



Deckhouse Airlock

Kubernetes-native система управления
привилегированным доступом

Почему классического РАМ уже недостаточно?

Текущие RAM-решения не созданы для Kubernetes

! Доступ на уровне узлов или сервера

Не видно, к какому сервису подключался пользователь

👤 Нет гранулярного управления доступом

Нельзя гибко настраивать доступ к управлению определенными примитивами Kubernetes

🛡️ Взаимодействие на уровне пользователей

Классические RAM в первую очередь проектировались для управления привилегированным доступом для людей, а не приложений и сервисов

Пример

Мы **не знаем**, кто и когда подключался к базе данных в production

Решение есть! Deckhouse Airlock

Deckhouse Airlock — система для управления привилегированным доступом.

Ключевое преимущество — возможность работать на уровне сервисов Kubernetes, а не на уровне узлов или сервера.

Teleport CE

Проверенное open source
ядро

K8s-native

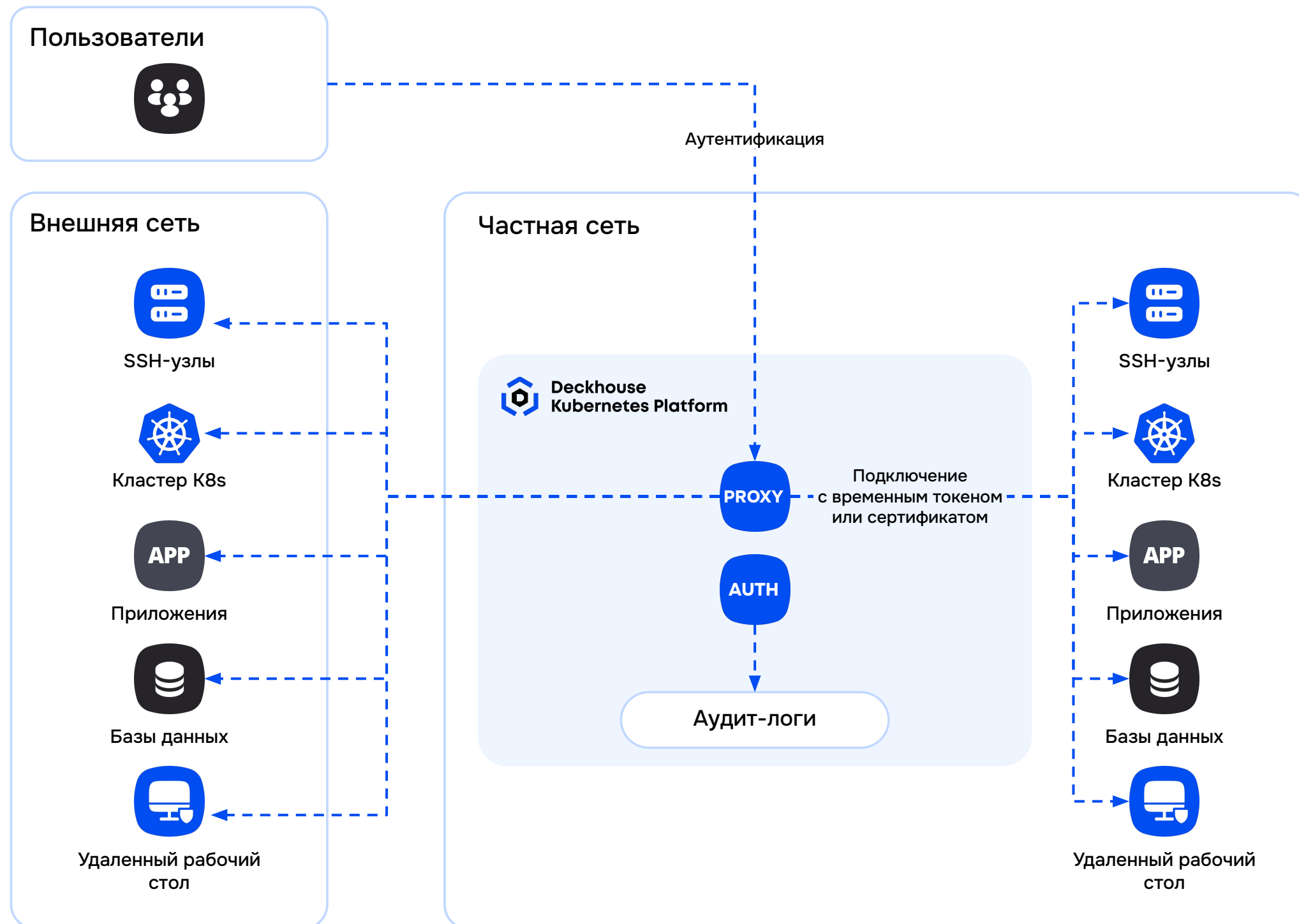
CRD, операторы, kube API
«из коробки»

Service based

Доступ на уровне сервисов,
а не узлов или сервера

а также надежная поддержка вендора, дорожная карта развития,
интеграция с экосистемой Deckhouse и многое другое...

Как работает Deckhouse Airlock



- Пользователь проходит аутентификацию в Deckhouse Airlock через SSO, MFA или другой подключённый identity-provider
- Deckhouse Airlock выдаёт короткоживущий сертификат или токен доступа вместо постоянных логинов и паролей
- Пользователь подключается к нужному ресурсу: Kubernetes-кластеру, серверу, базе данных или внутреннему приложению через прокси
- Deckhouse Airlock применяет политики доступа: проверяет роль, время, контекст, устройство и другие условия
- Сессия логируется и может записываться для аудита и расследований
- После завершения сессии доступ исчезает, потому что он был временным, а не постоянным

Deckhouse Airlock vs классические PAM

	Deckhouse Airlock	Классический PAM
Just-in-time доступ	✓	✓
Встроенная запись и воспроизведение сессий	✓	✓
Краткоживущие сертификаты	✓	✗
Нативный доступ к кластерам Kubernetes	✓	✗
Работа с kubectl без лишних надстроек	✓	✗
Отказ от постоянных статических учётных данных	✓	✗
Детализированный аудит действий в кластере Kubernetes	✓	✗
Отказ от обязательного VPN для части сценариев	✓	✗
Простая интеграция с DevOps-инструментами и процессами	✓	✗
Декларативное управление конфигурациями (IaC)	✓	✗

