FLANT
Deckhouse
Kubernetes Platform

Дорожная карта по получению сертификата соответствия **Deckhouse** Kubernetes Platform **требованиям безопасности** информации, установленным **ФСТЭК** России

+7 (495) 721-10-27

contact@deckhouse.ru deckhouse.ru

Оглавление

От разработчиков

Этот документ описывает путь получения сертификата соответствия Kubernetes-платформы Deckhouse требованиям ФСТЭК России. Приведем описание и перечень всех компонентов, которые включены в состав сертифицированной версии Deckhouse.

Deckhouse — это платформа, которая помогает компаниям управлять, масштабировать и автоматизировать создание идентичных кластеров Kubernetes в любой инфраструктуре. Как результат — сокращение временных и финансовых издержек.

Под капотом Deckhouse — ванильный Kubernetes и сбалансированный набор Open Source-инструментов, которые стали индустриальным стандартом. Внесен в единый реестр российского ПО (реестровая запись №12338 от 21.12.2021). У Deckhouse Kubernetes Platform есть две редакции: Community Edition и Enterprise Edition.

Сертификация необходима для подтверждения того, что Deckhouse соответствуют действующим требованиям безопасности ФСТЭК России и может использоваться при работе с конфиденциальными данными в составе тех информационных систем, в которых использование сертифицированных продуктов является обязательным (например, госкомпании, госкорпорации, банки, федеральные и региональные органы исполнительной власти).

Deckhouse будет сертифицирован на соответствие Требованиям по безопасности информации, устанавливающим уровни доверия к средствам технической защиты информации и средствам обеспечения безопасности информационных технологий (утверждены приказом ФСТЭК России от 2 июня 2020 г. № 76.) — по 4 уровню доверия; Требованиям безопасности информации к средствам контейнеризации (утверждены приказом ФСТЭК России № 118 от 4 июля 2022 г.) — по 4 классу защиты.

В результате сертификации у Deckhouse Kubernetes Platform появится сертифицированная в ФСТЭК России редакция — Certified Security Edition. Сертифицированная ФСТЭК редакция Deckhouse будет развиваться отдельно от основной ветки платформы. Deckhouse станет первой российской сертифицированной платформой оркестрации контейнеров.

Сертифицированную редакцию Deckhouse можно будет использовать для защиты информации, не содержащей сведений, которые составляют государственную тайну:

- в государственных информационных системах до 1 класса защищенности включительно согласно приказу ФСТЭК России от 11 февраля 2013 г. № 17 «Об утверждении требований о защите информации, не составляющей государственную тайну, содержащейся в государственных информационных системах» и методическому документу от 11 февраля 2014 г. «Меры защиты информации в государственных информационных системах»;

- в информационных системах персональных данных до 1 уровня защищенности включительно согласно приказу ФСТЭК России от 18 февраля 2013 г. № 21 «Об утверждении состава и содержания организационных и технических мер по обеспечению безопасности персональных данных при их обработке в информационных системах персональных данных»;
- на значимых объектах критической информационной инфраструктуры Российской Федерации до 1 категории значимости включительно согласно приказу ФСТЭК России от 25 декабря 2017 г. № 239 «Об утверждении требований по обеспечению безопасности значимых объектов критической информационной инфраструктуры Российской Федерации»;
- в защищенных автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды до 1 класса защищенности включительно согласно приказу ФСТЭК России от 14 марта 2014 г. № 31 «Об утверждении Требований к обеспечению защиты информации в автоматизированных системах управления производственными и технологическими процессами на критически важных объектах, потенциально опасных объектах, а также объектах, представляющих повышенную опасность для жизни и здоровья людей и для окружающей природной среды».

Сертификация — это процедура оценки соответствия продукта требованиям, в рамках которой, согласно положения о сертификации средств защиты информации, утвержденном приказом ФСТЭК России № 55 от 3 апреля 2018 г., проводится проверка средства защиты заявленным требованиям с привлечением следующих сторон:

- ФСТЭК России,
- органа по сертификации,
- испытательной лаборатории,
- самого разработчика.

