resource/{resource_name}`

Parameters

Name	Type	Description
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get a specific setting resource

`GET /api/v4/vscode/settings_sync/v1/resource/{resource_name}/{id}`

Parameters

Name	Type	Description
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`
`id` Path, required	String	ID of the resource to retrieve

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeSetting
`204`	No content	—
`400`	400 bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Deletes all user vscode setting resources

`DELETE /api/v4/vscode/settings_sync/{settings_context_hash}/v1/collection`

Parameters

Name	Type	Description
`settings_context_hash` Path, required	String	Settings context hash

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get the settings manifest for Settings Sync

`GET /api/v4/vscode/settings_sync/{settings_context_hash}/v1/manifest`

Parameters

Name	Type	Description
`settings_context_hash` Path, required	String	Settings context hash

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeManifest
`400`	Bad Request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get a list of references to one or more vscode setting resources

`GET /api/v4/vscode/settings_sync/{settings_context_hash}/v1/resource/{resource_name}`

Parameters

Name	Type	Description
`settings_context_hash` Path, required	String	Settings context hash
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeSettingReference
`400`	400 bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Creates or updates a specific setting

`POST /api/v4/vscode/settings_sync/{settings_context_hash}/v1/resource/{resource_name}`

Parameters

Name	Type	Description
`settings_context_hash` Path, required	String	Settings context hash
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Get a specific setting resource

`GET /api/v4/vscode/settings_sync/{settings_context_hash}/v1/resource/{resource_name}/{id}`

Parameters

Name	Type	Description
`settings_context_hash` Path, required	String	Settings context hash
`resource_name` Path, required	String	Name of the resource such as settings Allowed values: `settings`, `extensions`, `globalState`, `machines`, `keybindings`, `snippets`, `tasks`, `profiles` Minimum length: `1`
`id` Path, required	String	ID of the resource to retrieve

Responses

Code	Description	Schema
`200`	OK	APIVsCodeSettingsEntitiesVsCodeSetting
`204`	No content	—
`400`	400 bad request	—
`401`	401 Unauthorized	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIVsCodeSettingsEntitiesVsCodeManifest

Property	Type	Description
`latest`	String	—
`session`	String	Example: `1`

APIVsCodeSettingsEntitiesVsCodeSetting

Property	Type	Description
`content`	String	—
`machineId`	String	—
`machines`	String	—
`version`	String	—

APIVsCodeSettingsEntitiesVsCodeSettingReference

Property	Type	Description
`created`	String	—
`url`	String	—

Web commits

Use this API to retrieve the public key that GitLab uses to sign the commits created with the web editor.

Retrieve the public signing key

`GET /api/v4/web_commits/public_key`

Retrieves the GitLab public key for signing web commits. This feature was introduced in GitLab 17.4.

Responses

Code	Description	Schema
`200`	OK	—
`404`	Public key not found.	—
`503`	The git server, Gitaly, is not available at this time. Please contact your administrator.	—

Wikis

Use this API to manage group wikis. An API for project wikis is also available.

Comments on a wiki page are called `notes`. To interact with them, use the notes API.

List all wiki pages for a group

`GET /api/v4/groups/{id}/wikis`

Lists all wiki pages for a specified group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`with_content` Query	Boolean	If `true`, includes the content of each wiki page Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesWikiPage Basic
`400`	Bad Request	—
`404`	Not found	—

Create a wiki page for a group

`POST /api/v4/groups/{id}/wikis`

Creates a wiki page for a specified group. Requests can define the title, slug, and content.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project

Request body (`application/json`)

Property	Type	Description
`content` Required	String	Content of the wiki page
`format`	String	Format of the wiki page Allowed values: `markdown` , `rdoc` , `asciidoc` , `org` Default: `markdown`
`front_matter`	Object	YAML front matter of the wiki page
`front_matter.title`	String	Title in the front matter of the wiki page
`title` Required	String	Title of the wiki page

Responses

Code	Description	Schema
`201`	Created	APIEntitiesWikiPage
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Upload an attachment to a group wiki

`POST /api/v4/groups/{id}/wikis/attachments`

Uploads a file to the `uploads` directory in a specified group wiki.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project

Request body (`multipart/form-data`)

Property	Type	Description
`branch`	String	The name of the branch
`file` Required	String (binary)	The attachment file to be uploaded

Responses

Code	Description	Schema
`201`	Created	APIEntitiesWikiAttachment
`400`	Bad Request	—
`404`	Not found	—

Retrieve a wiki page for a group

`GET /api/v4/groups/{id}/wikis/{slug}`

Retrieves a specified wiki page for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`
`version` Query	String	Version SHA of the wiki page
`render_html` Query	Boolean	If `true` , returns the rendered HTML of the wiki page Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesWikiPage
`400`	Bad Request	—
`404`	Not found	—

Update a wiki page for a group

`PUT /api/v4/groups/{id}/wikis/{slug}`

Updates a specified wiki page for a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`

Request body (`application/json`)

Property	Type	Description
`content`	String	Content of the wiki page. At least one of `content`, `title`, `format` must be provided
`format`	String	Format of the wiki page. At least one of `content`, `title`, `format` must be provided Allowed values: `markdown`, `rdoc`, `asciidoc`, `org` Default: `markdown`
`front_matter`	Object	YAML front matter of the wiki page
`front_matter.title`	String	Title in the front matter of the wiki page
`title`	String	Title of the wiki page. At least one of `content`, `title`, `format` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesWikiPage
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a wiki page for a group

`DELETE /api/v4/groups/{id}/wikis/{slug}`

Deletes a specified wiki page from a group.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Validation error	—
`404`	Not found	—

List all wiki pages for a project

`GET /api/v4/projects/{id}/wikis`

Lists all wiki pages for a specified project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`with_content` Query	Boolean	If `true`, includes the content of each wiki page Default: `false`

Responses

Code	Description	Schema
`200`	OK	Array of APIEntitiesWikiPage Basic
`400`	Bad Request	—
`404`	Not found	—

Create a wiki page for a project

`POST /api/v4/projects/{id}/wikis`

Creates a wiki page for a specified project. Requests can define the title, slug, and content.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project

Request body (`application/json`)

Property	Type	Description
`content` Required	String	Content of the wiki page
`format`	String	Format of the wiki page Allowed values: `markdown` , `rdoc` , `asciidoc` , `org` Default: `markdown`
`front_matter`	Object	YAML front matter of the wiki page
`front_matter.title`	String	Title in the front matter of the wiki page
`title` Required	String	Title of the wiki page

Responses

Code	Description	Schema
`201`	Created	APIEntitiesWikiPage
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Upload an attachment to a project wiki

`POST /api/v4/projects/{id}/wikis/attachments`

Uploads a file to the `uploads` directory in a specified project wiki.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project

Request body (`multipart/form-data`)

Property	Type	Description
`branch`	String	The name of the branch
`file` Required	String (binary)	The attachment file to be uploaded

Responses

Code	Description	Schema
`201`	Created	APIEntitiesWikiAttachment
`400`	Bad Request	—
`404`	Not found	—

Retrieve a wiki page for a project

`GET /api/v4/projects/{id}/wikis/{slug}`

Retrieves a specified wiki page for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`
`version` Query	String	Version SHA of the wiki page
`render_html` Query	Boolean	If `true`, returns the rendered HTML of the wiki page Default: `false`

Responses

Code	Description	Schema
`200`	OK	APIEntitiesWikiPage
`400`	Bad Request	—
`404`	Not found	—

Update a wiki page for a project

`PUT /api/v4/projects/{id}/wikis/{slug}`

Updates a specified wiki page for a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`

Request body (`application/json`)

Property	Type	Description
`content`	String	Content of the wiki page. At least one of `content`, `title`, `format` must be provided
`format`	String	Format of the wiki page. At least one of `content`, `title`, `format` must be provided Allowed values: `markdown`, `rdoc`, `asciidoc`, `org` Default: `markdown`
`front_matter`	Object	YAML front matter of the wiki page
`front_matter.title`	String	Title in the front matter of the wiki page
`title`	String	Title of the wiki page. At least one of `content`, `title`, `format` must be provided

Responses

Code	Description	Schema
`200`	OK	APIEntitiesWikiPage
`400`	Validation error	—
`404`	Not found	—
`422`	Unprocessable entity	—

Delete a wiki page for a project

`DELETE /api/v4/projects/{id}/wikis/{slug}`

Deletes a specified wiki page from a project.

Parameters

Name	Type	Description
`id` Path, required	String or integer	ID or URL-encoded path of the group or project
`slug` Path, required	String	URL-encoded slug of the wiki page, a unique string such as `dir%2Fpage_name`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Validation error	—
`404`	Not found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesWikiAttachment

Property	Type	Description
`branch`	String	Example: `main`
`file_name`	String	Example: `dk.png`
`file_path`	String	Example: `uploads/6a061c4cf9f1c28cb22c384b4b8d4e3c/dk.png`
`link`	Object	—
`link.markdown`	String	Example: `![dk](uploads/6a061c4cf9f1c28cb22c384b4b8d4e3c/dk.png)`
`link.url`	String	Example: `uploads/6a061c4cf9f1c28cb22c384b4b8d4e3c/dk.png`

APIEntitiesWikiPage

Property	Type	Description
`content`	String	Example: `Here is an instruction how to deploy this project.`
`encoding`	String	Example: `UTF-8`
`format`	String	Example: `markdown`
`front_matter`	Object	Example: `{"title":"deploy"}`
`slug`	String	Example: `deploy`
`title`	String	Example: `deploy`
`wiki_page_meta_id`	Integer (int64)	Example: `123`

APIEntitiesWikiPageBasic

Property	Type	Description
`format`	String	Example: `markdown`
`slug`	String	Example: `deploy`
`title`	String	Example: `deploy`
`wiki_page_meta_id`	Integer (int64)	Example: `123`

Other

Create a new application FE version

`POST /api/v4/applications`

This feature was introduced in GitLab 10.5

Request body (`application/json`)

Property	Type	Description
`confidential`	Boolean	The application is used where the client secret can be kept confidential. Native mobile apps \ and Single Page Apps are considered non-confidential. Defaults to true if not supplied Default: `true`
`name` Required	String	Name of the application Example: `MyApplication`
`redirect_uri` Required	String	Redirect URI of the application Example: `https://redirect.uri`
`scopes` Required	String	Scopes of the application. You can specify multiple scopes by separating\ each scope using a space

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplicationWithSecret
`400`	Bad Request	—

Delete an application FE version

`DELETE /api/v4/applications/{id}`

Delete a specific application

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the application (not the application_id)