Примеры использования

Получение JIT-доступа

Deckhouse

Resources

Resources

Access Requests

Active Sessions

Access Requests

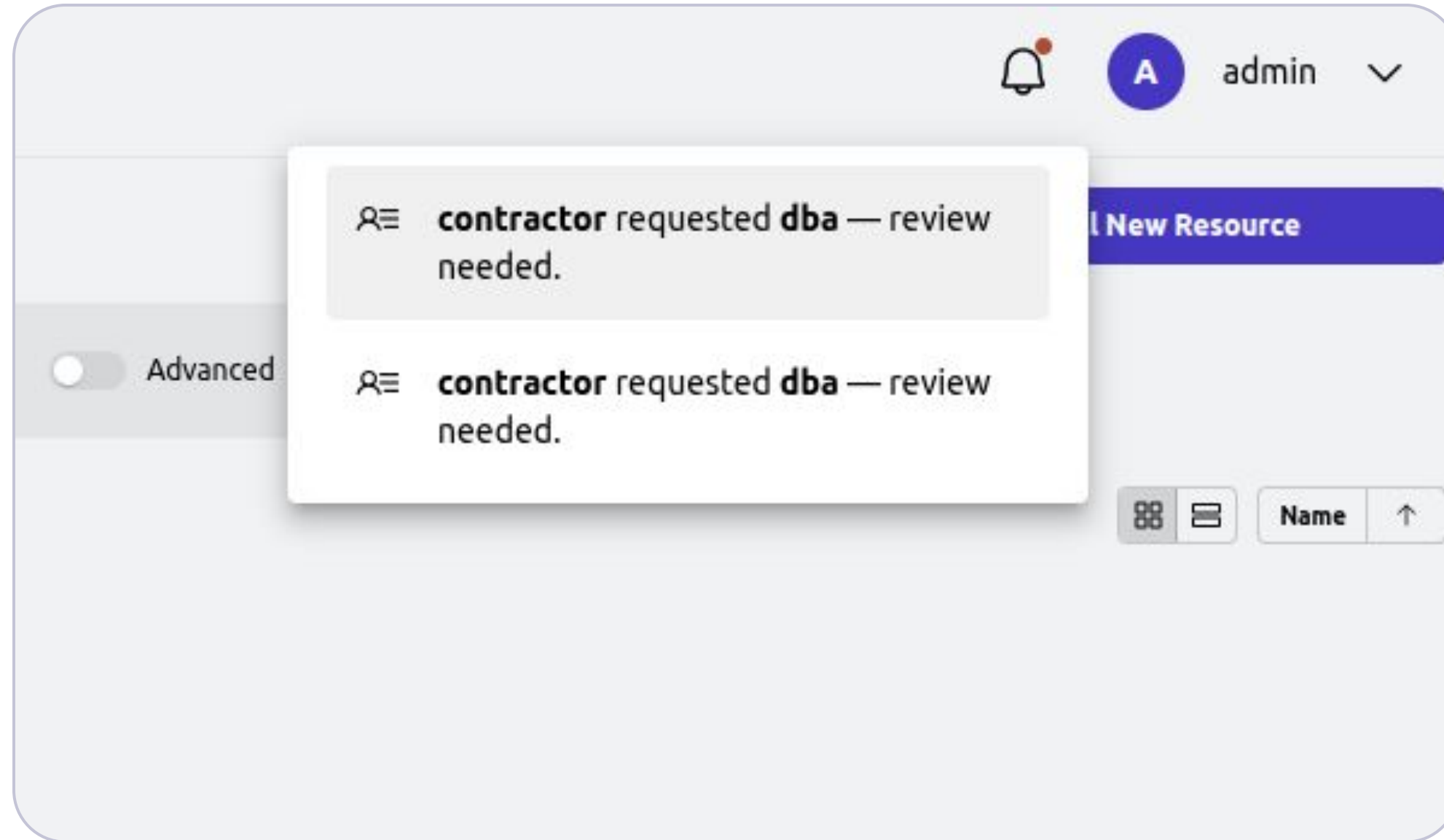
SWITCH BACKNEW REQUESTREFRESH

ALLPENDINGAPPROVEDDENIED

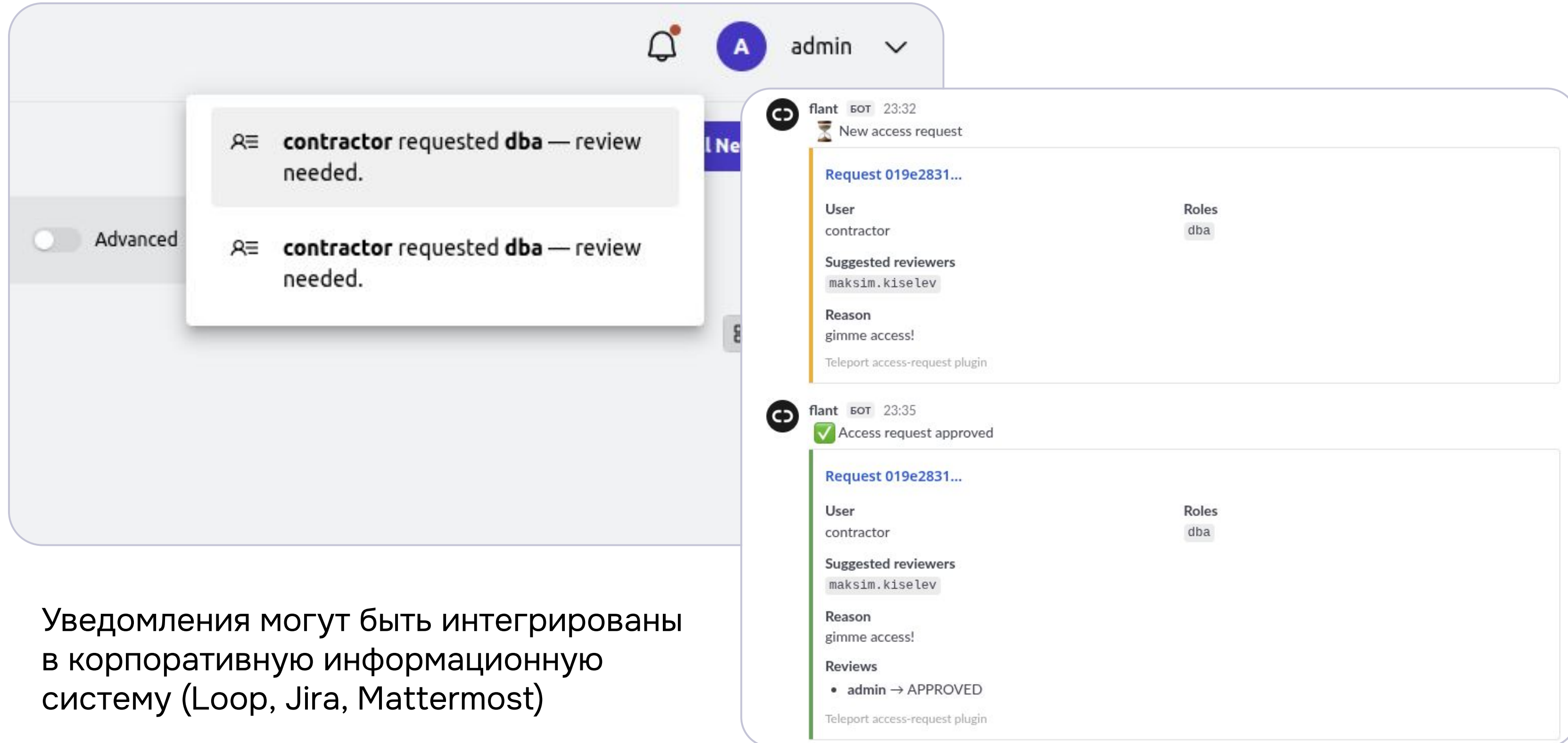
SEARCH...