В системе сертификации ФСТЭК России каждому из участников процесса отведены определенная роль и зона ответственности:

1. ФСТЭК России организует проведение сертификации средств защиты информации, разрабатывает и устанавливает в пределах своей компетенции требования по безопасности информации к средствам защиты информации, а также выполняет функции федерального органа по сертификации.
2. Орган по сертификации осуществляет сертификацию средств защиты информации, оформляет сертификаты соответствия средств защиты информации требованиям по безопасности информации.
3. Испытательная лаборатория проводит сертификационные испытания средств защиты информации и по их результатам оформляет технические заключения и протоколы. Испытательные лаборатории должны обеспечивать полноту сертификационных испытаний средств защиты информации и достоверность их результатов.
4. Разработчик производит средства защиты информации в соответствии с требованиями по безопасности информации.

Для приведения к соответствию требованиям ФСТЭК России в платформе Deckhouse было сделано следующее:

- переписаны участки исходного кода написанные на интерпретируемых языках программирования на компилируемый язык программирования Go;
- доработаны механизмы регистрации событий безопасности;
- реализованы механизмы контроля целостности с использованием алгоритмов ГОСТ;
- подключен банк данных уязвимостей ФСТЭК России;
- выполнены прочие требования, определенные в приказе ФСТЭК России № 118.

Мы сертифицируем Deckhouse по схеме серийного производства средства защиты информации (раздел I, пункт 12 приказа ФСТЭК России № 55 от 3 апреля 2018 г.) и после получения сертификата соответствия будем поддерживать и развивать сертифицированную версию Certified Security Edition параллельно с основной веткой Deckhouse.

Команда разработки Deckhouse

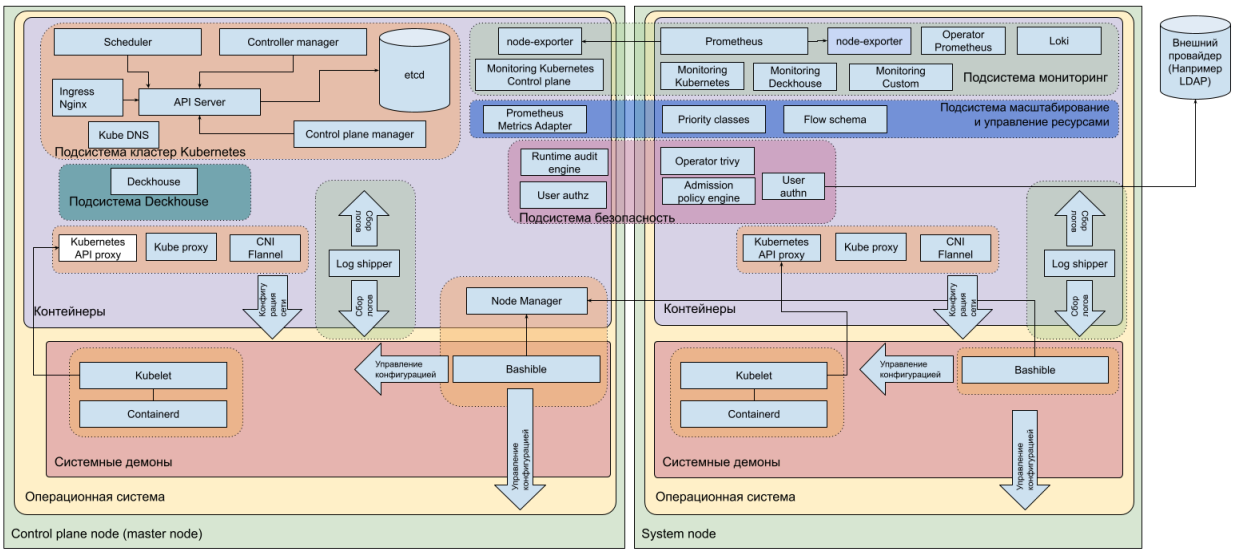
Дорожная карта сертификации Deckhouse Platform