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Renew an application secret FE version

`POST /api/v4/applications/{id}/renew-secret`

Renew the secret of a specific application

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the application (not the application_id)

Responses

Code	Description	Schema
`200`	OK	APIEntitiesApplicationWithSecret
`400`	Bad Request	—
`404`	Not Found	—

List group approval rules

`GET /api/v4/groups/{id}/approval_rules`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not Found	—

Create a new group approval rule

`POST /api/v4/groups/{id}/approval_rules`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch` , `all_protected_branch` , `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular` , `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Update a group approval rule

`PUT /api/v4/groups/{id}/approval_rules/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch`, `all_protected_branch`, `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular`, `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a group approval rule

`DELETE /api/v4/groups/{id}/approval_rules/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the group Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List Gitaly object pool members

`GET /api/v4/internal/gitaly/object_pool_members`

Parameters

Name	Type	Description
`disk_paths` Query, required	Array of strings	On-disk paths of the pool repositories. Limited to 500
`storage` Query, required	String	Name of the storage shard
`upstream_only` Query	Boolean	If `true`, returns only the upstream repository Default: `false`

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—

List project approval rules

`GET /api/v4/projects/{id}/approval_rules`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not Found	—

Create a new project approval rule

`POST /api/v4/projects/{id}/approval_rules`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch`, `all_protected_branch`, `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular`, `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

For new merge request

`GET /api/v4/projects/{id}/approval_rules/merge_requests/for-new-merge-request`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not Found	—

List merge_request approval rules

`GET /api/v4/projects/{id}/approval_rules/merge_requests/{merge_request_iid}`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	—	—

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not Found	—

Create a new merge_request approval rule

`POST /api/v4/projects/{id}/approval_rules/merge_requests/{merge_request_iid}`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	—	—

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch` , `all_protected_branch` , `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular` , `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Reset to project defaults

`POST /api/v4/projects/{id}/approval_rules/merge_requests/{merge_request_iid}/reset-to-project-defaults`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	—	—

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Update a merge_request approval rule

`PUT /api/v4/projects/{id}/approval_rules/merge_requests/{merge_request_iid}/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	—	—

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch` , `all_protected_branch` , `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular` , `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a merge_request approval rule

`DELETE /api/v4/projects/{id}/approval_rules/merge_requests/{merge_request_iid}/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	—	—

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

Update a project approval rule

`PUT /api/v4/projects/{id}/approval_rules/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Request body (`application/json`)

Property	Type	Description
`approval_rule` Required	Object	—
`approval_rule.approvals_required`	Integer	—
`approval_rule.approver_group_ids`	Array of integers	—
`approval_rule.approver_user_ids`	Array of integers	—
`approval_rule.custom_branch_rule`	String	Allowed values: `all_branch` , `all_protected_branch` , `manually`
`approval_rule.name`	String	—
`approval_rule.protected_branch_id`	Integer	—
`approval_rule.rule_type`	String	Allowed values: `regular` , `any_approver`
`approval_rule.target_branch_text`	String	—

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Delete a project approval rule

`DELETE /api/v4/projects/{id}/approval_rules/{approval_rule_id}`

Parameters

Name	Type	Description
`approval_rule_id` Path, required	Integer	—
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`

Responses

Code	Description	Schema
`204`	No Content	—
`400`	Bad Request	—
`404`	Not Found	—

List approval rules, code owners, and approval settings for merge request

`GET /api/v4/projects/{id}/merge_requests/{merge_request_iid}/approval_rules_and_code_owners`

Parameters

Name	Type	Description
`id` Path, required	String or integer	The ID or URL-encoded path of the project Example: `11`
`merge_request_iid` Path, required	Integer	The internal ID of an issuable

Responses

Code	Description	Schema
`200`	OK	—
`404`	Not found	—

Swagger compatible API description

`GET /api/v4/swagger_doc`

Responses

Code	Description	Schema
`200`	OK	—

Swagger compatible API description for specific API

`GET /api/v4/swagger_doc/{name}`

Parameters

Name	Type	Description
`name` Path, required	String	Resource name of mounted API
`locale` Query	String	Locale of API documentation

Responses

Code	Description	Schema
`200`	OK	—
`400`	Bad Request	—
`404`	Not Found	—

Revoke a GPG key owned by currently authenticated user FE version

`POST /api/v4/user/gpg_keys/{key_id}/revoke`

This feature was added in GitLab 10.0

Parameters

Name	Type	Description
`key_id` Path, required	Integer	The ID of the GPG key

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Add a GPG key to a specified user Available only for admins

`POST /api/v4/users/{id}/gpg_keys`

(FE version).

This endpoint has been overridden to change its behavior.

Parameters

Name	Type	Description
`id` Path, required	Integer	User ID

Request body (`application/json`)

Property	Type	Description
`key` Required	String	GPG Key

Responses

Code	Description	Schema
`201`	Created	APIEntitiesGpgKey
`400`	Bad Request	—
`404`	Not Found	—

Revokes an existing GPG key from a specified user

`POST /api/v4/users/{id}/gpg_keys/{key_id}/revoke`

Available only for admins. (FE version).

This feature was added in GitLab 10.0

Parameters

Name	Type	Description
`id` Path, required	Integer	The ID of the user
`key_id` Path, required	Integer	The ID of the GPG key

Responses

Code	Description	Schema
`201`	Created	—
`400`	Bad Request	—
`404`	Not Found	—

Schemas

Objects returned by the operations above and objects nested in their request bodies.

APIEntitiesApplicationWithSecret

Property	Type	Description
`application_id`	String	Example: `5832fc6e14300a0d962240a8144466eef4ee93ef0d218477e55f11cf12fc3737`
`application_name`	String	Example: `MyApplication`
`callback_url`	String	Example: `https://redirect.uri`
`confidential`	Boolean	Example: `true`
`id`	Integer (int64)	Example: `1`
`scopes`	Array	Example: `["api","read_user"]`
`secret`	String	Example: `ee1dd64b6adc89cf7e2c23099301ccc2c61b441064e9324d963c46902a85ec34`

APIEntitiesGpgKey

Property	Type	Description
`created_at`	String (date-time)	Example: `2017-09-05T09:17:46.264Z`
`id`	Integer (int64)	Example: `1`
`key`	String	Example: `-----BEGIN PGP PUBLIC KEY BLOCK-----`

История изменений

1.16.5

Дата выпуска: 22.10.2026
Сборка GitLab: 19.4.3_0

Ключевые изменения

- GitLab CE 19.4.3 закрывает критическое целочисленное переполнение в компиляторе регулярных выражений (CVE-2026-110264) и обход аутентификации через поиск email в LDAP при привязке учётных записей OmniAuth (CVE-2026-15723); обновите каждую установку, где используется автоматическая привязка к LDAP.
- Закрыты ещё шесть уязвимостей, среди них отравление кеша конечных точек обнаружения OAuth и OIDC и обход авторизации в API вех группы.
- Helm-чарт монтирует учётные данные OpenSearch, заданные ссылкой на секрет, а его задания резервного копирования, переиндексации и geo-logcursor запускают образ toolbox продукта.

Улучшения

- Helm-чарт:
  - Задание проверки перед обновлением получает запросы и лимиты ресурсов по умолчанию (`upgradeCheck.resources` : запрос 64Mi памяти, лимиты 200m CPU и 128Mi памяти), поэтому кластеры, чьи квоты или политики допуска отклоняют поды без них, больше не блокируют обновление.

Исправления

- Слияние fast-forward выполняется, когда тег носит имя целевой ветки.
- Workhorse отвечает на запрос архива кодом 429 вместо ошибки, когда Gitaly отклоняет его из-за исчерпания ресурсов.
- Claim `sub` ID-токенов CI/CD в пайплайнах запросов на слияние из форков содержит claims задания и исходного проекта.
- Helm-чарт:
  - Учётные данные OpenSearch, заданные ссылкой на секрет (`username` и `password` с `secret` и `key`), монтируются в поды Rails; встроенные учётные данные имеют приоритет, если заданы оба варианта, а ссылка без одного из двух полей останавливает рендеринг.
  - Задание резервного копирования, задание переиндексации базы данных и развёртывание geo-logcursor запускают образ toolbox из `global.codeHelm.images` ; раньше они собирали имя образа upstream GitLab и тянули образ, которого в поставке нет.

Обновления безопасности

- CVE-2026-110264, критическая: целочисленное переполнение в компиляторе регулярных выражений Ruby. Патч есть в продукте и в образах компонентов модуля.
- CVE-2026-15723, высокая: некорректная аутентификация при поиске email, который привязывает вход через OmniAuth к учётной записи LDAP.
- CVE-2026-112212, высокая: отравление кеша конечных точек обнаружения OAuth и OIDC; их URL теперь строятся из настроенного адреса инстанса.
- CVE-2026-12664, средняя: поле адреса email рабочего элемента в GraphQL читалось токеном только на чтение.
- CVE-2026-105777, средняя: внутренние учётные данные в заголовках ответа не вырезались, если имя заголовка было в другом регистре.
- CVE-2026-13319, средняя: обход авторизации в API вех группы через непроверяемый параметр.
- CVE-2026-105711, низкая: заголовок запроса на слияние оставался в закешированном фрагменте страницы.

Зависимости

- GitLab CE: 19.4.2 → 19.4.3.

1.15.4

Дата выпуска: 22.10.2026
Сборка GitLab: 19.3.6_0

Ключевые изменения

- GitLab CE 19.3.6 закрывает критическое целочисленное переполнение в компиляторе регулярных выражений (CVE-2026-110264) и обход аутентификации через поиск email в LDAP при привязке учётных записей OmniAuth (CVE-2026-15723); обновите каждую установку, где используется автоматическая привязка к LDAP.
- Закрыты ещё шесть уязвимостей, среди них отравление кеша конечных точек обнаружения OAuth и OIDC и обход авторизации в API вех группы.

Исправления

- Слияние fast-forward выполняется, когда тег носит имя целевой ветки.
- Workhorse отвечает на запрос архива кодом 429 вместо ошибки, когда Gitaly отклоняет его из-за исчерпания ресурсов.
- Claim `sub` ID-токенов CI/CD в пайплайнах запросов на слияние из форков содержит claims задания и исходного проекта.

