Deckhouse
ФЛАНТ

TRON ASOC

Безопасность секретов и соответствие РБПО с TRON ASOC и Deckhouse Stronghold

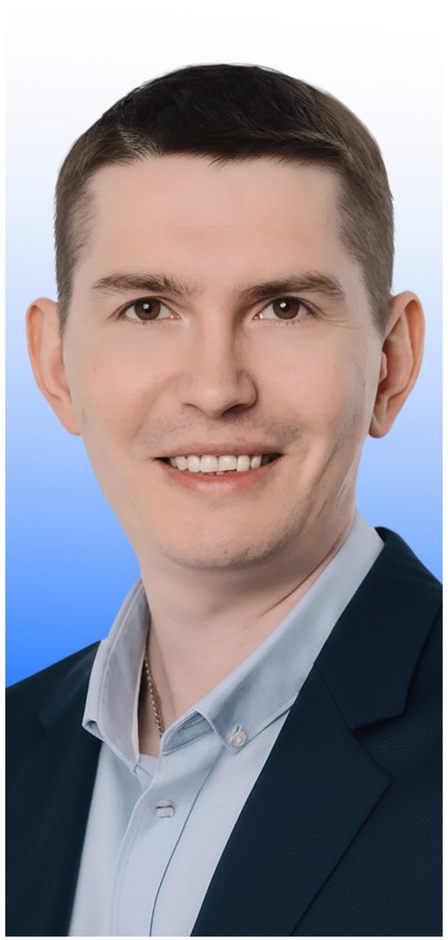
Спикеры

Владимир Девятайкин

Менеджер продукта Deckhouse Stronghold, «Флант»

Ильдар Гарипов

Руководитель отдела информационной безопасности, «Флант»



Александр Фатин

Руководитель отдела аналитики, Ximi Lab

О компании «Флант»

Deckhouse
ФЛАНТ

17+
лет опыта
в Open Source

С 2017
года используем
Kubernetes в production

№1
контрибьютор в проекты
CNCF из России

500+
сотрудников

>260
компаний-пользователей

В топе
вендоров ИТ-решений для банков*
и промышленности**

Реестр
российского ПО

Лицензии и сертификат
ФСТЭК России

АРПП «Отечественный
софт»

* Рейтинг «Крупнейшие ИТ-вендоры в банках», TAdviser, 2024
** Рейтинг «Крупнейшие ИТ-вендоры в промышленности», TAdviser, 2024

ФЛАНТ

Синергия опыта вендора, интегратора, сервисной и консалтинговой компании

Deckhouse

Deckhouse — продуктовое подразделение, разработчик продуктов для построения надёжной enterprise-инфраструктуры

DaaS

DaaS — комплексное DevOps-сопровождение инфраструктуры в режиме 24/7 силами выделенной DevOps-команды

Express 42

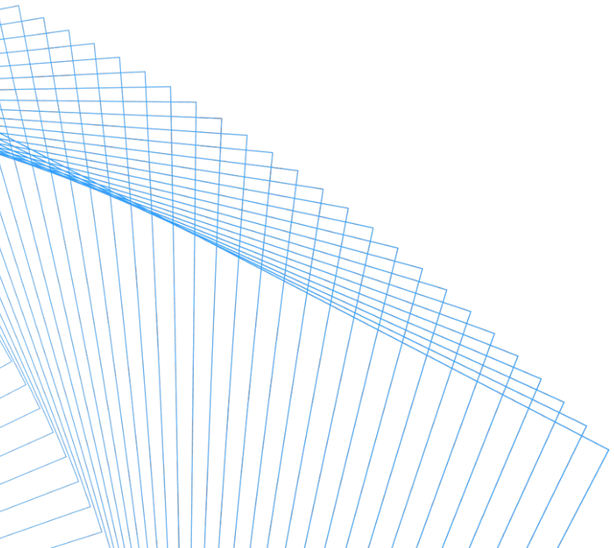
«Экспресс 42» — DevOps-консалтинг. От анализа узких мест в ИТ-процессах до создания роадмапа изменения ИТ для реализации цифровой трансформации

Что можно считать секретами?