Что уже сделано

Январь 2023	Принято решение о том, что необходимо сертифицировать Deckhouse Kubernetes Platform (далее — Deckhouse) в системе сертификации ФСТЭК России на соответствие требованиям по безопасности информации к средствам контейнеризации, утв. приказом ФСТЭК России № 118 от 4 июля 2022 г. (далее — приказ ФСТЭК).
Январь 2023	Начата подготовка к получению лицензий ФСТЭК России на деятельность по технической защите конфиденциальной информации и на деятельность по разработке и производству средств защиты конфиденциальной информации.
Февраль 2023	Анализ требований приказа ФСТЭК, формирование Требований к доработке Deckhouse.
Февраль 2023	Заключение договора аренды для подготовки защищаемого помещения, необходимого для осуществления лицензируемых видов деятельности.
Февраль 2023	Получение сертификата соответствия ГОСТ Р ИСО 9001-2015 (Национальный стандарт Российской Федерации. Системы менеджмента качества. Требования).
Март 2023	Приобретение программных, технических средств, необходимых для осуществления лицензируемых видов деятельности и нормативно-методической документации.
Март – июль 2023	Разработка документации, необходимой для сертификации Deckhouse.
Март – август 2023	Доработка Deckhouse для приведения в соответствие требованиям приказа ФСТЭК.
Апрель 2023	Монтаж, установка и настройка средств защиты, аттестация автоматизированной системы и защищаемого помещения.

Май 2023	Подача документов в ФСТЭК России для получения лицензий.
Июнь 2023	Лицензии ФСТЭК России получены.
Июль 2023	Подана заявка на сертификацию Deckhouse в ФСТЭК России.
Август 2023	Получено решение ФСТЭК России о возможной сертификации.
Сентябрь 2023	Передача Deckhouse в испытательную лабораторию.
Сентябрь – декабрь 2023	Проведение испытаний Deckhouse.
Январь – март 2024	Получение заключения органа по сертификации.
Мы находимся здесь	
Q2 2024	Получение сертификата соответствия.
Далее	Поддержка и развитие Deckhouse Certified Security Edition.

1. Состав сертифицированной версии Deckhouse



В этом разделе приведена наглядная структурная схема Deckhouse Kubernetes Platform. С ее помощью можно проследить взаимосвязи между подсистемами, модулями и компонентами нашей платформы. Более подробное описание каждого модуля и компонента можно прочитать, кликнув на его название.

Подсистема	Модуль	Компонент
Кластер Kubernetes	Control Plane manager	Control Plane manager
		etcd
		API Server
		Controller manager
		Scheduler
		Kubernetes API proxy
	Ingress-Nginx	Controller
		Controller failover
		Proxy failover

		Kruise controller manager
	Node Manager	Bashible apiserver
		Bashible
		containerd
		crictl
		curl
		jq
		kbectl
		kubelet
		Kubernetes CNI
		Toml-merger
	Kube DNS	CoreDNS
	kube-proxy	kube-proxy
	CNI Flannel	Flannel
Deckhouse	Deckhouse	Deckhouse
		Webhook handler
Мониторинг	Monitoring Kubernetes	Kube state metrics
		Node exporter
	Monitoring Kubernetes Control Plane	Control Plane proxy
	Operator Prometheus	Prometheus config reloader
		Prometheus operator
	Loki	Loki
	Log shipper	Log shipper agent
	Prometheus	Alertmanager
		Grafana

		Prometheus main
		Prometheus longterm
		Trickster
Масштабирование и управление ресурсами	Prometheus metrics adapter	Prometheus metrics adapter
Безопасность	User authz	User authz webhook
	Operator Trivy	Operator
		Node collector
		Scan vulnerability report
		BDU Exporter
	Admission policy engine	Gatekeeper audit
		Gatekeeper controller manager
	User authn	Dex
	Runtime audit engine	Runtime audit engine

2. Сопоставление функций безопасности ПО Deckhouse Kubernetes Platform

В этом разделе мы приводим таблицу, в которой сопоставляются требования ФСТЭК и компоненты Deckhouse Kubernetes Platform, которые реализуют эти требования. Более подробное описание каждого компонента можно прочитать, кликнув на его название.