Обновления безопасности

- CVE-2026-110264, критическая: целочисленное переполнение в компиляторе регулярных выражений Ruby. Патч есть в продукте и в образах компонентов модуля.
- CVE-2026-15723, высокая: некорректная аутентификация при поиске email, который привязывает вход через OmniAuth к учётной записи LDAP.
- CVE-2026-112212, высокая: отравление кеша конечных точек обнаружения OAuth и OIDC; их URL теперь строятся из настроенного адреса инстанса.
- CVE-2026-12664, средняя: поле адреса email рабочего элемента в GraphQL читалось токеном только на чтение.
- CVE-2026-105777, средняя: внутренние учётные данные в заголовках ответа не вырезались, если имя заголовка было в другом регистре.
- CVE-2026-13319, средняя: обход авторизации в API вех группы через непроверяемый параметр.
- CVE-2026-105711, низкая: заголовок запроса на слияние оставался в закешированном фрагменте страницы.

Зависимости

- GitLab CE: 19.3.5 → 19.3.6.

1.17.0

Дата выпуска: 19.10.2026
Сборка GitLab: 19.5.0_0

Ключевые изменения

- Сканирование безопасности доступно проектам в режиме раннего доступа: политики выполнения сканирования запускают Opengrep, Trivy и CodeScoring в пайплайнах проекта, а результаты попадают в список зависимостей, соответствие лицензий и DefectDojo (Enterprise Edition).
- GitLab CE 19.5.0 закрывает атаку по времени на токены вебхуков оповещений и чтение пути к конфигурации CI без разрешения read_code.
- Модуль Deckhouse Kubernetes Platform больше не задерживает обновление, пока фоновые миграции GitLab завершены с ошибкой или ещё выполняются.

Новый функционал

- SAST-сканирование с помощью Opengrep в режиме раннего доступа: политика выполнения сканирования добавляет в пайплайны проекта обязательное сканирование, которое ищет уязвимости в исходном коде; проект не может отключить сканирование или изменить его настройки (Enterprise Edition).
- Сканирование с помощью Trivy в режиме раннего доступа: политика выполнения сканирования запускает Trivy в пайплайнах проекта и проверяет зависимости на уязвимости

и лицензии, репозиторий на закоммиченные секреты, а образы контейнеров на уязвимости (Enterprise Edition).
- Страницы «Список зависимостей» и «Соответствие лицензий» проекта в режиме раннего доступа: они показывают компоненты из SBOM, который сканирование безопасности загрузило в последний пайплайн, и лицензии, под которыми эти компоненты распространяются (Enterprise Edition).
- Интеграция с CodeScoring в режиме раннего доступа: проект подключается к собственному серверу CodeScoring, а политика выполнения сканирования запускает в пайплайнах проекта его анализ состава ПО и поиск секретов (Enterprise Edition).
- Интеграции сканеров безопасности CodeScoring, PT AI и DefectDojo в режиме раннего доступа: они хранят адрес сервера, токен и доверие к сертификату, которые нужны обязательным сканированиям, поэтому политика сканирования не хранит секретов в репозитории (Enterprise Edition).
- Интеграция с DefectDojo в режиме раннего доступа: отчёты сканирования безопасности из пайплайнов проекта отправляются в инстанс DefectDojo, а найденные там уязвимости обновляются по мере выполнения пайплайнов (Enterprise Edition).

Улучшения

- Helm-чарт:
  - Задание миграций реестра контейнеров выполняется как хук post-install и post-upgrade и пересоздаётся при каждом обновлении, поэтому изменение values больше не ломает обновление на неизменяемом шаблоне задания.
- Модуль Deckhouse Kubernetes Platform:
  - Обновление продолжается, когда фоновые миграции GitLab завершились с ошибкой или ещё выполняются; оператор пишет предупреждение в журнал вместо остановки.

Обновления безопасности

GitLab CE 19.5.0 закрывает атаку по времени на токены вебхуков оповещений и чтение пути к конфигурации CI через GraphQL без разрешения read_code; sidekiq-cron, Gitaly и GitLab Pages обновлены с собственными исправлениями безопасности.

Документация

В документации модуля остаются только страницы о самом модуле; разделы о рекомендациях по безопасности, обслуживании и FAQ, которые теперь покрывает документация продукта, заменены ссылками на неё.

Зависимости

- GitLab CE: 19.4.2 → 19.5.0.
- GitLab Elasticsearch Indexer: 5.14.14 → 5.14.17.
- Helm-чарт и модуль Deckhouse Kubernetes Platform:
  - Go в образах компонентов: 1.26.9 → 1.27.2.

Журнал изменений

1.17

1.17.0

Версия GitLab: 19.5.0_0

- Deckhouse Code:
  - Добавлено:
    - SAST-сканирование с помощью Opengrep в режиме раннего доступа: политика выполнения сканирования добавляет в пайплайны проекта обязательное сканирование, которое ищет уязвимости в исходном коде; проект не может отключить сканирование или изменить его настройки (Enterprise Edition)
    - Сканирование с помощью Trivy в режиме раннего доступа: политика выполнения сканирования запускает Trivy в пайплайнах проекта и проверяет зависимости на уязвимости и лицензии, репозиторий на закоммиченные секреты, а образы контейнеров на уязвимости (Enterprise Edition)
    - Страницы «Список зависимостей» и «Соответствие лицензий» проекта в режиме раннего доступа: они показывают компоненты из SBOM, который сканирование безопасности загрузило в последний пайплайн, и лицензии, под которыми эти компоненты распространяются (Enterprise Edition)
    - Интеграция с CodeScoring в режиме раннего доступа: проект подключается к собственному серверу CodeScoring, а политика выполнения сканирования запускает в пайплайнах проекта его анализ состава ПО и поиск секретов (Enterprise Edition)
    - Интеграции сканеров безопасности CodeScoring, PT AI и DefectDojo в режиме раннего доступа: они хранят адрес сервера, токен и доверие к сертификату, которые нужны обязательным сканированиям, поэтому политика сканирования не хранит секретов в репозитории (Enterprise Edition)
    - Интеграция с DefectDojo в режиме раннего доступа: отчёты сканирования безопасности из пайплайнов проекта отправляются в инстанс DefectDojo, а найденные там уязвимости обновляются по мере выполнения пайплайнов (Enterprise Edition)
  - Безопасность:
    - GitLab обновлён до 19.5.0: закрыты атака по времени на токены вебхуков оповещений и чтение пути к конфигурации CI через GraphQL без разрешения read_code; sidekiq-cron, Gitaly и GitLab Pages обновлены с собственными исправлениями безопасности

    - Зависимости:
        - GitLab CE 19.4.2 → 19.5.0
        - GitLab Elasticsearch Indexer 5.14.14 → 5.14.17

- Omnibus Docker:
    - Зависимости:
        - Сборка GitLab 19.4.2_0 → 19.5.0_0

- Helm-чарт:
    - Изменено:
        - Задание миграций реестра контейнеров выполняется как хук post-install и post-upgrade и пересоздаётся при каждом обновлении (`registry.database.migrations.annotations`); раньше изменение values ломало обновление, потому что шаблон задания неизменяем
    - Зависимости:
        - Upstream-чарт 10.4.2 → 10.5.0, `global.gitlabVersion` 19.4.2 → 19.5.0
        - Go в образах компонентов 1.26.9 → 1.27.2

- Модуль:
    - Исправлено:
        - Обновление больше не блокируется, когда фоновые миграции GitLab завершились с ошибкой или ещё выполняются; оператор пишет предупреждение в журнал и продолжает обновление
        - Node.js 22.22.3 в образе сборки rails, чтобы устанавливались frontend-зависимости GitLab CE 19.5.0
        - Патч пользовательского маршрутизатора чтения Gitaly использует путь модуля v19
    - Документация:
        - Разделы о рекомендациях по безопасности, обслуживании и FAQ, которые теперь покрывает документация продукта, заменены ссылками на неё
    - Зависимости:
        - Go в образах компонентов 1.26.9 → 1.27.2
        - Gitaly, KAS, Pages и Shell следуют за GitLab 19.5.0; реестр контейнеров 4.41.0 → 4.42.0; Zoekt 1.19.1 → 1.20.1; GitLab Exporter 17.0.2 → 17.1.0; OpenBao 2.5.5 → 2.6.2

1.16

1.16.5

Версия GitLab: 19.4.3_0

- Deckhouse Code:
  - Безопасность:
    - GitLab обновлён до 19.4.3: закрыты CVE-2026-110264 (критическая, целочисленное переполнение в компиляторе регулярных выражений), CVE-2026-15723 (некорректная аутентификация при поиске email для автоматической привязки учётной записи OmniAuth к LDAP), CVE-2026-112212 (отравление кеша конечных точек обнаружения OAuth и OIDC), CVE-2026-12664 (контроль доступа к полю адреса email рабочего элемента в GraphQL), CVE-2026-105777 (раскрытие внутренних учётных данных в заголовках ответа), CVE-2026-13319 (обход авторизации в API вех группы) и CVE-2026-105711 (заголовок запроса на слияние оставался в кеше)
  - Исправлено:
    - Слияние fast-forward, когда тег носит имя целевой ветки
    - Workhorse возвращает 429 вместо ошибки, когда Gitaly отклоняет запрос архива из-за исчерпания ресурсов
    - Claims `sub` ID-токена пайплайнов запросов на слияние из форков содержат claims задания и исходного проекта
    - Окружение Rack освобождается только после закрытия тела ответа
  - Зависимости:
    - GitLab CE 19.4.2 → 19.4.3
    - gitlab-markup → 2.2.0, rdoc → 8.1.0
- Helm-чарт:
  - Добавлено:
    - Запросы и лимиты ресурсов по умолчанию для задания проверки перед обновлением (`upgradeCheck.resources` : запрос 64Mi памяти, лимиты 200m CPU и 128Mi памяти) для кластеров, чьи квоты или политики допуска отклоняют поды без них
  - Исправлено:
    - Учётные данные OpenSearch, заданные ссылкой на секрет (`username` / `password` с `secret` и `key`), монтируются в поды Rails; встроенные учётные данные имеют приоритет, а неполная ссылка останавливает рендеринг
    - Задание резервного копирования, задание переиндексации базы данных и развёртывание geo-logcursor берут образ toolbox из `global.codeHelm.images` ; раньше они собирали имя образа upstream GitLab

- Модуль:
  - Безопасность:
    - Ruby в образах компонентов содержит патч переполнения идентификатора повтора в регулярных выражениях (CVE-2026-110264) для Ruby 3.3, 3.4 и 4.0
  - Зависимости:
    - Gitaly, KAS и Pages v19.4.2 → v19.4.3