- Пароли
- Сертификаты
- Токены
- Ключи API
- SSH-ключи

Источники утечек секретов

- Значимая доля образов на Docker Hub и репозиториев содержит действующие секреты
- Многие разработчики не отзывают скомпрометированные ключи после обнаружения утечки



Исходный код и репозитории

Docker-образы и артефакты сборки

Конфигурации и окружения

Бинарники и прошивки

CI/CD — логи, переменные, кеш

Секреты в CWE и CVE и последствия утечек

Классификация уязвимостей (CWE):

CWE-798 **CWE-798** — Use of Hard-coded Credentials

CWE-259 **CWE-259** — Use of Hard-coded Password

CWE-321 **CWE-321** — Use of Hard-coded Cryptographic Key

CVE: множество зарегистрированных уязвимостей связаны с CWE-798, CWE-259. Поиск в базах CVE по CWE-798 / CWE-259 даёт актуальный перечень инцидентов.

Последствия утечки секретов:

01 Обход аутентификации и несанкционированный доступ

02 Извлечение секретов из бинарников и дизассемблированного кода

03 Использование одного и того же секрета во всех инсталляциях продукта (массовая компрометация)

04 Латеральное перемещение и закрепление в сети и т. д.

Фрагментированное хранение секретов: риски и последствия

Во многих компаниях секреты — пароли, API-ключи, сертификаты, SSH-ключи, токены — хранятся **фрагментировано**: в Git-репозиториях, CI/CD-пайплайнах, Docker-образах и конфигурационных файлах, без единой точки контроля

Последствия отсутствия единого управления:

01 Нет единой точки контроля

02 Секреты в истории коммитов

03 Рост риска утечки

04 Аудит и соответствие

05 Ответственность

Выстраивание управляемого процесса работы с секретами и единой точки контроля — необходимое условие снижения рисков

Секреты в контексте SSDLC *

Безопасность секретов — часть безопасной разработки на всех этапах:

- **Требования** — запрет хардкода, использование vault/secret manager
- **Проектирование** — архитектура доступа к секретам
- **Разработка** — не коммитить секреты, использовать сканирование
- **Тестирование** — тестовые секреты отдельно, не в коде
- **Развёртывание и эксплуатация** — внедрение секретов через защищённые механизмы, ротация

Подход принят регуляторами, отраслевыми стандартами и крупными поставщиками ПО по всему миру

Ключевые международные фреймворки и стандарты:

01 **NIST SSDF** — Secure Software Development Framework, SP 800-218

02 **ISO/IEC 27034** — Application security

03 **BSIMM** — Building Security In Maturity Model

04 **OWASP SAMM** — Software Assurance Maturity Model

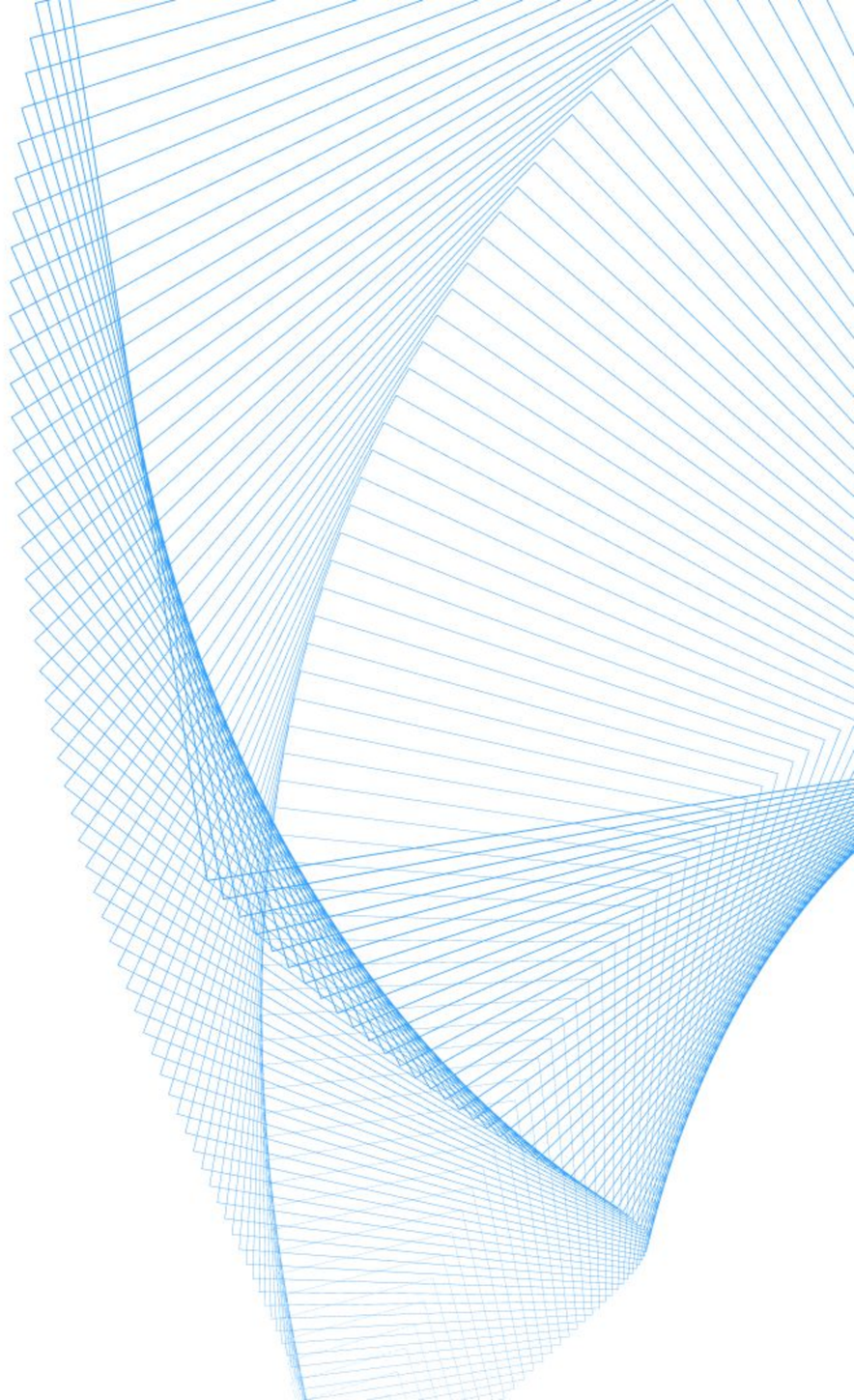
05 **ISO/IEC 15408** — Common Criteria

* Secure software development lifecycle

SSDLC* в России

Российская нормативная база задаёт обязательные требования к безопасной разработке ПО и его сертификации и в целом согласована по духу с общемировым подходом SSDLC

ГОСТ Р 56939-2024 «Защита информации. Разработка безопасного программного обеспечения. Общие требования» — базовый документ для построения и аудита процессов РБПО** в России



* Secure software development lifecycle

** Разработка безопасного программного обеспечения

Зачем внедрять SSDLC*

01 **Экономика**

Стоимость исправления уязвимости растёт на порядки к концу ЖЦ. Плюс регрессии, патчи, координация с заказчиками

02 **Регуляторика**

187-ФЗ в части КИИ; приказы ФСТЭК России № 239 (КИИ), № 117 (ГИС), Положение Банка России № 719-П — явные или фактически обязательные требования к процессам РБПО и/или сертификации. PCI DSS, отраслевые стандарты — те же тренды

03 **Цепочки поставок**

SBOM, зависимость от стороннего кода, требования к поставщикам делают процессы безопасной разработки обязательными