Функция безопасности	Компонент
Требования к обеспечению изоляции контейнеров	
❖ Реализация механизмов изоляции контейнеров	Ядро ОС
❖ Изоляция пространств идентификаторов процессов контейнеров	Ядро ОС
❖ Изоляция пространств имен для межпроцессного взаимодействия контейнеров	Ядро ОС
❖ Изоляция пространств имен для пользователей и групп контейнеров	Ядро ОС
❖ Изоляция пространств имен хостов и доменов контейнеров	Ядро ОС
❖ Изоляция сетевых пространств имен контейнеров	Ядро ОС
❖ Изоляция пространств имен для иерархии каталогов контейнеров	Ядро ОС
Требования к обеспечению выявления уязвимостей в образах контейнеров	
❖ Выявлять известные уязвимости при создании, установке образа контейнера в информационной (автоматизированной) системе и хранении образов контейнеров во взаимодействии с сертифицированным средством контроля и анализа защищенности на основе сведений,	Operator (Operator Trivy)

содержащихся в банке данных угроз безопасности информации, ведение которого осуществляется ФСТЭК России, а также в иных источниках, содержащих сведения об известных уязвимостях	
❖ Оповещать о выявленных уязвимостях в образах контейнеров разработчика образов контейнеров и администратора безопасности информационной (автоматизированной) системы	Operator (Operator Trivy)
❖ Запрещать создание образов контейнеров, содержащих известные уязвимости критического и высокого уровня опасности	Gatekeeper controller manager (Admission policy engine)
❖ Осуществлять выявление известных уязвимостей образов контейнеров не реже одного раза в неделю.	Operator (Operator Trivy)
Требования к проверке корректности конфигурации контейнеров в Deckhouse	
❖ Ограничение прав прикладного программного обеспечения, выполняемого внутри контейнера, на использование периферийных устройств, устройств хранения данных и съемных машинных носителей информации (блочных устройств), входящих в состав информационной (автоматизированной) системы	Gatekeeper controller manager (Admission policy engine)
❖ Ограничение прав прикладного программного обеспечения, выполняемого внутри контейнера, на использование вычислительных ресурсов (оперативной памяти, операций ввода-вывода за период времени) хостовой операционной системы	Gatekeeper controller manager (Admission policy engine)

❖ Монтирование корневой файловой системы хостовой операционной системы в режиме «только для чтения»	Gatekeeper controller manager (Admission policy engine)
Требования к контролю целостности контейнеров и их образов в Deckhouse	
❖ Контролировать самостоятельно или с применением средств контроля целостности хостовой операционной системы и иных сертифицированных средств защиты информации целостность образов контейнеров и исполняемых файлов контейнеров	containerd, Runtime audit engine (Runtime audit engine), Gatekeeper controller manager (Admission policy engine)
❖ Информировать администратора информационной (автоматизированной) системы и администратора безопасности средства контейнеризации о нарушении целостности объектов контроля	Prometheus (Prometheus)
❖ Контролировать целостность параметров настройки Deckhouse	Runtime audit engine (Runtime audit engine)
❖ Контролировать целостность сведений о событиях безопасности самостоятельно или во взаимодействии с хостовой операционной системой и иными сертифицированными средствами защиты информации	Loki (Loki)
❖ Контролировать целостность образов контейнеров и параметров настройки Deckhouse при установке образа контейнера в информационной (автоматизированной) системе и далее периодически за счет применения цифровой подписи самостоятельно или во взаимодействии с хостовой операционной системой и иными сертифицированными средствами защиты информации	Gatekeeper controller manager (Admission policy engine)