1.15

1.15.4

Версия GitLab: 19.3.6_0

- Deckhouse Code:
  - Безопасность:
    - GitLab обновлён до 19.3.6: закрыты CVE-2026-110264 (критическая, целочисленное переполнение в компиляторе регулярных выражений), CVE-2026-15723 (некорректная аутентификация при поиске email для автоматической привязки учётной записи OmniAuth к LDAP), CVE-2026-112212 (отравление кеша конечных точек обнаружения OAuth и OIDC), CVE-2026-12664 (контроль доступа к полю адреса email рабочего элемента в GraphQL), CVE-2026-105777 (раскрытие внутренних учётных данных в заголовках ответа), CVE-2026-13319 (обход авторизации в API вех группы) и CVE-2026-105711 (заголовок запроса на слияние оставался в кеше)
  - Исправлено:
    - Слияние fast-forward, когда тег носит имя целевой ветки
    - Workhorse возвращает 429 вместо ошибки, когда Gitaly отклоняет запрос архива из-за исчерпания ресурсов
    - Claims `sub` ID-токена пайплайнов запросов на слияние из форков содержат claims задания и исходного проекта
  - Зависимости:
    - GitLab CE 19.3.5 → 19.3.6
    - gitlab-markup → 2.2.0, rdoc → 8.1.0
- Omnibus Docker:
  - Зависимости:
    - Сборка GitLab 19.3.5_0 → 19.3.6_0
- Модуль:
  - Безопасность:
    - Ruby в образах компонентов содержит патч переполнения идентификатора повтора в регулярных выражениях (CVE-2026-110264) для Ruby 3.3, 3.4 и 4.0

    - Зависимости:
        - Gitaly, KAS и Pages v19.3.5 → v19.3.6

1.16

1.16.4

GitLab version: 19.4.2_0

- Module:
    - fixes:
        - Bumped haproxy version to v3.4.7
- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE to 19.4.2
        - Kept the documentation link on the Help page - the upstream setting that hides marketing entries also hid this product’s documentation link, and the checkbox for it is not in the interface, so an instance that had it on could not get it back
        - Added a warning before a merge request joins a merge train - if the train pipeline will contain manual jobs that must succeed, a dialog lists them, so users know the train and the merge requests behind it may stall
        - Fixed issues - closing merge request header links built from the request host, deletion of protected npm dist-tags, granular token boundary bypasses, CI pipelines run through workload abilities, an external authorization bypass through group work items, unanchored commit SHAs in raw file refs, DNS rebinding SSRF in imports, accessibility, code quality and Terraform reports readable without access to them, commit authors approving their own merge requests, and access checks skipped in work item hierarchy, work item notes and merge request discussions
        - Fixed the merge request widget staying empty after a merge or a close until its next poll tick; it now refreshes as soon as the merge request leaves the open state

1.16.3

GitLab version: 19.4.1_2

- Module:
    - fixes:
        - Webservice scaling edit for 3000 and 5000 targerUserCount. Now 4 max replicas for 3000 targerUserCount and 5 max for 5000 targerUserCount

- Deckhouse Code:
  - features:
    - Added Kubernetes agent operation on behalf of a user. If the agent configuration specifies “access_as=user”, requests routed through the proxy agent enter the cluster as that user; the user’s group membership is determined by their roles in projects and groups, and Kubernetes RBAC governs their permissions

1.16.2

GitLab version: 19.4.1_1

- Module:
  - fixes:
    - Fixed ldap fistName and lastName render
    - Bumped haproxy version to v3.4.6
  - features:
    - Added readTimeout, adminRequestTimeout and indexingRequestTimeout settings to spec.globalSearch.openSearch section that now allows to set timeouts for opensearch requests
- Deckhouse Code:
  - fixes:
    - Added DORA metrics pages for projects and groups - deployment frequency, lead time for changes, time to restore service and change failure rate, shown as summary tiles and daily trend charts
    - Replaced the logos and favicons with the new Deckhouse Code identity
    - Fixed Code Owners approval and reviewer auto-assignment counting files that a merge request does not change in the end
    - Added a Value streams dashboard for groups and projects, gathering overview counters, lifecycle metrics and DORA metrics into one Analyze page
    - Fixed a grammatical error in the Russian merge request widget message saying the merge request is ready to be merged by members who can write to the target branch
    - Unified two terms in the Russian interface - a fork is now «форк» everywhere instead of «ответвление», and approvals are spelled «апрув», as in the product documentation
    - Fixed the date range captions in the analytics reports

1.16.1

GitLab version: 19.4.1_0

- Module:
  - fixes:
    - Bumped haproxy version to v3.4.5

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE to 19.4.1
    - Updated GitLab version to 19.4.1, closing a denial of service through expensive regular expressions in CI/CD rules, a job log readable through GraphQL without the read_build_trace permission, tokens with the mcp scope reaching API routes outside MCP, a race between concurrent MCP requests, importers downgrading how imported users are matched, and path traversal through diff file URLs
    - Fixed the merge request approvals widget demanding an approval from Code Owners when every CODEOWNERS entry matching the changes was optional - such a merge request now reports that approval is optional, which is what the merge itself already allowed
    - The repository mirroring form now disables the ‘Mirror only protected branches’ option while the project has no protected branches, with a note under the option explaining that mirroring them would sync nothing; as soon as a branch is protected the option becomes available and the note turns into a link to the Protected branches section of the same page. A pull mirror now also states under the branch filter that the filter narrows tag synchronization - with ‘Mirror only protected branches’ or ‘Mirror matching branches’ set, a tag is synced only if its commit is on one of the mirrored branches. The ‘Mirror matching branches’ field now counts, as the expression is typed, how many branches it matches, or reports that the expression is invalid; the count is over the branches this repository already holds, not the remote. The protected-branches guard with its notes, the branch counter, and the tag note apply to pull mirroring only; the push form is unchanged from upstream. The regular expression of ‘Mirror matching branches’ is now compiled the same way in the form and in the synchronization, so a pattern the counter used to reject while the synchronization applied it anyway, such as a lookahead, no longer disagrees between the two; an expression that fails to compile is now refused when the mirror is saved instead of failing on every synchronization

1.16.0

GitLab version: 19.4.0_1

- Module:
  - features:
    - Added CiCatalogBundles bucket. Bucket is now also required for ci catalog bundles
    - Added new scaling options for 3000 and 5000 users in CRD CodeInstance
- Deckhouse Code:
  - features:
    - Updated GitLab version to 19.4.0, closing an SSO enforcement bypass through email one-time passwords, Sidekiq exhaustion through unbounded import jobs, unverified

Workhorse JWT on the remaining send-data endpoints, a missing protected environment check in terminal websocket authentication, merge request approval rules reset without the update_approvers permission, unvetted iframe sources in the content editor and unsanitised JSON table items, predictable ids in the OAuth consent form, overlong usernames accepted by SCIM and OAuth sign-in, a Terraform state upload path read from an unverified parameter, a generic package status changeable without update_package, and a service desk email routed onto a colliding slug; the GraphQL gem and the GitLab Elasticsearch Indexer are upgraded with their own fixes. Upstream removed the legacy AuditEvent model and every code path using it, which this fork's own per-scope audit event tables replace, and dropped the unused project_push_rules table. Upstream also dropped the Prometheus metric sidekiq_gvl_process_wait_seconds and the structured log field gvl_process_wait_s, leaving only their gvl_thread_wait counterparts, so dashboards and alerts built on the old names stop receiving data

- Added a group and project CI/CD analytics page. Five metrics for the selected period (total pipeline runs, median duration, and the share that failed, succeeded, or landed in another status), a duration chart (p50 and p95) and a runs-by-status chart, both broken down by day, with filters for the date range (7, 30, 90 or 180 days), the pipeline source and, on the project page, the branch. The group page rolls up all projects in the group and its subgroups. Access is limited by the applicable entitlement
- Added protected environments for projects and groups, with allowed-to-deploy rules, approval requirements, and deployment gating; changing a protected environment's attributes (external URL, tier, description), stopping it, or deleting it now also requires the actor to be on the allowed-to-deploy list, not just the Maintainer role
- Added pipeline warnings for the security reports DefectDojo refused. The pipeline page now names the report that did not arrive and the reason DefectDojo gave, instead of the failure being visible only in the instance log
- A "Value Stream Analytics" report has been added for groups and projects; it displays lifecycle analytics broken down by stage—including duration, volume, throughput, flow metrics, and task type distribution. Access is restricted by the corresponding permission
- A werf icon has been added for werf.yaml, werf.yml, werf-giterminism.yaml, werf-giterminism.yml, werf-includes.yaml, and werf-includes.yml files. Werf configuration files are now easily recognizable in the repository file list, on the file page, and in the merge request changes tree
- A link to merge trains has been added to the merge requests list

◦ fixes:

- Fixed the hourly user synchronization. It now also updates user names when the sync_name setting is on, and an empty email address coming from the directory no longer removes the user's email address

        - Fixed the Dependency list and License compliance pages showing “Could not read the SBOM report” on a valid CycloneDX report. A JSON parsing limit sized for API requests, not SBOM reports, rejected anything past about 1,000 components. A report that really is past the size these pages read now says so, instead of sending the reader to the scan job’s logs where nothing is wrong, and an implausibly large artifact is refused before it is read into memory. Parsing is also faster and uses less memory
        - Fixed an issue in the branch/merge request creation window for work items. The submit button click could be ignored while the branch name was still being validated. If a user entered a name and clicked the button immediately, nothing happened, and the window remained open with the button still active
        - Fixed a bug where a merge request would never be accepted into a merge train. When merge trains were enabled, the merge request had no way to join the train, and the API rejected every attempt with the message “No available merge train strategy”

1.15

1.15.3

GitLab version: 19.3.5_0

- Module:
    - fixes:
        - Bumped Golang version to 1.26.9
- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE to 19.3.5

1.15.2

GitLab version: 19.3.3_0

- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE to 19.3.3

1.15.1

GitLab version: 19.3.2_2

- Deckhouse Code:
    - fixes:
        - Fixed merge trains never taking a merge request. With merge trains enabled the merge request offered no way to join the train and the API refused every attempt with “No available merge train strategy”

1.15.0

GitLab version: 19.3.2_1

- Module:
  - fixes:
    - Disabled dynamic partitions on sidekiq deployment and backup jobs
    - Fixed credentials mount for External Opensearch
  - features:
    - Added alerts related to LDAP sync and Pull mirroring features