ID	USER	STATE	ROLES	REASON	CREATED	EXPIRES	ACTIONS
019e279a...	contractor	Approved	dba	gimme	2026-05-14 20:47:54	2026-05-14 21:47:54	ASSUME ROLESDELETE
019e279c...	contractor	Approved	access, editor	gimme access and editor	2026-05-14 20:50:37	2026-05-15 08:50:13	ActiveDELETE

Уведомления о запросе доступа



Уведомления о запросе доступа



The image displays a user interface for managing access requests. On the left, a notification panel shows two identical alerts: "contractor requested dba — review needed." Each alert includes a search icon and a toggle switch labeled "Advanced".

On the right, a chat window shows two messages from the bot "flant" at 23:32 and 23:35. The first message, "New access request", details a request for role "dba" by user "contractor", suggested reviewer "maksim.kiselev", and reason "gimme access!". The second message, "Access request approved", shows the same request with a green checkmark and a review entry "admin → APPROVED".

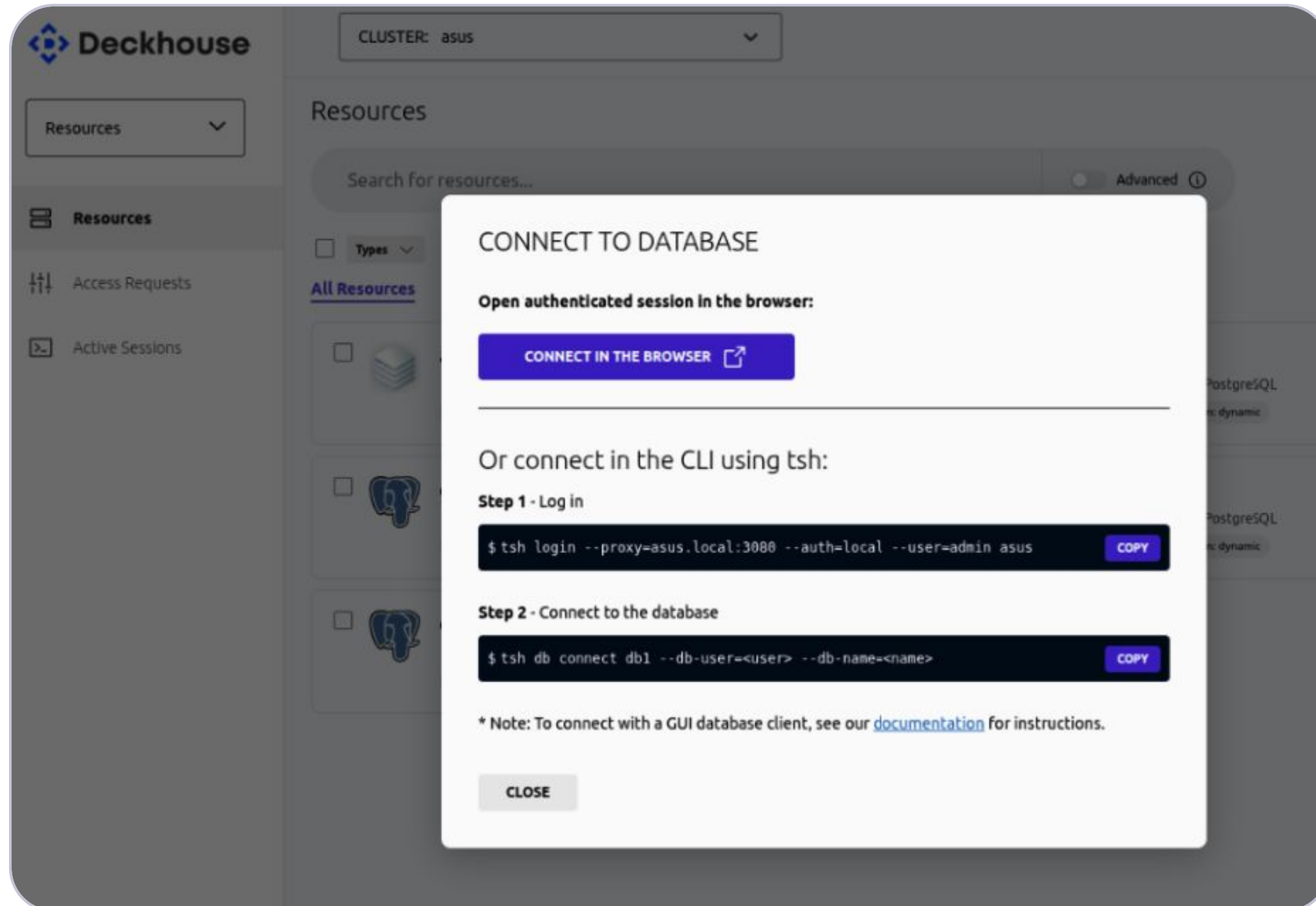
Both messages include a table with the following data:

User	Roles
contractor	dba

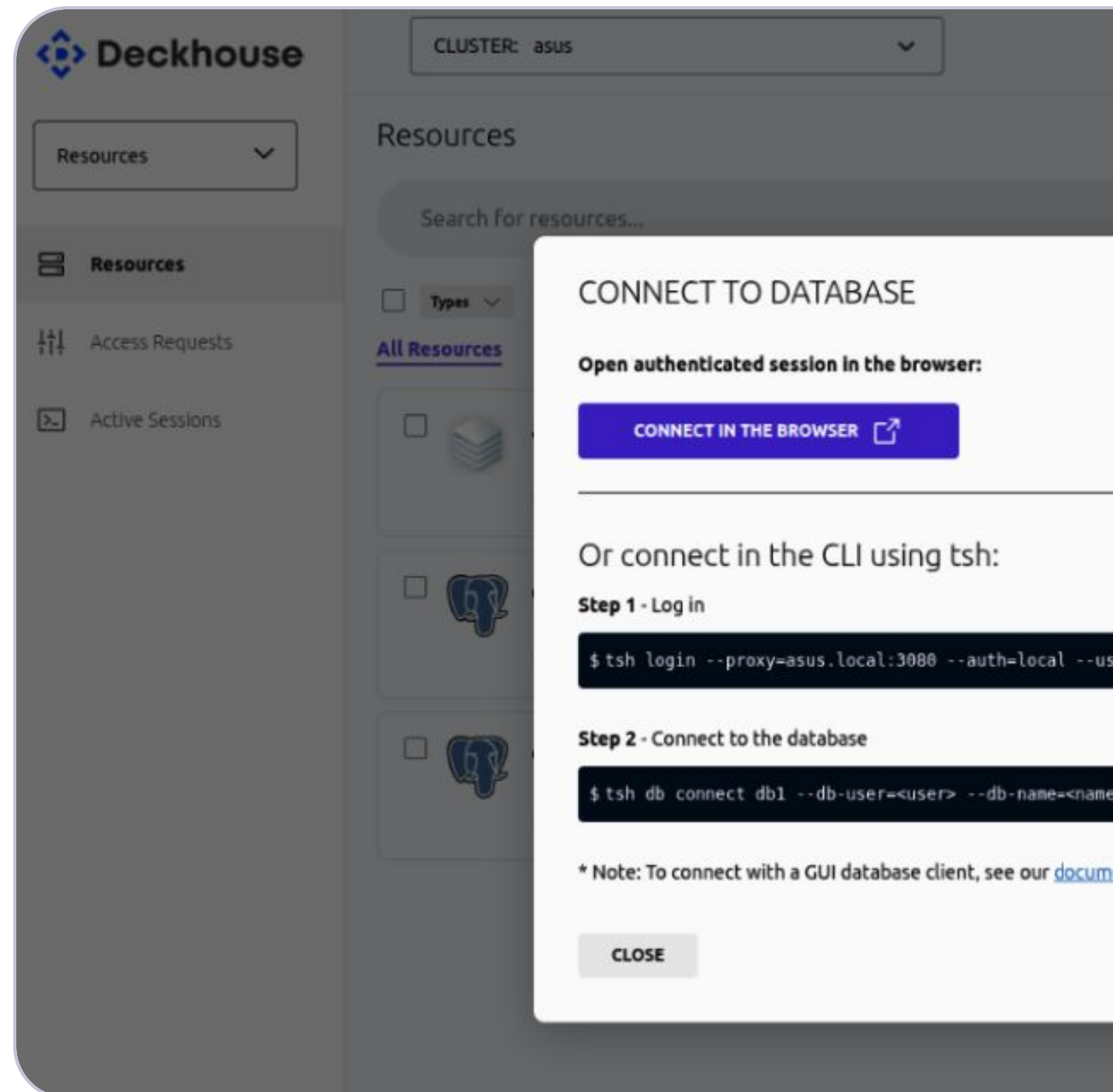
Additional details in the messages include "Request 019e2831...", "Suggested reviewers: maksim.kiselev", "Reason: gimme access!", and "Reviews: admin → APPROVED". The interface also shows a user profile for "admin" at the top right.

Уведомления могут быть интегрированы в корпоративную информационную систему (Loop, Jira, Mattermost)