❖ Блокировать запуск образа контейнера при нарушении его целостности	containerd, Gatekeeper controller manager (Admission policy engine)
Требования к регистрации событий безопасности	
❖ Регистрировать события, относящиеся к инцидентам безопасности Deckhouse, связанные с попытками осуществления несанкционированного доступа к Deckhouse	Runtime audit engine (Runtime audit engine)
❖ Оповещать администратора безопасности Deckhouse и администратора информационной (автоматизированной) системы об инцидентах безопасности	Prometheus (Prometheus)
❖ Выполнять действия, являющиеся реакцией на инциденты безопасности	Prometheus (Prometheus)
❖ Осуществлять сбор и хранение записей в журнале событий безопасности, которые позволяют определить, когда и какие действия происходили	Loki (Loki)
Требования к записями в журнале событий безопасности	
❖ Для каждой функции безопасности в Deckhouse должен быть определен перечень событий, необходимых для регистрации и учета	Примечание. Это требование реализуется администратором — он сам должен выбрать, что именно ему необходимо регистрировать.
❖ Для регистрируемых событий безопасности в каждой записи журнала событий безопасности должны указываться номер (уникальный идентификатор) события, дата, время, тип события безопасности	Runtime audit engine (Runtime audit engine)
❖ Записи журнала событий безопасности должны представляться в структурированном виде и содержать информацию о времени события безопасности, взятую из хостовой операционной системы	Runtime audit engine (Runtime audit engine)

❖ Средство контейнеризации должно осуществлять запись событий безопасности контейнеров в журнал событий безопасности информационной (автоматизированной) системы с указанием идентификатора контейнера	Runtime audit engine (Runtime audit engine)
❖ Журнал событий безопасности средства контейнеризации должен быть доступен только для чтения. При исчерпании области памяти, отведенной под журнал событий безопасности средства контейнеризации, средство контейнеризации должно осуществлять архивирование журнала с последующей очисткой указанного журнала	Loki (Loki)
Типы событий безопасности, которые необходимо регистрировать	
❖ Неуспешные попытки аутентификации пользователей Deckhouse	Runtime audit engine (Runtime audit engine)
❖ Создание, модификация и удаление образов контейнеров	Runtime audit engine (Runtime audit engine)
❖ Получение доступа к образам контейнеров	Runtime audit engine (Runtime audit engine)
❖ Запуск и остановка контейнеров с указанием причины остановки	Runtime audit engine (Runtime audit engine)
❖ Изменение ролевой модели	Runtime audit engine (Runtime audit engine)
❖ Модификация запускаемых контейнеров	Runtime audit engine (Runtime audit engine)
❖ Выявление известных уязвимостей в образах контейнеров и некорректности конфигурации	Gatekeeper controller manager (Admission policy engine)
❖ Факты нарушения целостности объектов контроля	Runtime audit engine (Runtime audit engine)

Сведения о событиях безопасности, которые необходимо фиксировать	
❖ Описание события безопасности	Runtime audit engine (Runtime audit engine)
❖ Сведения о критичности события безопасности	Runtime audit engine (Runtime audit engine)
Запись событий безопасности контейнеров в журнал событий безопасности	
❖ Обеспечение записи событий безопасности контейнеров в журнал событий безопасности информационной (автоматизированной) системы с указанием идентификатора пользователя хостовой операционной системы, от имени которого был запущен контейнер (должно осуществляться средством контейнеризации)	Log shipper agent (Log shipper)
Ролевой метод управления доступом с тремя ролями пользователей в Deckhouse	
❖ Разработчик образов контейнеров	User authz webhook (User authz), API server (Control Plane manager)
❖ Администратор информационной (автоматизированной) системы	User authz webhook (User authz), API server (Control Plane manager)
❖ Администратор безопасности Deckhouse	User authz webhook (User authz), API server (Control Plane manager)
Права разработчика образов контейнеров	
❖ Создавать, модифицировать и удалять образы контейнеров	User authz webhook (User authz), API server (Control Plane manager)
Права администратора информационной (автоматизированной) системы	
❖ Запускать и останавливать контейнеры	User authz webhook (User authz), API server (Control Plane manager)
Права администратора безопасности Deckhouse	