1.14

1.14.2

GitLab version: 19.3.2_0

- Module:
  - fixes:
    - Bumped Golang version to 1.26.8
- Deckhouse Code:
  - features:
    - Added ‘STRONGHOLD_SERVER_URL’, ‘STRONGHOLD_NAMESPACE’, ‘STRONGHOLD_AUTH_PATH’ and ‘STRONGHOLD_AUTH_ROLE’ as alternative CI/CD variable names for the four Stronghold connection s>
    - Added the ‘security_manager’ role to LDAP group sync’s ‘role_mapping’. An unrecognized name in ‘role_mapping’ is now logged as a sync error listing the valid names, without stop>
    - Added a project Code review analytics page (a table of the open merge requests under review, longest wait first, with milestone and label filters)
    - Added a group Contribution Analytics page (per-member push, issue and merge request counts over the group hierarchy, charts and a sortable table, three period tabs)
    - Added a group Productivity analytics page - it displays information about merge requests for the group and helps identify your development velocity
    - Added a project Merge request analytics page (mean time to merge, a throughput chart and a table of merged merge requests with filters and a date range)
  - fixes:
    - Fixed auto-merge getting stuck when several merge requests’ pipelines finished nearly at once

        - Moved the audit event log onto four tables of its own, one per scope, so audit events carry the sharding key organization isolation needs - the audit event page, API and CSV exp>
        - Updated Gitlab CE to 19.3.2

1.14.1

GitLab version: 19.3.1_1

- Module:
    - features:
        - Added support for multiple LDAP servers in CRD
    - fixes:
        - Fixed mount for secrets in mailroom and webservice deployment
        - Optimized rails based components startup
- Deckhouse Code:
    - features:
        - Added merge trains - merge requests line up per target branch and each one is validated against the result of merging everything ahead of it, so the target branch stays green without manual rebases. A project turns the train on in its merge request settings — with a cap on parallel pipelines and an option to let a merge skip the queue — and the queue itself is visible on its own page, in the merge request widget and through the REST and GraphQL APIs
    - fixes:
        - Fixed a 500 when creating a group in the Admin area with the runner registration setting submitted; the setting is now applied to the new group
        - Fixed the ‘Created before’ calendar on the admin Audit Events page running off the right edge of the window - it now attaches to the page body, so it can flip away from the edge
        - Added the ‘Use OpenSearch for search’ checkbox to Admin > Settings > Search, so an administrator can turn OpenSearch-backed search on and off from the interface instead of only through the API; the page warns when the indices are empty or their status cannot be read, because search switched on at that moment returns nothing or fails
        - Changed the admin area components list and the help page to report the version of the delivery the instance was installed from (the module, chart or package version), so an administrator sees the version they were shipped rather than the application’s own build tag

1.14.0

GitLab version: 19.3.1_0

- Module:
  - features:
    - Added checksumDisabled parameter to disable AWS checksums for registry component
  - fixes:
    - Bumped haproxy version to v3.4.4
    - Bumped Golang version to 1.26.7
- Deckhouse Code:
  - features:
    - Added metric ‘pull_mirror_errors_total’ with labels by project path and errors type
  - fixes:
    - Updated Gitlab CE to 19.3.1
    - Updated russian translations for various places including workitems, CI/CD catalogue, 2FA authentication
    - Fixed string case for russian translations. Unified translations for roles Reporter and Maintainers
    - Fixed internal server error, when project is created from builtin project template
    - Refreshed security credentials page, added GPG keys, SSH keys, tokens from project and group level service accounts. Inactive keys and tokens are hidden with filters
    - Fixed database write on LDAP synchronisation in maintenance mode. Sign-in won’t create groups and update membership in maintenance mode
    - Renamed metric ’ldap_sync_runs_total’, ’ldap_sync_runs_error_total’ , enriched with label ‘server’ both in logs and metrics

1.13

1.13.6

GitLab version: 19.2.6_0

- Module:
  - fixes:
    - Bumped Golang version to 1.26.8
- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE to 19.2.6

        - Mitigated issues related to an unauthenticated read of an arbitrary local file through the commits API, code injection into a merge request pipeline through an unpinned checkout commit, an unauthenticated denial of service through the GraphQL complexity multiplier, and routing spoofing through a group wiki

1.13.5

GitLab version: 19.2.5_0

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE to 19.2.5
    - Mitigated security issues CVE-2026-77801 and CVE-2025-10903

1.13.4

GitLab version: 19.2.4_0

- Module:
  - fixes:
    - Fixed preflight check for postgres extensions
  - docs:
    - Updated alerts table on maintenance page with actual severity level
    - Updated guide for custom certificate setup
- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE to 19.2.4
    - Mitigated security issues CVE-2026-19478 and CVE-2026-19650 - a critical code injection issue via a GraphQL directive and a cross-site request forgery issue in the GraphQL multiplex query handler
  - features:
    - Added merge request title validation - project maintainers can define a required title pattern and example, and merge requests with nonmatching titles are blocked from merging

1.13.3

GitLab version: 19.2.2_0

- Module:
  - fixes:
    - Added more sidekiq related panels to Grafana Dashboard
    - Bumped Golang version to 1.25.13

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE to 19.2.2
    - Fixed sign in when LDAP group sync error occurs
    - Fixed approval rules in merge quests from forked repo
    - Fixed approval rules bypass by administrator

1.13.2

GitLab version: 19.2.1_2

- Module:
  - fixes:
    - Added soft pod anti-affinity and topology spread constraints for haproxy and gitaly to improve pod distribution across nodes in HA mode
- Deckhouse Code:
  - fixes:
    - Fixed branding, description and translation on Help page
    - Fixed writing audit events to current database table

1.13.1

GitLab version: 19.2.1_1

- Deckhouse Code:
  - features:
    - Improved interface localization
    - Improved the relevance of code search and result previews. Files containing verbatim phrases now appear above files containing the same words scattered throughout the text, and pasting a code fragment with quotation marks and punctuation into the search bar no longer returns empty results
  - fixes:
    - Fixed a read timeout for the OpenSearch client. If the cluster accepts a connection but stops responding, search, index admin pages, and background indexing will fail with an error instead of waiting indefinitely; the timeout duration is configurable in fe.search.opensearch
    - Minor interface changes have been made to better comply with the company’s branding guidelines
    - The remaining telemetry channels have been closed - the version check setting is always reported as disabled, it cannot be enabled either in the admin panel or via the API, and the billing event endpoints introduced in 19.2.1 have been reset. The instance does not send anything out

        - The failure reason in the CI job status tooltip has been translated into Russian - the failed job tooltip no longer mixes the two languages
        - On the project import form by URL, the “Mirror repository” checkbox is enabled - the imported repository is immediately configured as a mirror

1.13.0

GitLab version: 19.2.1_0

- Module:
    - fixes:
        - Bumped haproxy version to v3.4.3
- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 19.2.1
        - Updated translation on import project page
        - Removed creation limit from service accounts
    - features:
        - Added Scoped labels for issues, merge requests

1.12

1.12.6

GitLab version: 19.1.7_0

- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 19.1.7

1.12.5

GitLab version: 19.1.6_0

- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 19.1.6

1.12.4

GitLab version: 19.1.4_0

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 19.1.4

1.12.3

GitLab version: 19.1.3_0

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 19.1.3

1.12.2

GitLab version: 19.1.2_1

- Module:
  - fixes:
    - Fixed mount for registry db credentials
    - Disable db partition sync on webservice and toolbox deployment
    - Optimize rails load for Jobs, including migrations
- Deckhouse Code:
  - fixes:
    - Fixed server error on global search requests

1.12.1

GitLab version: 19.1.2_0

- Module:
  - docs:
    - Added project import/export docs to FAQ
    - Small fixes in module documentation layout
  - fixes:
    - Edited alerts severity levels
    - Added new alert for unavailable repositories in Gitaly
    - Added SecurityPolicyExceptions for workload
- Deckhouse Code:
  - features:
    - Added Merge results pipelines

        - Added global search api
        - Added notification about missing license
    - fixes:
        - Updated Gitlab CE version to 19.1.2
        - Fixed bug with approvals sidebar

1.12.0

GitLab version: 19.1.1_3

- Module:
    - features:
        - Added global search feature for CRD CodeInstance
        - Added support for internal mode for opensearch instance
        - Added separate sidekiq deployment for global search feature
        - Added support for MTLS using the Istio module
    - fixes:
        - Bumped Golang version to 1.25.12
        - Bumped haproxy version to v3.4.2
        - Fixed registry migration job healthchecks
        - Added message in status field if preprocessing CodeInstance error occurs
        - Fixed toolbox tmp pvc reconcile
        - Fixed rediss preflight check
    - docs:
        - Removed step duplicate in registry v2 migration guide
- Deckhouse Code:
    - features:
        - Added External Status Checks for merge requests
        - Added OpenSearch support
        - Added a toggle to disable Code Owners approval checks
        - Added redirect path and user agent to unauthenticated audit event
        - Added maintenance mode
        - Added pull mirroring silent mode
    - fixes:
        - Updated Gitlab CE version to 19.1.1
        - Added auto-assign Code Owners as reviewers
        - Fixed approval rules for MRs created without a rules param
        - Cleaned up Russian translations and fixed maintenance message wording
        - Fixed search on Security Credentials admin pages
        - Updated user select label for multiple selections
        - Fixed selection single id_token for Vault CI secrets

    - chores:
        - Removed include directive support in CODEOWNERS

1.11

1.11.2

GitLab version: 19.0.3_0

- Module:
    - fixes:
        - Bumped haproxy version to v3.4.1
- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 19.0.3

1.11.1

GitLab version: 19.0.2_1

- Module:
    - fixes:
        - Fixed registry migration job fail on healtcheck
        - Fixed server-side backups
        - Bumped haproxy version to v3.4.0

1.11.0

GitLab version: 19.0.2_1

- Module:
    - fixes:
        - Fixed s3 storage configuration to support aws sdk update
- Deckhouse Code:
    - fixes:
        - Fixed s3 storage configuration to support aws sdk update
        - Add Prometheus metrics for pull mirroring
        - Granular permisions and fix auditor
        - Guard application setting when table does not exists

1.10

1.10.10

GitLab version: 18.11.6_0

- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 18.11.6

1.10.9

GitLab version: 18.11.5_1

- Deckhouse Code:
  - fixes:
    - Guard fe application setting when table does not exists

1.10.8

GitLab version: 18.11.5_0

- Module:
  - fixes:
    - Fixed operator sa token rotation
    - Bumped Golang version to 1.25.11
  - features:
    - Added cluster alert for token rotation errors
- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 18.11.5