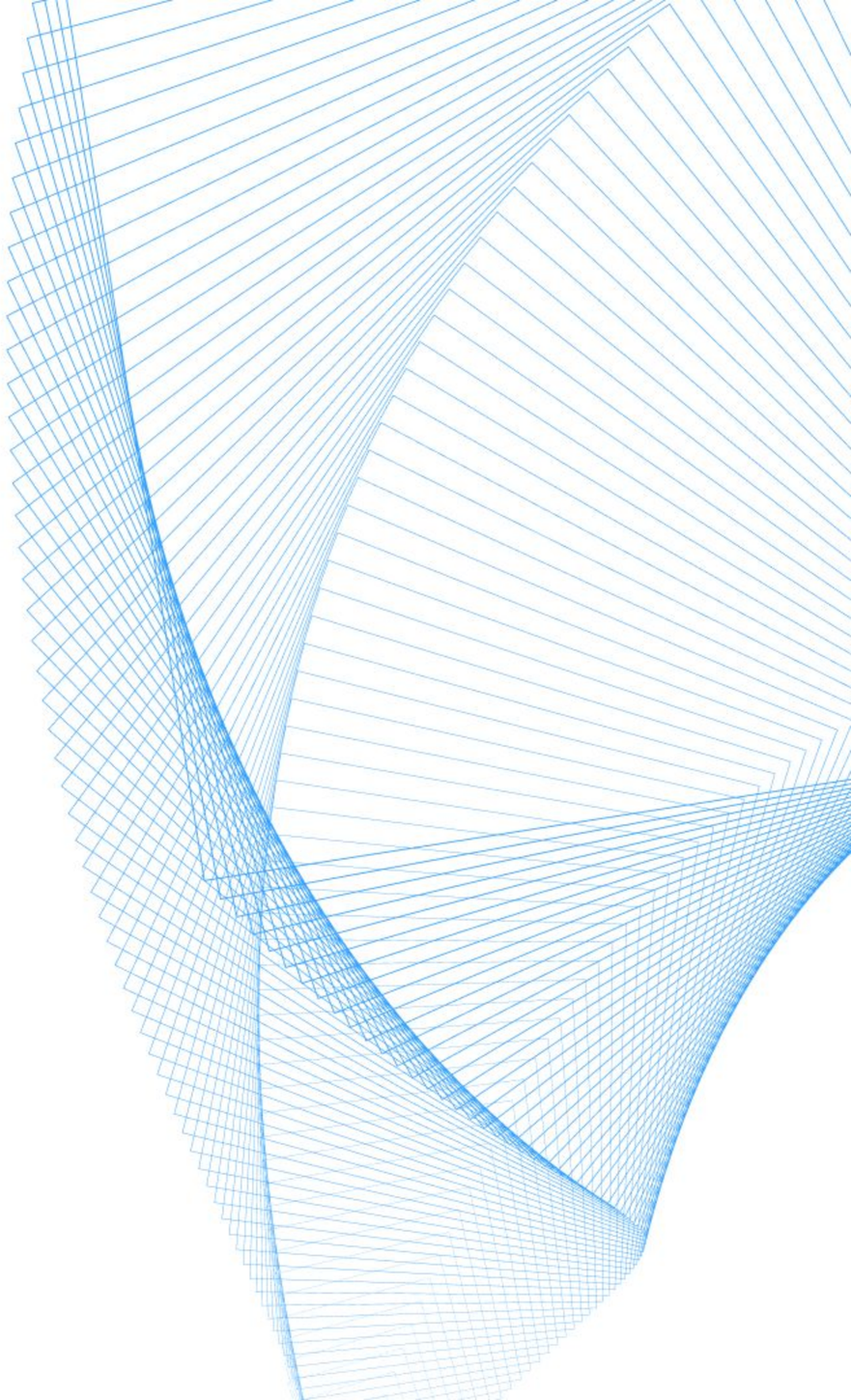
04 **Метрики и зрелость**

Управляемый процесс, измеримые показатели (например, время до закрытия уязвимостей, покрытие SAST/SCA)

* Secure software development lifecycle

Выполнение требований ГОСТ Р 56939-2024

Раздел 5 ГОСТ Р 56939-2024 задаёт процессы разработки безопасного программного обеспечения. Каждый процесс описывает цели, входы/выходы и требования, выполнение которых необходимо для соответствия стандарту и для аудита/сертификации