Мастер настройки подключения



Мастер настройки подключения



CONNECT TO DATABASE

Database name

postgres

Database user

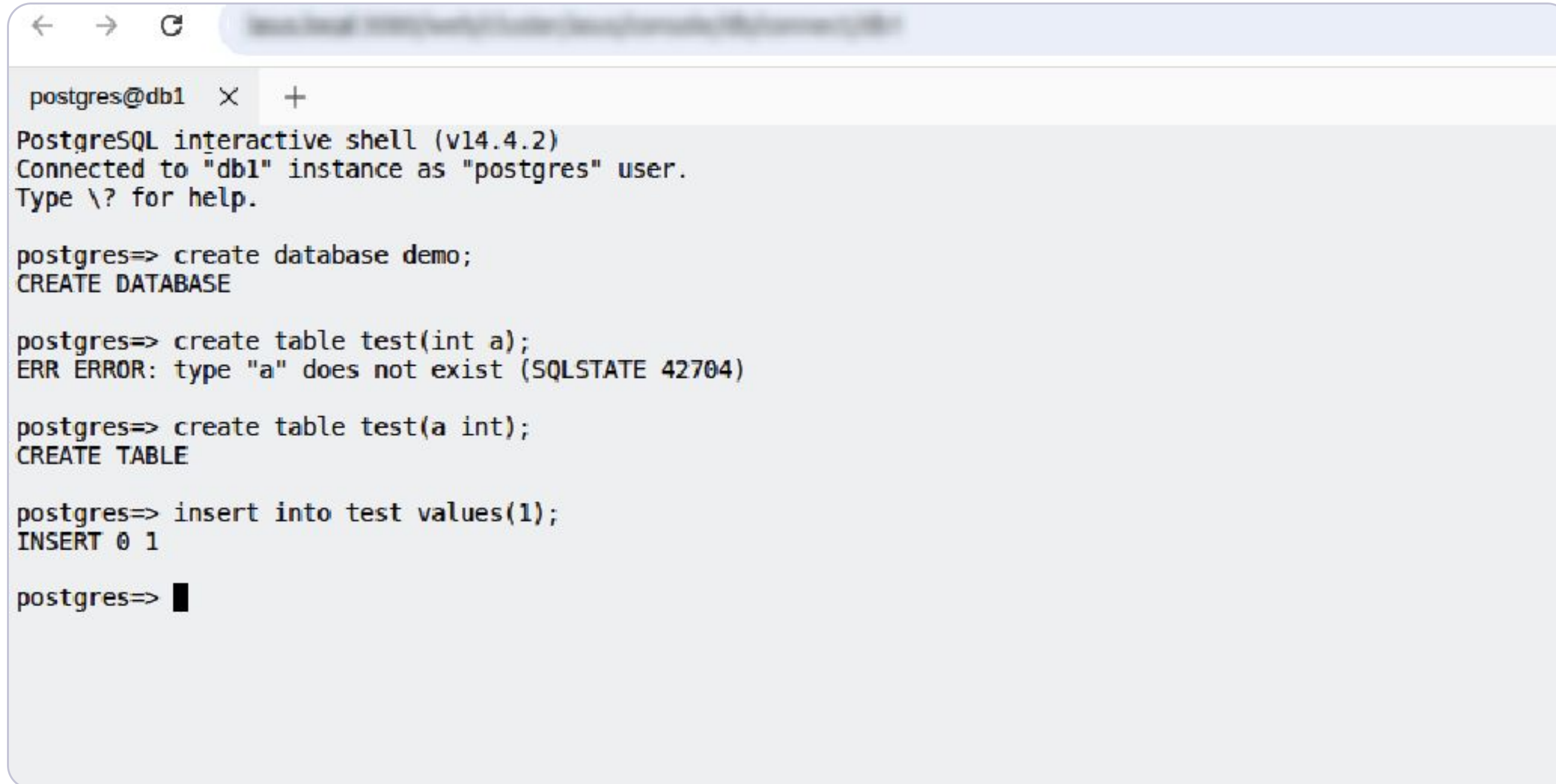
postgres

CLOSE

CONNECT

Доступно подключение к базам данных из внешних инструментов и напрямую из веб-браузера

Подключение к БД через веб-браузер



The screenshot shows a web browser window with a single tab titled "postgres@db1". The browser's address bar is empty. The main content area displays a PostgreSQL interactive shell (v14.4.2) interface. The text in the terminal is as follows:

```
postgres@db1 X +
PostgreSQL interactive shell (v14.4.2)
Connected to "db1" instance as "postgres" user.
Type \? for help.

postgres=> create database demo;
CREATE DATABASE

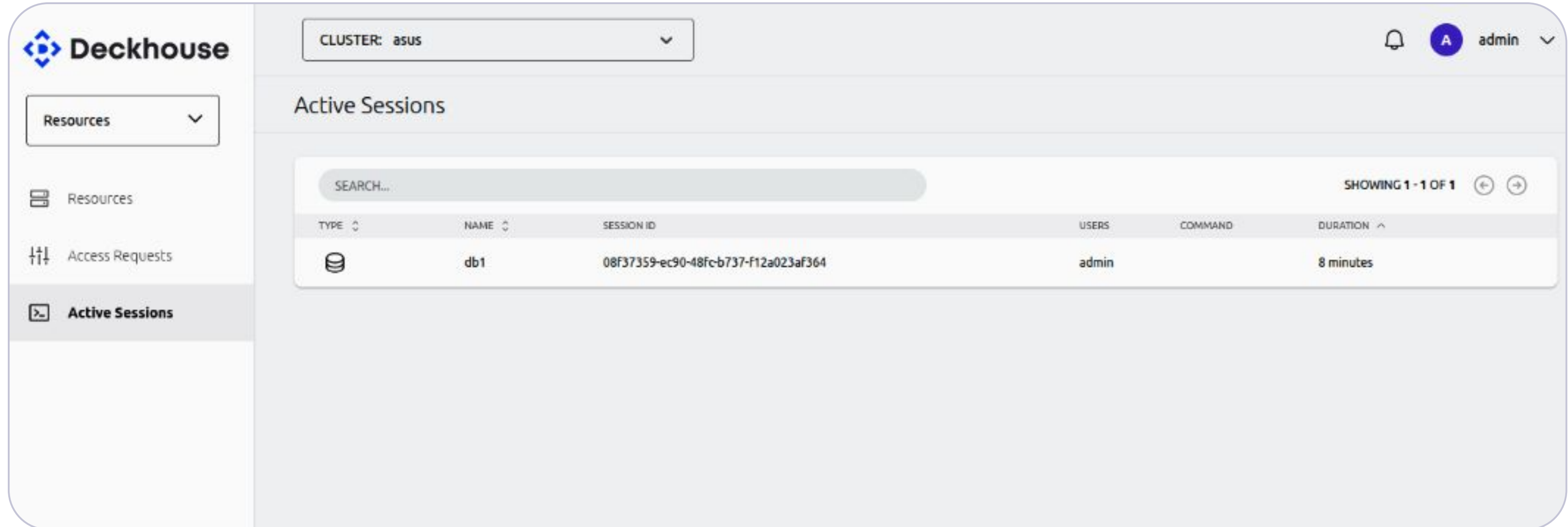
postgres=> create table test(int a);
ERR ERROR: type "a" does not exist (SQLSTATE 42704)

postgres=> create table test(a int);
CREATE TABLE

postgres=> insert into test values(1);
INSERT 0 1

postgres=> █
```