1.10.7

GitLab version: 18.11.4_0

- Module:
  - fixes:
    - Fixed gitaly statefulset drop hook

1.10.6

GitLab version: 18.11.4_0

- Module:
  - fixes:
    - Fixed gitaly statefulset deploy on k8s 1.34
    - Fixed missing common webservice pod labels
    - Fixed missing PVC storage class name in gitaly stateful set template

1.10.5

GitLab version: 18.11.4_0

- Module:
  - fixes:
    - Fixed ssh session rate panel on monitoring dashboard
    - Fixed backup-before-update job ownerReference
  - features:
    - Added toolbox pod restart on backup-tmp pvc resize
    - Added `spec.appConfig.gpgCommitSigning` settings to enable web-based commits signing
  - docs:
    - Updated documentation about Redis TLS
    - Updated backup docs related to restore process
- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 18.11.4
    - Fixed audit events empty diff message
    - Fixed uninitialized constant on rails components startup

1.10.4

GitLab version: 18.11.3_0

- Module:
  - fixes:
    - Fixed backup before update bug on clean install
    - Fixed s3 preflight false positive error message
    - Backup cronjob now skips pages and registry backup if they are not enabled
  - docs:
    - Added guide for restoring specific repository
    - Added information about gzip compression

        - Added yandex s3 requied permissions for backup tmp bucket
        - Updated backup docs

- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 18.11.3

1.10.3

GitLab version: 18.11.2_0

- Module:
    - fixes:
        - Added ttl for jobs and cronjobs
        - Bumped haproxy version to v3.3.10

1.10.2

GitLab version: 18.11.2_0

- Module:
    - fixes:
        - Fixed mount for redis server CA certificate
        - Bumped haproxy version to v3.3.9

1.10.1

GitLab version: 18.11.2_0

- Module:
    - fixes:
        - Fixed panic in module hook ‘010_check_version_before_update’
        - Bumped haproxy version to v3.3.8

- Deckhouse Code:
    - fixes:
        - Updated Gitlab CE version to 18.11.2
        - Fixed bug with gitaly signed gpg
        - Fixed bug with hook memoization
        - Disabled Gitlab ping checkbox in UI
        - Updated translations

1.10.0

GitLab version: 18.11.1_0

- Module:
  - features:
    - Added `feVersion` to CodeInstance status subresource
    - Added `spec.appConfig.validate` option that validates appconfig fields before applying to cluster
    - Added CronJob for database re-index
    - Added mtls support for redis connections
  - chores:
    - Added deleting pages oauth application if pages disabled
    - Removed module hooks that used for previous release migrations
    - Removed CPU limits from ‘dependencies’ init container
  - fixes:
    - Bumped haproxy version to v3.3.7
    - Webservice now sets 3 puma workers and 8 puma threads for 100 targetUserCount
    - Added `D8CodeCoreServiceMetricsAbsent` alert to monitoring
    - Fixed s3 preflight check for YC `BucketsCountQuotaInCloudExceeded` error occurance
    - Removed CPU and memory limits from kube-rbac-proxy sidecar container
    - Fixed module servicemonitor template
    - Added registry DB to backup process
- Deckhouse Code:
  - fixes:
    - Updated Gitlab CE version to 18.11.1
    - Fixed ‘Create wiki’ button on group wiki page

1.9

1.9.16

GitLab version: v18.8.9_1

- Module:
  - fixes:
    - Fixed validation webhook scaling and scheduling

1.9.15

GitLab version: 18.8.9_1

- Module:
  - fixes:
    - Fixed backups listing for auto-cleanup backup stage

1.9.14

GitLab version: 18.8.9_1

- Module:
  - features:
    - Added `scaling.deploymentStrategy` property to CodeInstance to control deployments strategies
  - fixes:
    - Bumped Golang version to 1.25.9
  - docs:
    - Updated alerts descriptions. Added mitigation guide for `D8CodeDbPartitionMissing` alert
- Deckhouse Code:
  - fixes:
    - Update Gitlab CE to v18.8.9
    - Skipped creation of approval rule during merge request creation, if it is any_approver rule with 0 approvals required
    - Fixed push rule propagation errors
    - Fixed lfs import for pull mirroring
    - Fixed respect force push in pull mirroring
    - Added application setting for pull mirroring timeout

1.9.13

GitLab version: 18.8.8_0

- Module:
  - fixes:
    - Fixed operator migrations check api request

1.9.12

GitLab version: 18.8.8_0

- Module:
  - features:
    - Added secret ref support to `spec.storages.postgres.external.serverCA` CR property
    - Added secret ref support to `spec.features.registry.postgres.external.serverCA` CR property
    - Added secret ref support to `spec.storages.redis.external.serverCA` CR property
    - Deployment strategy edited to `Recreate` for 10 and 100 targetUserCount
    - Added `allowedGroups` to `spec.omniauth.provider`
    - Added `groupsAttribute` to `spec.omniauth.provider`
    - Added `adminGroups` to `spec.omniauth.provider`
  - fixes:
    - Fixed API calls from operator to webservice
  - chores:
    - Removed HTTP port from webservice service
  - docs:
    - Fixed full component schema on Getting Started page
    - Added rake task for secrets verify in module documentation
    - Added mention to backup docs that rails secrets are not included in backup
    - Updated translation for security recommendations page

1.9.11

GitLab version: 18.8.8_0

- Module:
  - features:
    - Added `spec.appConfig.monitoring.ipWhitelist` option to CRD
    - Fixed postgres CA validation in preflight checks
- Deckhouse Code:
  - fixes:
    - Update Gitlab CE to v18.8.8
    - Disabled pull mirroring for archived or pending deletion projects
    - Fixed push rules rollback migrations

1.9.10

GitLab version: 18.8.7_0

- Module:
  - features:
    - Added flag `backup.serverSide` to disable server-side backup
  - fixes:
    - Fixed monitoring creation for pages/registry
    - Fixed render for sidekiq podmonitor
    - Bumped Golang version to 1.25.8
    - Fixed monitoring whitelist for webservice deployments
  - chores:
    - Removed CPU limits from init containers
  - docs:
    - Updated custom certificate setup guide for web UI
- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.8.7
  - chores:
    - Disabled `What's new` section in UI

1.9.9

GitLab version: 18.8.5_2

- Module:
  - fixes:
    - Fixed operator panic on CR status reconcile
    - Fixed validation for registry postgres ssl mode

1.9.8

GitLab version: 18.8.5_2

- Module:
  - features:
    - Added options `backup.keepLast` to rotate automatic backups
  - fixes:
    - Bumped haproxy version to v3.3.6
    - Make `loadBalancerClass` option immutable in CodeInstance CR
    - Remove toolbox pod limits

        - Updated secret masking in operator logs
        - Default IngressClass name now is being taken from global params
        - Toolbox s3cfg `use_https` value for generic s3 provider
        - Registry postgres ssl mode and serverCA validation
        - Bug with redis auth password secretRef
        - Fixed operator panic when `omniauth.autoLinkUser` is set
        - S3 preflight checks, added s3 write and head check
        - Fixed emptyDir for toolbox and backup CronJob
        - Fixed module hook that enables monitoring
    - docs:
        - Update info about `placement.dedicated` behavior

1.9.7

GitLab version: 18.8.5_2

- Module:
    - fixes:
        - Fixed postgres sslmode propagation for container registry
        - Updated helm-lib to 1.68.2
        - Added monitoring rule for code operator
        - Added HA-mode for operator
    - features:
        - Added `loadBalancerClass` option in CodeInstance CR for Haproxy service
        - Added `s3Proxy` param for container registry feature in CodeInstance CR
    - docs:
        - Improved documentation for network configuration, backup, recovery, and migration processes
- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.8.5
    - fixes:
        - Added correct message for groups mention in ChatOps
        - Updated localization in UI

1.9.6

GitLab version: 18.8.4_2

- Module:
    - fixes:
        - Fixed backup job creation when no schedule in CodeInstance CR

        - Fixed postgres server CA handle in container registry configuration
        - Fixed S3 path style for server side backups

1.9.5

GitLab version: 18.8.4_2

- Deckhouse Code:
    - fixes:
        - Fixed issues on security credentials page

1.9.4

GitLab version: 18.8.4_1

- Module:
    - features:
        - Added support for secret ref in different properties in CodeInstance CRD
    - fixes:
        - Added fix to slow tag listing to container-registry
        - Added validation for Gitaly replicas in HA-mode
        - Fixed webservice ingress annotations
        - Fixed Pages ingress tls secret reconcile bug on pages ingress mode switch
        - Updated CR status when optional component is disabled
        - Fixed cache for module webhook
        - Bumped haproxy version to v3.3.4
        - Bumped Golang version to 1.24.13
    - chore:
        - Moved S3 client for operator to AWS SDK
        - Added .features.toolbox.enabled option to enable/disable toolbox deployment creation
        - Added haproxy scaling for HA mode
        - Updated scaling policy and docs
        - Reduced code-operator resources
        - Registry HPA scale policy modified
    - docs:
        - Updated S3 bucket list in Getting Started page
- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.8.4
        - Added support for multiple reviewers and assigners
        - Added audit events for push rules
        - Added a new user type auditor

    - fixes:
        - Added max login attempts and unlock period (in minutes) to the UI
        - Fix unchecking disabled flag in sign-in restriction section
        - Updated localizations in UI
        - Fixed audit events and approval rule creation when creating a merge request
        - Added service rake task for migration

1.9.3

GitLab version: 18.8.2_0

- Module:
    - fixes:
        - Fixed default `https.mode` for network section.