Процесс 5.15 — цель и требования

Назначение процесса: исключить утечку и неправомерное использование секретов, используемых при разработке, сборке, тестировании и поставке ПО

Что нужно учитывать при выстраивании процесса:

01 Идентификация секретов

02 Недопущение попадания секретов в код и артефакты

03 Безопасное хранение и передача

04 Контроль и обнаружение

Практики
по направлениям

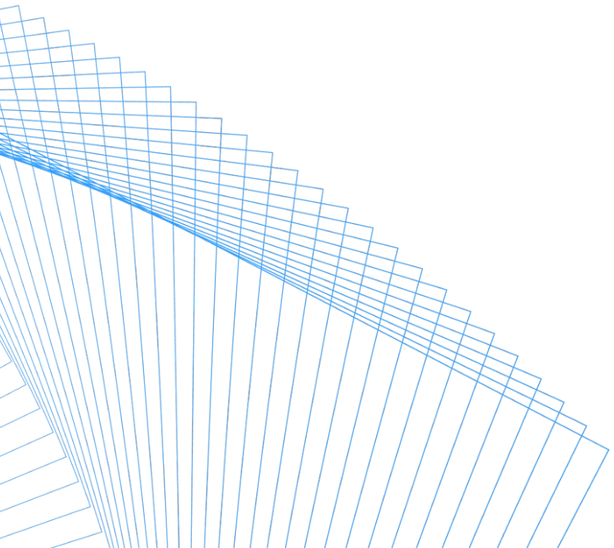
Исходный код
и репозитории

Сборка
и CI/CD

Хранение
и доступ

Обнаружение
и реагирование

Документация



Практики по направлениям

Направления	Практики обеспечения безопасности секретов
Исходный код и репозитории	Запрет коммита секретов; использование переменных окружения, конфигов вне репозитория или ссылок на хранилище секретов; pre-commit / CI-проверки (secret scanning); .gitignore для конфигов с секретами
Сборка и CI/CD	Секреты в CI — только через защищённые переменные пайплайна или интеграцию с vault; не логировать и не выводить секреты в артефакты; отдельные учётные записи и токены для сборки с минимальными правами
Хранение и доступ	Централизованное хранилище секретов (Deckhouse Stronghold или аналог); шифрование секретов; учёт и контроль доступа (кто, когда, к какому секрету); ротация по регламенту
Обнаружение и реагирование	Автоматическое сканирование кода и истории (Tron.ASOC, Gitleaks, TruffleHog, встроенные средства GitLab/GitHub и др.); при обнаружении — ротация секрета, удаление из истории (если применимо), разбор причин и обновление правил
Документация	Регламент или политика по обращению с секретами (классификация, где хранить, как передавать, ротация); обучение разработчиков

Регуляторика РБПО * в России

ГОСТ

ГОСТ Р 56939-2024

- Общие требования к процессам РБПО

ФСТЭК России

ФСТЭК России

- **Приказ № 239 от 25.12.2017** – требования по обеспечению безопасности ЗО КИИ
- **Приказ № 117 от 11.04.2025** – требования о защите информации в ГИС
- **Приказ № 76 от 02.06.2020** – уровни доверия к СЗИ и средствам обеспечения безопасности ИТ

Положение Банка России

Положение Банка России № 719-П от 04.06.2020

- Требования к обеспечению защиты информации при осуществлении переводов денежных средств

* Разработка безопасного программного обеспечения

Секреты на практике: обнаружение и защита

Обнаружение

- Сканирование кода и истории репозитория (pre-commit, CI, PR)
- Сканирование образов и артефактов
- При срабатывании — блокировка, ротация секрета, расследование

Защита

- Секреты не в коде и не в репозитории
- Хранение в vault или защищённых переменных
- Получение в рантайме
- Маскирование в логах
- Минимальные права для сборки/деплоя
- Аудит и регламент

TRON ASOC

Передовая платформа для автоматизации процессов безопасной разработки

Интеграции в TRON ASOC

Источники сканирования

- Github
- GitLab
- Bitbucket
- Harbor
- JFrog Artifactory
- Nexus OSS
- URL приложения
- Jenkins
- Teamcity