Мониторинг активных сессий

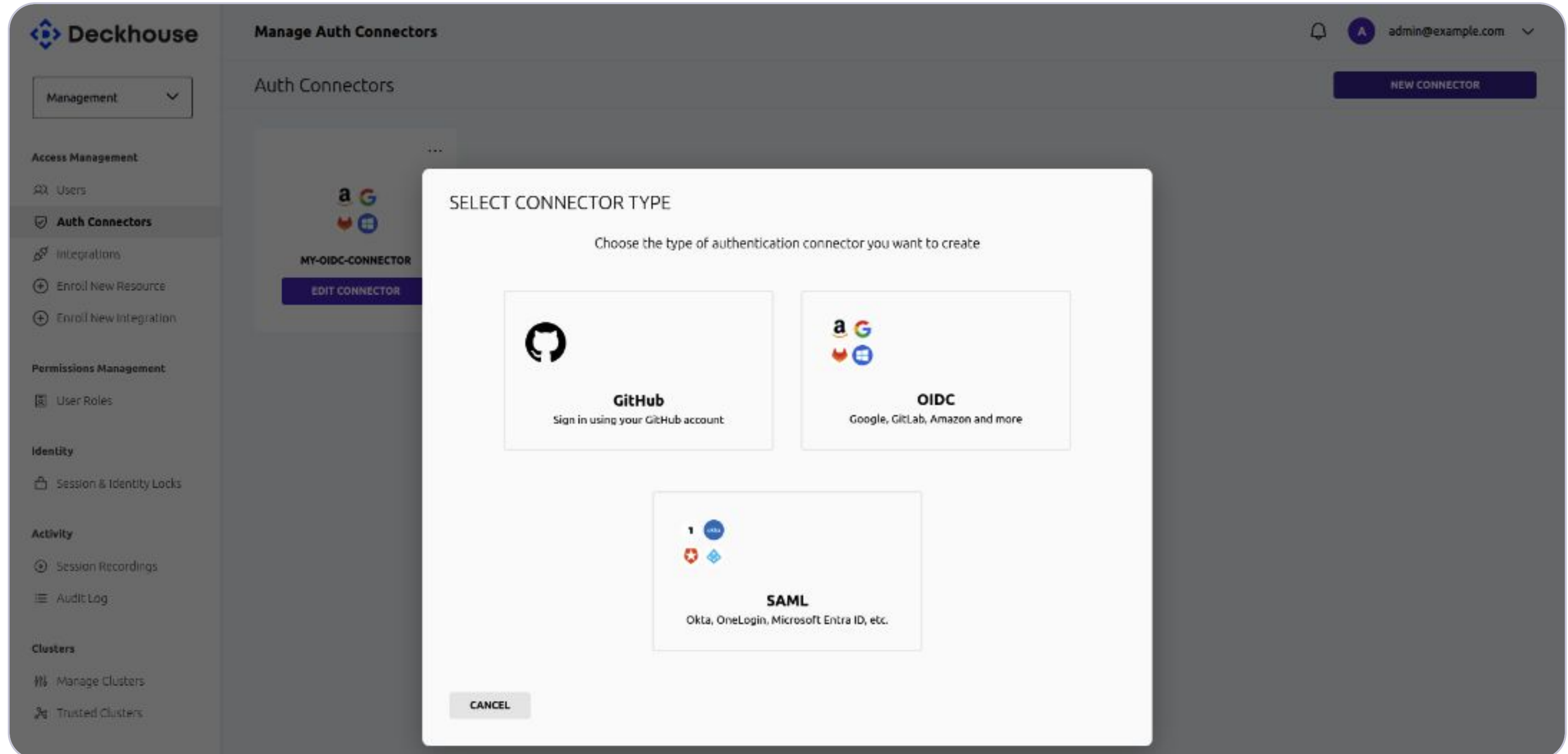


The screenshot shows the Deckhouse web interface for monitoring active sessions. The left sidebar contains the Deckhouse logo and navigation links for Resources, Access Requests, and Active Sessions. The main content area displays the 'Active Sessions' page for the 'asus' cluster. It includes a search bar, a table with session details, and pagination controls.

TYPE	NAME	SESSION ID	USERS	COMMAND	DURATION
db	db1	08F37359-ec90-48fc-b737-f12a023af364	admin		8 minutes

Администратор может посмотреть список активных сессий и прерывать их при необходимости

Поддержка интеграции с внешними IDP



Управление ресурсами IaC (интеграция с IDP)

The screenshot displays the Deckhouse management interface. The left sidebar contains navigation links for Management, Access Management (Users, Auth Connectors, Integrations), Permissions Management (User Roles), Identity (Session & Identity Locks), Activity (Session Recordings, Audit Log), and Clusters (Manage Clusters, Trusted Clusters). The main panel is titled 'Manage Auth Connectors' and shows a 'NEW CONNECTOR' button. A modal window titled 'CREATING A NEW SAML CONNECTOR' is open, displaying a YAML specification for a SAML connector.

CREATING A NEW SAML CONNECTOR

Spec

```
1 kind: saml
2 version: v2
3 metadata:
4   name: new_saml_connector
5 spec:
6   # connector display name that will be appended to the title of "Login with"
7   # button on the cluster login screen so it will say "Login with <display>"
8   display: SAML
9   # SAML provider will make a callback to this URL after successful authentication
10  # cluster-url is the address the cluster UI is reachable at
11  # The last segment of the URL must be identical to the connector metadata name.
12  acs: 'https://<cluster-url>/v1/webapi/saml/acs/new_saml_connector'
13  # Controls whether IdP-initiated SSO is allowed. If false, all such requests will be rejected with an error.
14  allow_idp_initiated: false
15  # Mapping of SAML attributes to Teleport roles
16  attributes_to_roles:
17    - name: groups
18      value: admins
19      roles:
20        - 'access'
21        - 'editor'
22  # Provides a path to the IdP metadata.
23  entity_descriptor_url: https://<saml-provider-url>/metadata
24
25
```

At the bottom of the modal are two buttons: 'SAVE CHANGES' and 'CANCEL'.

Управление ресурсами IaC (управление ролями)

The screenshot displays the Deckhouse web interface for managing authentication connectors. The main header shows the Deckhouse logo and the title 'Manage Auth Connectors'. The user is logged in as 'admin@example.com'. A sidebar on the left contains navigation links for 'Management', 'Access Management', 'Auth Connectors', 'Integrations', 'Permissions Management', 'Identity', 'Activity', and 'Clusters'. The 'Auth Connectors' section is currently active, showing a list of connectors and a 'NEW CONNECTOR' button. A modal window titled 'EDIT ROLE' is open, showing the configuration for the 'approver' role. The modal contains a 'Spec' field with a YAML configuration and a 'SETUP INSTRUCTIONS' section with a warning about YAML formatting. At the bottom of the modal are buttons for 'SAVE CHANGES', 'CANCEL', and 'VIEW DOCUMENTATION'.