❖ Назначать права доступа пользователям Deckhouse к образам контейнеров	User authz webhook (User authz), API server (Control Plane manager)
❖ Иметь доступ на чтение к журналу событий безопасности средства контейнеризации	User authz webhook (User authz), API server (Control Plane manager)
❖ Формировать отчеты с учетом заданных критериев отбора, выгрузку (экспорт) данных из журнала событий безопасности Deckhouse	User authz webhook (User authz), API server (Control Plane manager)

3. Описание модулей и компонентов

3.1.Модуль Control Plane manager

3.1.1.Компонент Control Plane manager

Компонент Control Plane manager (CPM) отвечает за управление компонентами кластера. Он запускается на всех мастер-узлах кластера (узлах, имеющих метку node-role.kubernetes.io/control-plane: "").

3.1.2.Компонент etcd

Компонент etcd — это последовательно распределенное и высоконадежное хранилище ключевых значений, используемое в ПО Deckhouse Kubernetes Platform в качестве резервного хранилища для всех данных кластера.

3.1.3. Компонент API server

API server — это компонент ПО Deckhouse Kubernetes Platform, который предоставляет API для управления кластером. Он является главным компонентом управления кластером и обрабатывает все запросы к API server.

3.1.4. Компонент Controller manager

Controller manager — это демон, который встраивает основные контуры управления. Контроллер — это контур управления, который следит за общим состоянием кластера через API server и вносит изменения, пытаясь переместить текущее состояние в желаемое.

3.1.5. Компонент Scheduler

Scheduler — это компонент, который выбирает оптимальный узел для запуска вновь созданных или еще не запланированных подов.

3.1.6. Компонент Kubernetes API proxy

Kubernetes API proxy — это компонент ПО Deckhouse Kubernetes Platform, который обеспечивает связь между управляющей плоскостью Kubernetes API Server и рабочими узлами кластера. Он работает как прокси-сервер, который принимает запросы от клиентов и перенаправляет их на соответствующие узлы кластера.

3.2.Модуль Ingress-NGINX

3.2.1.Controller

Компонент Controller обеспечивает балансировку HTTP/HTTPS-трафика доступа. Компонент может быть запущен в отказоустойчивой конфигурации с несколькими репликами, а также в кластере возможно создание произвольного числа независимых экземпляров компонента Controller.

3.2.2.Controller failover
Компонент является резервным для компонента Controller — он принимает нагрузку (HTTP/HTTPS-трафик) в случае неработоспособности компонента Controller. Перенаправление трафика на компонент Controller failover выполняется благодаря настройкам, которые выполняет компонент Proxy failover.

3.2.3.Proxy failover
Компонент управляет правилами брандмауэра для перенаправления трафика на компонент Controller failover.

3.2.4.Kruise controller manager
Компонент Kruise controller manager управляет обновлением компонента Controller. Задача компонента Kruise controller manager — обеспечить обновление компонента Controller на новую версию с минимальным временем недоступности.

3.3.Модуль Node Manager

3.3.1.Bashible apiserver
Bashible apiserver — это сервер API, который используется в инструменте Bashible для управления конфигурацией и автоматизации развертывания инфраструктуры.

3.3.2.Bashible
Bashible — это компонент для автоматизации установки и настройки приложений на базе Bash-скриптов.

3.3.3.containerd
containerd — это компонент, реализующий исполняемую среду для запуска контейнеров. Компонент управляет всем жизненным циклом контейнера: от получения и хранения образа до запуска контейнера через runC и контролирования его работы.

3.3.4.crictl
crictl — это компонент для CRI-совместимых режимов выполнения контейнеров. Компонент используется для общения с containerd и другими исполняемыми средами для контейнеров, соответствующими CRI, с целью выявления проблем на узле.