ИТ окружение

- Jira
- LDAP
- SSO
- Active Directory

Инструменты безопасности

- Kaspersky Container Security
- Positive Technologies Application Inspector
- Solar AppScreener
- CodeScoring
- Trivy
- Grype
- KICS
- AQUA
- OWASP Dependency Check
- Semgrep
- PVS Studio
- ESLint
- Gitlab Advanced SAST
- SASTAV
- GitLeaks
- AppSec.Sting
- И другие
- \+ Любой другой инструмент через SARIF или формат ASOC'а

Дополнительно

- Открытый API, полностью покрытый Swagger;
- Кастомизируемый алертинг через почту по любым событиям в системе.
- Интеграция со средствами разработки

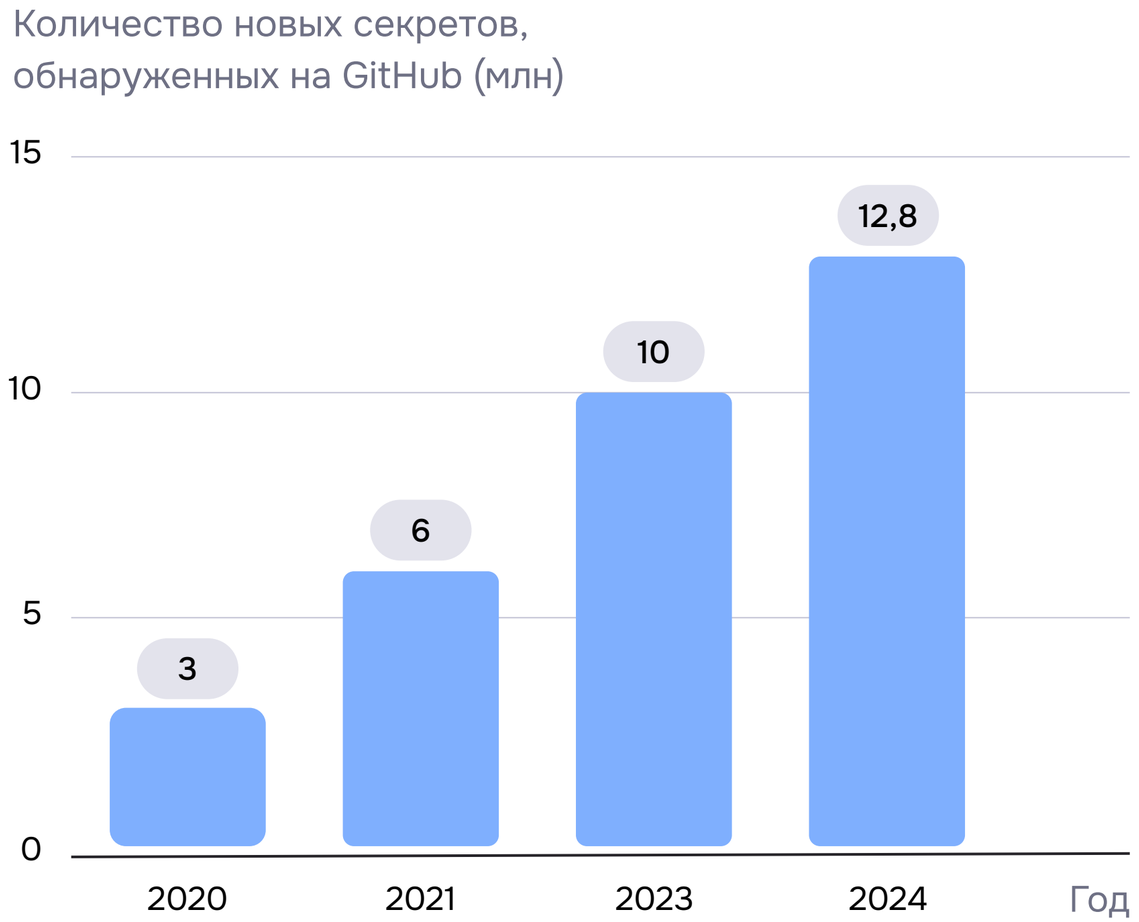
Инструменты для поиска секретов

TRON ASOC поддерживает интеграции
с большинством наиболее популярных в мире
инструментов для поиска секретов