Deckhouse Manage Auth Connectors

Auth Connectors

NEW CONNECTOR

EDIT ROLE

approver

Spec

```
1 kind: role
2 metadata:
3   id: 3778854187700281147
4   name: approver
5 spec:
6   allow:
7     review_requests:
8       roles:
9         - kube
10        - editor
11        - access
12        - ssh
13   deny: {}
14   options:
15     cert_format: standard
16     create_db_user: false
17     create_desktop_user: false
18     desktop_clipboard: true
19     desktop_directory_sharing: true
20     enhanced_recording:
21       - command
22       - network
23     forward_agent: false
24     idp:
25       saml:
26         enabled: true
27         max_session_ttl: 30h0m0s
28         pin_source_ip: false
29         port_forwarding: true
30         record_session:
31           default: best_effort
32           desktop: true
33         ssh_file_copy: true
34   version: v5
```

SETUP INSTRUCTIONS

WARNING Roles are defined using [YAML format](#). YAML is sensitive to white space, so please be careful.

SAVE CHANGES CANCEL VIEW DOCUMENTATION

Возможность управлять ресурсами через Kubernetes-примитивы

```
root@demo-master-0: ~
root@demo-master-0: ~ 156x36
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: resources.teleport.dev/v3
kind: TeleportOIDCConnector
metadata:
  creationTimestamp: "2026-05-03T12:22:19Z"
  finalizers:
  - resources.teleport.dev/deletion
  generation: 9
  name: my-oidc-connector
  namespace: d8-teleport
  resourceVersion: "473307173"
  uid: 77e94c91-e832-4e33-8328-afb3e5545fb6
spec:
  allow_unverified_email: true
  claims_to_roles:
  - claim: groups
    roles:
    - access
    - editor
    - auditor
    - approver
    - ssh
    - allow-session-join
    - access-request-admin
    - reviewer
    - apps-editor
    value: admins
  - claim: groups
    roles:
    - users
    value: users
  client_id: dex-client-teleport@d8-teleport
```

1,1 Top

Управление разрешениями роли на доступ к ресурсам Kubernetes

EDIT ROLE

kube-access

Spec

```
1 kind: role
2 metadata:
3   id: 1780397135770659451
4   name: kube-access
5 spec:
6   allow:
7     kubernetes_groups:
8     - viewers
9     kubernetes_labels:
10    - '*'
11    kubernetes_resources:
12    - kind: '*'
13      name: '*'
14      namespace: '*'
15      verbs:
16      - get
17      - list
18      - watch
19   deny: {}
20   options:
21     cert_format: standard
22     create_db_user: false
23     create_desktop_user: false
24     desktop_clipboard: true
25     desktop_directory_sharing: true
26     enhanced_recording:
27     - command
28     - network
29     forward_agent: false
30     idp:
31     - saml:
32       enabled: true
33     max_session_ttl: 30h0m0s
34     pin_source_ip: false
35     port_forwarding: true
36     record_session:
37     - default: best_effort
38     - desktop: true
39     ssh_file_copy: true
40 version: v7
41
```

Управление разрешениями роли на доступ к ресурсам Kubernetes

EDIT ROLE

kube-access

Spec

```
1 kind: role
2 metadata:
3   id: 1780397135770659451
4   name: kube-access
5 spec:
6   allow:
7     kubernetes_groups:
8     - viewers
9     kubernetes_labels:
10    | '*': '*'
11    kubernetes_resources:
12    - kind: '*'
13      name: '*'
14      namespace: '*'
15      verbs:
16      - get
17      - list
18      - watch
19   deny: {}
20   options:
21     cert_format: standard
22     create_db_user: false
23     create_desktop_user: false
24     desktop_clipboard: true
25     desktop_directory_sharing: true
26     enhanced_recording:
27     - command
28     - network
29     forward_agent: false
30     idp:
31     | saml:
32     |   enabled: true
33     max_session_ttl: 30h0m0s
34     pin_source_ip: false
35     port_forwarding: true
36     record_session:
37     | default: best_effort
38     | desktop: true
39     ssh_file_copy: true
40   version: v7
41
```

```
1 kind: role
2 metadata:
3   id: 1780397135770659451
4   name: kube-access
5 spec:
6   allow:
7     kubernetes_groups:
8     - viewers
9     kubernetes_labels:
10    | '*': '*'
11    kubernetes_resources:
12    - kind: '*'
13      name: '*'
14      namespace: '*'
15      verbs:
16      - get
17      - list
18      - watch
```

Логирование операций, выполняемых в кластере Kubernetes

```
max@asus:~$ tsh kube login api.stronghold-demo.flant.dev
Logged into Kubernetes cluster "api.stronghold-demo.flant.dev".
```

Your Teleport cluster runs behind a layer 7 load balancer or reverse proxy.

To access the cluster, use "tsh kubectl" which is a fully featured "kubectl" command that works when the Teleport cluster is behind layer 7 load balancer or reverse proxy. To run the Kubernetes client, use:

```
tsh kubectl version
```

Or, start a local proxy with "tsh proxy kube" and use the kubeconfig provided by the local proxy with your native Kubernetes clients:

```
tsh proxy kube -p 8443
```

Learn more at <https://goteleport.com/docs/architecture/tls-routing/#working-with-layer-preview>