3.3.5.curl
Компонент используется для взаимодействия с API server по протоколу HTTPS при обновлении объекта узла кластера на этапе его первичной настройки.

3.3.6.jq
Компонент jq — это «фильтр»: принимает входные данные и выдает выходные. Существует множество встроенных фильтров для извлечения определенного поля объекта, преобразования числа в строку и других стандартных задач.

3.3.7.kubectl
Функцией данного компонента является предоставление пользователю ПО Deckhouse Kubernetes Platform CLI для управления кластером.

3.3.8.kubelet

kubelet выполняется непосредственно на рабочих узлах, где работают фактические контейнеры подов.

3.3.9.Kubernetes CNI

Kubernetes CNI — это компонент, который позволяет плагинам сети взаимодействовать с Kubernetes. CNI предоставляет возможность создавать и управлять сетевыми интерфейсами контейнеров, а также устанавливать правила маршрутизации и настройки сетевых политик.

3.3.10.Toml-merger

Toml-merger — это компонент, отвечающий за объединение нескольких файлов формата TOML (Tom's Obvious, Minimal Language) в один файл.

3.4.Модуль Kube DNS

3.4.1.CoreDNS

Компонент CoreDNS — это DNS-сервер, который используется в кластерах Kubernetes для обеспечения разрешения имен хостов внутри кластера. CoreDNS может быть настроен для использования различных источников данных, таких как файлы, базы данных, сервисы облачных провайдеров и т. д.

3.5.Модуль kube-proxy

3.5.1.Компонент kube-proxy

Функцией данного компонента является обеспечение сетевого взаимодействия между клиентскими подами и подами, реализующими какой-либо сервис. Поскольку поды могут создаваться, удаляться и перемещаться между узлами кластера, создается объект Service (служба) со стабильным IP-адресом, хранящий данные о всех подах, реализующих сервис, их IP-адресах и портах.

3.6.Модуль CNI Flannel

3.6.1.Flannel

Модуль CNI Flannel обеспечивает работу сети в кластере с помощью компонента Flannel. Компонент Flannel запускает небольшой одиночный двоичный агент под названием flanneld на каждом хосте и отвечает за распределение аренды подсети для каждого хоста из большего, предварительно сконфигурированного адресного пространства.

3.7.Модуль Deckhouse

3.7.1.Deckhouse

Базовый компонент, без которого работа ПО Deckhouse Kubernetes Platform невозможна. Компонент выполняет следующие задачи:

- настраивает уровень журналирования событий (Debug, Info, Error);

- включает и отключает модули согласно конфигурации;
- обновляет версии компонентов. При обновлении учитываются требования к окружению (например, версии ядра, версия Kubernetes, версии отдельных компонентов), выполняются необходимые миграции.

3.7.2.Webhook handler

Компонент Webhook handler обрабатывает входящие webhooks (HTTP POST запросы) от внешних сервисов и выполняет определенные действия на основе полученных данных.

3.8.Модуль Monitoring Kubernetes

3.8.1.Kube state metrics

Этот компонент, который прослушивает сервер Kubernetes API и генерирует метрики о состоянии объектов.

3.8.2.Node exporter

Компонент Node exporter собирает данные о состоянии аппаратных компонентов узла и параметрах операционной системы.

3.9.Модуль Monitoring Kubernetes Control Plane

3.9.1.Control Plane proxy

Компонент отвечает за мониторинг уровня управления Kubernetes. Он собирает метрики и предоставляет базовый набор правил для мониторинга

3.10.Модуль Operator Prometheus

3.10.1.Prometheus config reloader

Компонент Prometheus config reloader следит за изменениями конфигурационного файла prometheus.yaml. При изменении файла Prometheus config reloader по HTTP отправляет Prometheus запрос на перезагрузку.

3.10.2.Prometheus operator

Prometheus operator обеспечивает развертывание и управление системы мониторинга серверов и программ Prometheus и связанных с ним компонентов мониторинга на базе Kubernetes.