1.9.2

GitLab version: 18.8.2_0

- Module:
    - fixes:
        - Fixed migration module hook for webservice’s deployment

1.9.1

GitLab version: 18.8.2_0

- Module:
    - fixes:
        - Fixed reconcile for webservice deployment

1.9.0

GitLab version: 18.8.2_0

- Module:
    - features:
        - Added ingressClass field for Pages and Registry in CRD
        - Added alerts when number of partitions is not consistent in database
    - fixes:
        - Operator panic on spec.backup.cronSchedule omit
        - Redeploy pods on each reconcile with HA enabled
        - Migrate webservice resources from Helm
        - Fixed Gitlab Container registry with self-signed certificates
        - Bumped haproxy version to v3.3.2
        - Fixed gitlab-rails openid signing key header

        - Fixed incorrect backup s3 url in gitaly configmap
        - Fixed HPA and PDB resource deletion if HA is disabled
        - Migrations pod priority class name edited to production-high
        - Fixed postgres ssl mode env var for deployments
        - Fixed s3cmd error on S3 self-signed certificate
        - Fixed pages s3 credentials inherit in secrets
        - Updated helm-lib to 1.63.7
        - Fixed GC section render in configmap for Registry V2
    - chore:
        - Removed deprecated MRA resources
        - Moved backup job from Helm
        - Deprecated Helm engine for operator
        - Removed webservice and sidekiq CPU limits to optimize bandwidth
        - Improved performance by adding the RUBY_YJIT_ENABLE,MALLOC_ARENA_MAX variables to the Webservice and Sidekiq
        - Removed separate webservice deployment for internal api callbacks
    - docs:
        - Updated Registry V2 migration guide
        - Added example for Deckhouse authentication in Code
        - Fixed headers in security recommendation page
        - Added step to Omnibus migration guide
- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.8.2

1.8

1.8.9

GitLab version: 18.5.5_0

- Module:
    - fixes:
        - Fixed postgres ssl configure script for toolbox deployment

1.8.8

GitLab version: 18.5.5_0

- Module:
    - backport-1-9:
        - Fixed default `https.mode` for network section

1.8.7

GitLab version: 18.5.5_0

- Module:
  - docs:
    - Fixed changelog in module documentation

1.8.6

GitLab version: 18.5.5_0

- Module:
  - fixes:
    - Fixed secrets mount for Gitlab Exporter
    - Fixed postgres ssl mount in components
    - Fixed reconcilation for backup PVC
    - Bumped Golang version to 1.24.12
  - backport-1-9:
    - Fixed incorrect backup s3 url in gitaly configmap
- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.5.5
    - Added CodeOwners area in UI
    - Added Bot Access Tokens section for Security Credentials UI
  - fixes:
    - Update PAT/GAT: new default sort, hide revoked switcher, expiration section enhancements
    - Fixed PAT/GAT tables width
    - Added LDAP group sync filter validation on startup
    - Fixed LDAP name mask to correct only one group
    - Removed Support PIN area from UI
    - Corrected error message for diverged protected branches mirroring
    - Renamed required_approve status to not_approved
    - Added rake task for FE migrations rollback
    - Added default expiration 3 months for SA token in rake task
    - Fixed merge request author and committers approval revocation after corresponding approval rules changes

1.8.5

GitLab version: 18.5.4_0

- Module:
  - features:
    - Added embedded MRA. Separate MRA deployment is being deprecated. Scaling to 0 at first
- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.5.4
    - Added audit logs retention settings
    - Added audit events triggered when a user’s SSH key is deleted by an administrator
    - Added new audit events - `approval_rule_created`, `approval_rule_updated`, `approval_rule_deleted`, `approval_rule_propagated`
    - Logged audit events for changes to the retention period
  - fixes:
    - Disabled GitLab Runner version check
    - Fixed link to OAuth application on the audit event page
    - Fixed missing token_id for personal access token audit events
    - Improved human-readable text for nested JSON changes in audit events

1.8.4

GitLab version: 18.5.2_0

- Module:
  - features:
    - Add token expiration hours metrics and alerts
  - fixes:
    - Fixed a bug with incorrect token rotator logs
    - Fixed panic when `network.certificates.customCAs` items has no keys set
  - docs:
    - Add token expire troubleshooting

1.8.3

GitLab version: 18.5.2_0

- Module:
  - fixes:
    - Fixed tls section in redis config for sidekiq

1.8.2

GitLab version: 18.5.2_0

- Module:
  - fixes:
    - Migrate toolbox, sidekiq resources from Helm
    - Fixed certificate renewals causing new pods creation
    - Fixed redis rediss schema and redis serverCA validations
    - Fixed customCA configmapRef bug
    - Removed default replicas value from gitData section in CRD
    - Fixed .s3cfg file for toolbox
    - Fixed HPA default settings for rails based components
    - Bumped Golang version to 1.24.11
  - docs:
    - Refreshed migration process described in module documentation
    - Fixed descriptions, translations
    - Getting started docs prettified

1.8.1

GitLab version: 18.5.2_0

- Module:
  - fixes:
    - Fixed migrations for gitlab registry V2
  - docs:
    - Fixed module stage display in documentation
    - Improved documentation for Getting Started, Examples, and Scaling sections

1.8.0

GitLab version: 18.5.2_0

- Module:
  - features:
    - Attached OCI artifacts with VEX files for module images
  - fixes:
    - Scaling docs render
    - Updated base image from Debian 12 to 13
    - Fixed validation webhook for Omniauth configuration
    - Fixed pages oauth app register bug
    - Bumped haproxy version to v3.2.9

        - Mitigated CVE-2025-22868 and CVE-2025-22869 in kube-rbac-proxy image
        - Bumped Golang version to 1.24.10
        - Fix incoming email secret reconcile bug
    - docs:
        - Updated module documentation, added calculations for 5000 users
        - Fixed Getting started page in module documentation
    - backport-1-9:
        - Fix gitaly config map incorrect backup s3 url
- Deckhouse Code:
    - features:
        - Move module to `General Availability` stage
        - Update Gitlab CE to v18.5.2

1.7

1.7.4

GitLab version: 18.3.5_1

- Module:
    - backport-1-8:
        - Scaling docs render
        - Fixed validation webhook for Omniauth configuration
        - Fix incoming email secret reconcile bug
        - Updated module documentation, added calculations for 5000 users
        - Fixed Getting started page in module documentation

1.7.3

GitLab version: 18.3.5_1

- Module:
    - fixes:
        - Fix bug with predicate for deployment/statefulset
- Deckhouse Code:
    - fixes:
        - Access Keys Web UI fixes

1.7.2

GitLab version: 18.3.5_0

- Module:
  - fixes:
    - Fix bug with predicate for deployment/statefulset
- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.3.5
    - Added “Access Keys” section in administrator mode

1.7.1

GitLab version: 18.3.4_3

- Module:
  - fixes:
    - Fix internal-tls-ca
    - Fix webservice monitoring whitelist
  - docs:
    - Add ownLoadBalancer configuration mention in Getting-Started

1.7.0

GitLab version: 18.3.4_3

- Module:
  - fixes:
    - Bumped Golang version to 1.24.9
    - Bumped base debian image
    - Added validation for hostnames in CRD
    - Fixed reconciliation for internal TLS CA that triggers pod reload
    - Fixed reconciliation for Gitaly S3 backup credentials
    - Fixed components scaling for 3k users
    - Fix Gitlab SA token rotate
    - Fix toolbox s3cmd configmap
    - Fix pages hpa and pdb reconcile
    - Fix sidekiq resources scale
  - features:
    - Added `groupSync.prefix.nameMask` parameter for LDAP in CRD
    - Manual backups can be done without setting a schedule

    - docs:
        - Added recommendations for security configuration

1.6

1.6.5

GitLab version: 18.3.4_4

- Module:
    - backport-1-7:
        - Fix internal-tls-ca
        - Add ownLoadBalancer configuration mention in Getting-Started

1.6.4

GitLab version: 18.3.4_4

- Module:
    - fixes:
        - Fixed reconciliation for Gitaly S3 backup credentials
        - Fix Gitlab SA token rotate
        - Bumped Golang version to 1.24.9
        - Bumped base debian image
        - Added validation for hostnames in CRD
        - Fixed reconciliation for internal TLS CA that triggers pod reload
        - Fix toolbox s3cmd configmap

1.6.3

GitLab version: 18.3.4_4

- Module:
    - fixes:
        - Fixed helm chart render for ldap sync prefix name mask
    - docs:
        - Fix grammar and formalize style in security recommendations

1.6.2

GitLab version: 18.3.4_4

- Module:
    - docs:
        - Added recommendations for security configuration

- Deckhouse Code:
  - fixes:
    - Updated russian translation

1.6.1

GitLab version: 18.3.4_3

- Module:
  - backport-1-7:
    - Bumped Golang version to 1.24.8
    - Bumped base debian image
    - Added `groupSync.prefix.nameMask` parameter for LDAP in CRD
- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.3.4
  - backport-1-7:
    - Introduced full-text search in Gitlab FE

1.6.0

GitLab version: 18.3.2_1

- Module:
  - features:
    - Introduced Container Registry v2. Added database and gc section to CRD
    - Added deleting secrets, crd, pvc on module disabling
  - fixes:
    - Added init container for container registry when database section is present in CR
    - Migrate praefect resources from Helm
    - Migrate gitaly resources from helm
    - Fixed S3 default encryption options in CRD
    - Change outgoing mail default displayName
    - Fixed LDAP sync: added slugification of reserved extensions instead of discarding them in external user usernames.
    - Deleted module hooks that used for migration from 1.4 to 1.5 releases
    - Fixed Gitaly resources scaling
  - docs:
    - Added components schema and placeholder for hardening guideline
    - Added Gitaly components resources scaling precedence

- Deckhouse Code:
  - features:
    - Update Gitlab CE to v18.3.2

1.5

1.5.3

GitLab version: 18.2.8_0

- Deckhouse Code:
  - fixes:
    - Update Gitlab CE to v18.2.8

1.5.2

GitLab version: 18.2.6_1

- Module:
  - backport-1-6:
    - Fixed S3 default encryption options in CRD
    - Change outgoing mail default displayName
    - Fixed LDAP sync: added slugification of reserved extensions instead of discarding them in external user usernames.

1.5.1

GitLab version: 18.2.6_0

- Module:
  - fixes:
    - Fixed security issues
  - docs:
    - Add ldap docs

1.5.0

GitLab version: 18.2.6_0

- Module:
  - fixes:
    - Renamed sidekiq deploy, hpa and pdb
    - Fixed logic for CRD resources placement
    - Hardening security context in Gitaly pods. Added list of required capabilities to run
    - Added operator logs warning for postgresql version not fully support

        - Gitaly service type changed from `Headless` to `ClusterIP`
        - Praefect service type changed from `Headless` to `ClusterIP`
        - Fixed sidekiq startup checks
        - Added startup probe to sidekiq deployment probes
        - Fixed preflight ingress hook for network section in CRD
    - docs:
        - Added notes about supported postgresql version
        - Add ownLoadBalancer network faq for YCloud
        - Updated postgres requirements in Getting Started page
    - chore:
        - Removed migration for service monitors in operator
        - Removed unused variable ‘GITALY_FEATURE_DEFAULT_ON’ from deployments. This feature is default now.
        - Removed deprecated `useOwnLoadBalancer` from CRD
- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.2.6

1.4

1.4.10

GitLab version: 18.0.6_5

- Module:
    - backport-1-6:
        - Fixed S3 default encryption options in CRD
        - Change outgoing mail default displayName
        - Fixed LDAP sync: added slugification of reserved extensions instead of discarding them in external user usernames.