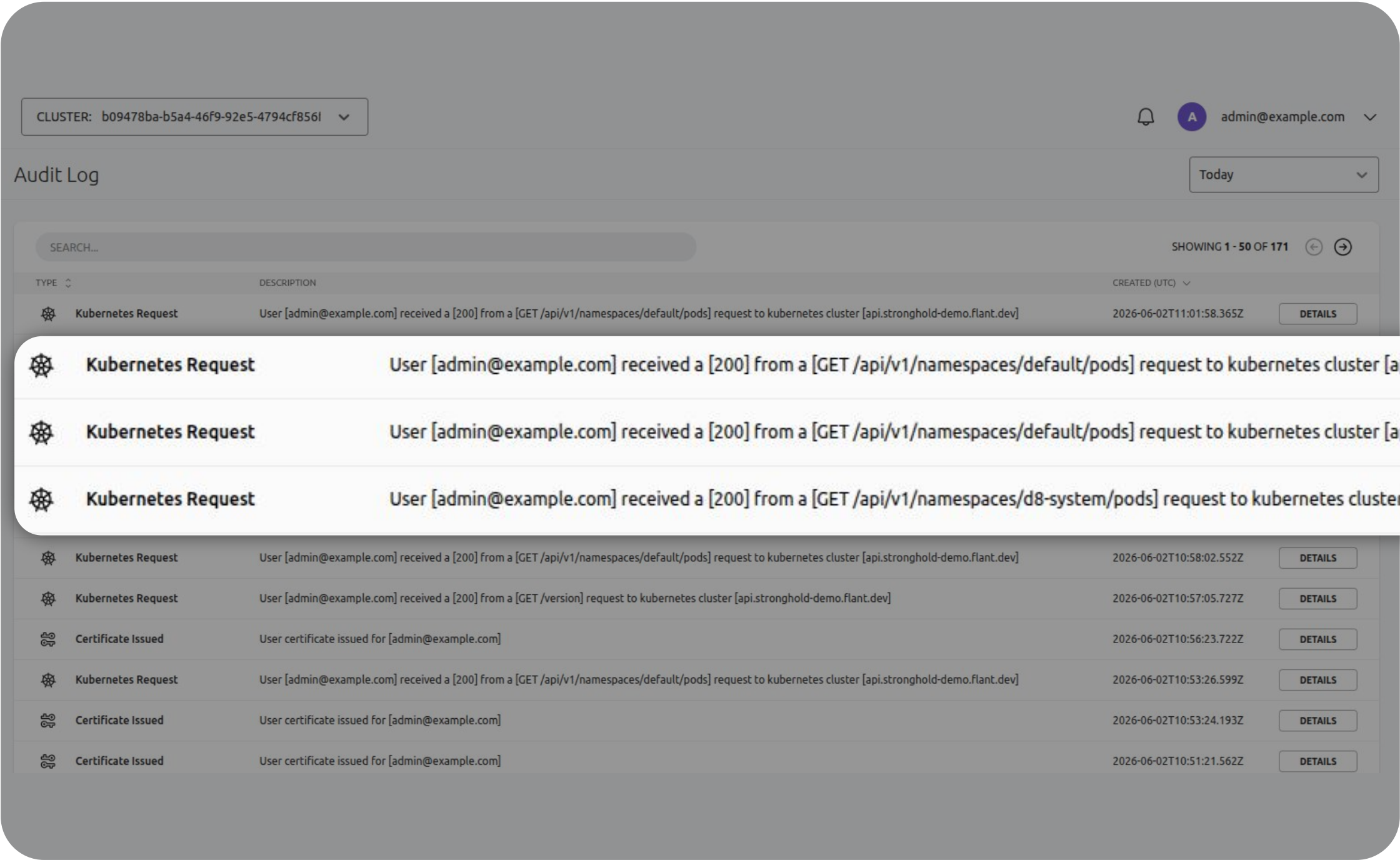
- Настраиваем клиента для подключения к Kubernetes

Логирование операций, выполняемых в кластере Kubernetes

```
max@asus:~$ tsh kubectl get po
NAME                                READY   STATUS    RESTARTS   AGE
clickhouse-7dc7f6b968-mcfjh        1/1     Running   0           137m
docs-967b76d97-jfbht               1/1     Running   0           137m
keycloak-0                         2/2     Running   0           7h3m
mongodb-576cbbdd67-gcscb           1/1     Running   0           137m
nginx-tofu-6d7cbc58c-gwk9k         1/1     Running   0           137m
openldap-6c995cbf8d-gg8ft          1/1     Running   0           137m
password-58ffb567b8-6v8q9          1/1     Running   0           137m
registry-0                         1/1     Running   0           137m
registry-6d54bb5c4d-69k8g          1/1     Running   0           137m
```

- Настраиваем клиента для подключения к Kubernetes
- Выполняем запрос на список подов

Логирование операций, выполняемых в кластере Kubernetes



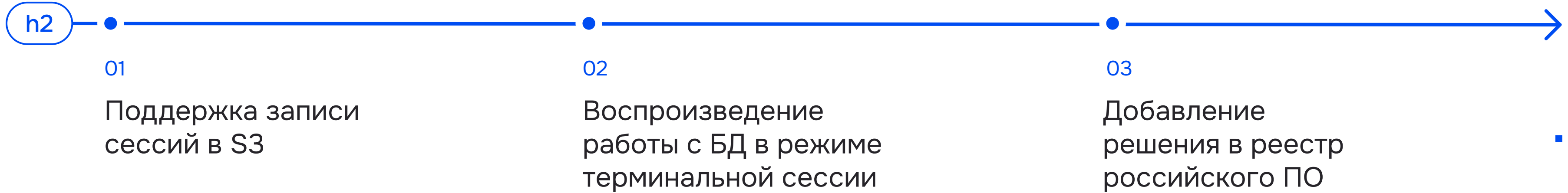
- Настраиваем клиента для подключения к Kubernetes
- Выполняем запрос на список подов
- Видим запрос в аудит-логах

Блокирование неразрешенных операций, выполняемых в кластере Kubernetes

```
max@asus:~$ tsh kubectl delete po nginx-66686b6766-fw8kq -n test
Error from server (Forbidden): pods "nginx-66686b6766-fw8kq" is forbidden: User "kube" cannot delete resource "pods" in
espace "test"
Ask your Teleport admin to ensure that your Teleport role includes access to the pod in "kubernetes_resources" field.
Check by running: kubectl auth can-i delete pods/nginx-66686b6766-fw8kq --namespace test
```

- Настраиваем клиента для подключения к Kubernetes
- Выполняем запрос на список подов
- Видим запрос в аудит-логах
- При выполнении операций, доступ на выполнение которых не предоставлен, происходит блокирование

Roadmap 2026



Интеграции «из коробки»

CI/CD, инфраструктура и мониторинг

Ansible

ELK

GitLab

Jenkins

Kubernetes

Базы данных

MySQL

MariaDB

Redis

PostgreSQL

MongoDB

Elasticsearch

CockroachDB

Системы аутентификации

OIDC

SAML

FIDO2

Keycloak

Local users

Будем рады
ОТВЕТИТЬ на ваши
вопросы!



[Telegram](#)



[RuTube](#)



[Блог](#)

 contact@deckhouse.ru 

 +7 (495) 721-10-27

 deckhouse.ru 