3.11.Модуль Loki

3.11.1.Loki

Компонент Loki — это горизонтально масштабируемая, высокодоступная, многопользовательская система хранения журналов, которая индексирует не содержимое журналов, а набор меток для каждого потока журналов.

3.12.Модуль Log shipper

3.12.1.Log shipper agent

Компонент Log shipper agent собирает журналы и отправляет их в систему хранения журналов. Конфигурируется с помощью ресурсов ClusterLoggingConfig, ClusterLogsDestination и PodLoggingConfig.

3.13.Модуль Prometheus

3.13.1.Alertmanager

Alertmanager обрабатывает оповещения, отправленные клиентскими приложениями, такими как сервер Prometheus.

3.13.2.Grafana

Управляемая платформа визуализации данных. Включает подготовленные dashboard'ы для всех модулей Deckhouse.

3.13.3.Prometheus main

Основной компонент оперативного мониторинга. Выполняет сбор метрик мониторинга за указанный интервал времени.

3.13.4.Prometheus longterm

Дополнительный компонент мониторинга. Предназначен для мониторинга за больший период, чем у компонента Prometheus main, но с меньшей детализацией.

3.13.5.Trickster

Компонент выполняет кеширование запросов, снижающее нагрузку на Prometheus main и Prometheus longterm.

3.14.Модуль Prometheus metrics adapter

3.14.1.Prometheus metrics adapter

Данный компонент регистрирует k8s-prometheus-adapter в качестве external API-сервиса, который расширяет возможности Kubernetes API.

3.15.Модуль User authz

3.15.1.User authz webhook

User authz webhook — это webhook, который настраивается в API server через параметр authorization-webhook-config-file и реализует multitenancy в кластере.

3.16.Модуль Operator Trivy

3.16.1.Operator

Компонент настраивает работу сканера уязвимостей в кластере. Обеспечивает создание в кластере периодических заданий (Job), запускающих сканирование.

3.16.2.Node collector

Node collector — это компонент для сбора информации о системе и ее компонентах. Используется для мониторинга производительности, отслеживания ошибок и сбора метрик.

3.16.3.Scan vulnerability report

Компонент позволяет запускать периодическое сканирование на уязвимости. Базируется на проекте Trivy.

3.16.4BDU Exporter

BDU Exporter — это экспортер, который, используя результаты сканирования Trivy, добавляет к Trivy информацию о найденных совпадениях в БДУ ФСТЭК России.

3.17.Модуль Admission policy engine

3.17.1.Gatekeeper audit

Компонент периодически (каждые 60 секунд) проверяет объекты кластера Kubernetes на соответствие установленным политикам безопасности.

3.17.2.Gatekeeper controller manager

Gatekeeper controller manager (GCM) — это компонент, который управляет процессом установки и обновления политик безопасности в Kubernetes.

3.18.Модуль User authn

3.18.1.Dex

Dex — это компонент аутентификации и авторизации, который позволяет использовать протоколы OpenID Connect и OAuth 2.0 для управления доступом к ресурсам.

3.19.Модуль Runtime audit engine

3.19.1.Runtime audit engine

Компонент предназначен для поиска угроз безопасности. Он собирает события ядра Linux, итоги аудита API Kubernetes, обогащает их метаданными о подах Kubernetes и генерирует события аудита безопасности по установленным правилам.

Заключение

После получения сертификата соответствия мы будем расширять функциональность Certified Security Edition, добавляя в эту редакцию модули, которые уже есть в Deckhouse Enterprise Edition.

Кроме того, мы планируем получить сертификат соответствия процедур безопасной разработки программного обеспечения. Сертификат подтверждает, что при дальнейшем развитии платформы команда разработки Deckhouse соблюдает требования в области защиты информации, установленные ФСТЭК России. Это позволит разработчикам Deckhouse добавлять в редакцию Certified Security Edition новые модули и фичи без повторного привлечения испытательной лаборатории, что ускорит поставку новой функциональности платформы.