1.4.9

GitLab version: 18.0.6_3

- Module:
    - backport-1-5:
        - Fixed security issues
        - Add ldap docs

1.4.8

GitLab version: 18.0.6_3

- Module:
  - fixes:
    - Fixed backup.cronSchedule field ignoring in cronJob manifest
    - Fixed map read in redis preflight hook

1.4.7

GitLab version: 18.0.6_3

- Module:
  - fixes:
    - Fixed role naming in ldap sync config

1.4.6

GitLab version: 18.0.6_3

- Module:
  - fixes:
    - Added omniauth provider’s idp cert fingerprint sha1 validation
    - Fixed backup-before-update hook error handling
    - Webservice deployment probes timing changed
    - Gitaly statefulset probes timing changed
    - Sidekiq deployment probes timing changed

1.4.5

GitLab version: 18.0.6_3

1.4.4

GitLab version: 18.0.6_2

1.4.3

GitLab version: 18.0.6_1

- Module:
  - fixes:
    - Made `network.gitSsh` able to omit in CRD
    - Fixed code-operator gitlab api client certificate check
    - Fixed registry with https CustomCertificate mode
    - Fixed Gitaly PVC render

    - docs:
        - Added section about resources placement in Getting Started
- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.0.6
        - Added push rules for groups and projects

1.4.2

GitLab version: 18.0.3_3

- Module:
    - fixes:
        - Fixed validation for backup and git storage class in CRD
        - Added validation for SAML provider
        - Added validation for OIDC provider
        - Fixed some variables and error messages in preflights for components
        - Fixed custom certificate preflight check for non-helm components
        - Added mra and gitlab-api-client debug logs

1.4.1

GitLab version: 18.0.3_3

- Module:
    - fixes:
        - Added hostname migration hook
        - Fixed s3 storages for Generic provider

1.4.0

GitLab version: 18.0.3_3

- Module:
    - docs:
        - Added Getting started to module documentation
        - Added changelog to module documentation
        - Added note about Gitaly and Containerd V2
    - features:
        - Added `.spec.placement.dedicated` to CR to control resource distribution over nodes
    - fixes:
        - Fixed CRD reconciliation when secret or configmap changed
        - Added labels to CRD related resources

        - Dropped `.spec.features.pages.s3.external.bucketPrefix` in favor of `.spec.features.pages.s3.external.bucketName` in CRD
        - Dropped `Global` value support in https sections
        - Added ingress section for registry in CRD
        - Added ingress section for pages in CRD
        - Fixed ingress reconcile when `https.mode` is `CertManager` (pages, registry)
        - Set default email for root user on fresh install
        - Add ownLoadBalancer CR validations
        - Moved ServiceMonitor/PodMonitor from operator to module reconciliation to remove DH alerts
        - Fixed storage class propagation in Gitaly
        - Fixed updates on status subresource in CR
        - Fixed traffic policy for service shell
        - Bumped Golang version to 1.24.6
        - Bumped helm-lib to v1.63.0
        - Fixed security context at pod and container scope
        - Added more validation to ldap section
        - Fixed panic on backup s3 preflight
        - Added warning when set non expendable storage class in CRD
        - Fixed secret reconcile for omniauth
        - Fixed ValidatingAdmissionPolicy for backup PVC
        - Fixed ValidatingAdmissionPolicy for gitData PVC
        - Fixed custom certificate search NS
        - Fixed ingress annotations for webservice
    - chore:
        - Removed runner registration token

- Deckhouse Code:
    - features:
        - Update Gitlab CE to v18.0.3
        - Added activity track to group wiki
        - Added audit event for access level change in group wiki
        - Added full name of project and group to audit event on delete
        - Added API for audit events
    - fixes:
        - Disabled event data transmission

1.3

1.3.24

GitLab version: 17.11.7_1

- Module:
  - docs:
    - Updated LDAP examples in module documentation

1.3.23

GitLab version: 17.11.7_1

- Module:
  - fixes:
    - Fixed s3 section migration configmap

1.3.22

GitLab version: 17.11.7_1

- Module:
  - fixes:
    - Backport fix for backup s3 preflight check
    - Separated Gitaly statefulset and PVC

1.3.21

GitLab version: 17.11.7_0

- Module:
  - fixes:
    - Fixed init container for Gitaly pods
  - docs:
    - Updated module documentation with information about Containerd V2
- Deckhouse Code:
  - fixes:
    - Updated Gitlab FE version to 17.11.7

1.3.20

GitLab version: 17.11.6_1

- Module:
  - fixes:
    - Fixed storage class propagation in Gitaly

1.3.19

GitLab version: 17.11.6_1

- Module:
  - fixes:
    - Fixed jq filter in module hook

1.3.18

GitLab version: 17.11.6_1

- Module:
  - chore:
    - Removed ‘disabled’ ssl mode from postgres in CRD
  - fixes:
    - Fixed type conversion in migration module hook

1.3.17

GitLab version: 17.11.6_1

- Module:
  - fixes:
    - Updated base image to the newest version to mitigate CVEs
    - Fixed postgres and redis tls secret mount in jobs
    - Fixed redis section in rails based services
  - docs:
    - Added example for Redis TLS in CRD
- Deckhouse Code:
  - fixes:
    - Updated Gitlab FE version to 17.11.6
    - Changed type for account `deckhouse_sa` to service account
    - Fixed initial page and added an alert when no admin user exists in the instance.

1.3.16

GitLab version: 17.11.4_3

- Module:
  - fixes:
    - Fixed postgres connection check
    - Updated Golang version to 1.24.5 in operator

1.3.15

GitLab version: 17.11.4_3

- Deckhouse Code:
  - features:
    - Added wiki at group level
  - fixes:
    - Updated russian translation
- Module:
  - fixes:
    - Fix s3 buckets webhook validations
    - Fix ownLoadBalancer services delete on disabling it

1.3.14

GitLab version: 17.11.4_2

- Module:
  - fixes:
    - Fixed definition for `network.certificates.customCAs` in CRD

1.3.13

GitLab version: 17.11.4_2

- Module:
  - features:
    - Added `network.certificates` section to CRD to handle custom TLS and CA certificates.
  - fixes:
    - Fixed NodePort range for `gitSsh.service` in CRD
    - Bumped MRA to v1.1.3
    - Fixed ValidatingAdmissionPolicy rules

1.3.12

GitLab version: 17.11.4_2

- Deckhouse Code:
  - features:
    - Added vault integration to gitlab CI
    - Added webhooks at group level
- Module:
  - fixes:
    - Bumped base images version to mitigate CVEs
    - Bumped Golang version to 1.24.5
    - Fixed CA certificate field in CRD for postgres
    - Fixed CA certificate field in CRD for redis
    - Fixed registry bucket name in toolbox and backup job
    - Added validation for external omniauth providers in CRD
    - Added validation for registry params
    - Fixed pages s3 configuration and registration job
    - Field `s3.external.endpoint` and `s3.external.region` now required for Generic provider
    - Fixed a bug where the gitlab client would not return errors for requests with an invalid status code
    - Fixed backup job reconciliation and added more validation to `backup` section in CRD
    - Fixed ldap parameters quote
  - docs:
    - Add ownLoadBalancer nginx-ingress mode required

1.3.11

GitLab version: 17.11.4_0

- Module:
  - features:
    - Added `spec.ownloadbalancer` section
    - Added `spec.ownloadbalancer.annotations`. Now it’s available to set custom annotations to ownLoadBalancer service
  - chore:
    - Deprecate `spec.useOwnLoadBalancer` key. Moved to `spec.ownloadbalancer.enabled` key
    - Deprecate `spec.ownLoadBalancerHttpBackends` key. Moved to `spec.ownloadbalancer.httpBackends` key
    - Remove `Global` from https available modes

    - docs:
        - Edited https modes available, removed `Global`
    - fixes:
        - Fixed https CustomCertificate mode (`.https.mode=CustomCertificate`)
        - Fixed https CertManager mode (`.https.mode=CertManager`)
        - Fixed bug with operator gitlab internal api usage

1.3.10

GitLab version: 17.11.4_0

- Module:
    - fixes:
        - Fixed mail secrets reconcile
        - Updated ports in some ServiceMonitors
        - Fixed bug with s3 external storage configs
        - Fixed sidekiq scaling map execution conditions
        - Fixed webservice scaling map execution conditions
        - Edited gitaly probes from grpc to exec type
        - Added hpa to predicates
        - Fixed rails replicas template with hpa enabled
        - Fixed deployment redeploy bug with targetUserCount more than 10
        - Fixed registry metrics port
- Deckhouse Code:
    - fixes:
        - Updated Gitlab FE version to 17.11.4

1.3.9

GitLab version: 17.11.3_2

- Module:
    - fixes:
        - Fixed omniauth provider secret missing
        - Fixed outgoing email smtp secret missing
        - Fields `*.s3.external.accessKey` and `*.s3.external.secretKey` are now required
        - Fields `*.s3.external.region` is now required if `*.s3.external.provider: Generic`
        - Fixed registry configmap syntax error
        - Fixed haproxy configmap syntax error
        - Fixed migration job `redis-secret` secret lose
        - Fixed pages reconcile logic

1.3.8

GitLab version: 17.11.3_2

- Module:
  - features:
    - Added param `network.ownLoadBalancerHttpBackends` in CRD to control haproxy routing

1.3.7

GitLab version: 17.11.3_2

- Module:
  - fixes:
    - Fixed handle of deprecated field in module hooks
    - Fixed pages S3 preflight hook
- Deckhouse Code:
  - fixes:
    - Updated Gitlab FE version
    - Disabled `Enabled sign-in` and `Enabled sign-up` in UI
  - features:
    - Added pull mirroring feature

1.3.6

GitLab version: 17.11.3_1

- Module:
  - chore:
    - Removed deprecated fields `pages.s3.bucketPrefix` and `backup.restoreFromBackupMode` from CRD

1.3.5

GitLab version: 17.11.3_1

- Module:
  - docs:
    - Fixed menu title in public module documentation
- Deckhouse Code:
  - fixes:
    - Fixed Gitlab FE bootstrap issue

1.3.4

GitLab version: 17.11.3_0

- Module:
  - features:
    - Added `gitData.replicas` param to CRD to control Gitaly nodes in high-available mode
  - fixes:
    - Fixed Omniauth provider arguments in CRD
  - docs:
    - Fixed bucket names in public module documentation