Gitleaks	Импорт отчётов
TruffleHog	—
Gitlab SAST	Импорт отчётов
GitGuardian	—
Solar appScreener	Импорт отчётов Нативная интеграция с запуском из ASOC
SASTAV	Импорт отчётов Нативная интеграция с запуском из ASOC
PT Application Inspector	Импорт отчётов Нативная интеграция с запуском из ASOC
AppSec.Wave	—
CodeScoring	Импорт отчётов Нативная интеграция с запуском из ASOC
Checkmarx One	—
SonarQube	—

Количество секретов непрерывно растёт

С ростом сложности цифровых цепочек поставок **разрастание секретов становится ахиллесовой пятой** для организаций любого размера и уровня безопасности*

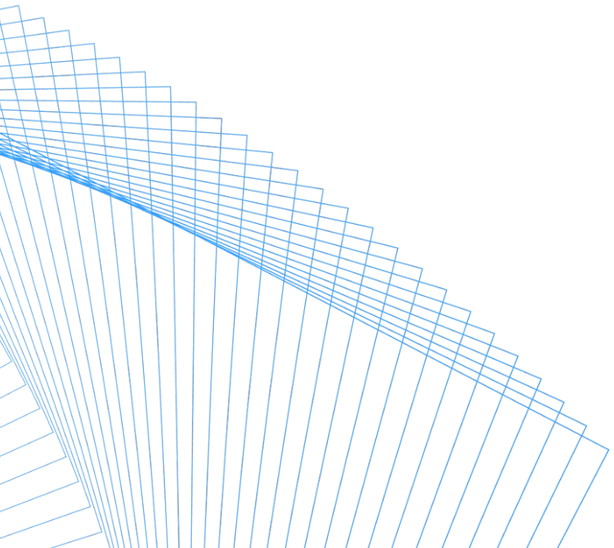


* По данным отчёта The State Of Secrets Sprawl 2024 компании GitGuardian

Где вы храните пароль от БД?

3560

ответов



28 %

В Git в зашифрованном виде

20 %

В HashiCorp Vault

19 %

В Git в открытом виде

12 %

В GitLab/GitHub/Jenkins

8 %

Внешнее KV

8 %

Нет пароля

5 %

Другое

Как правильно хранить секреты

В репозитории с секретами или репозитории приложений

В системе управления конфигурациями

В системе деплоя (Jenkins, Teamcity)

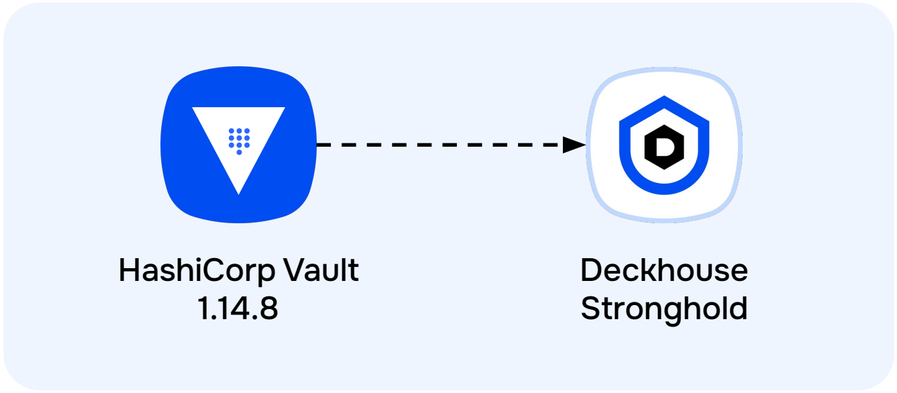
Только на серверах, на которых работает ваш сервис

Только на личном компьютере

В отдельном хранилище секретов

Что такое Deckhouse Stronghold

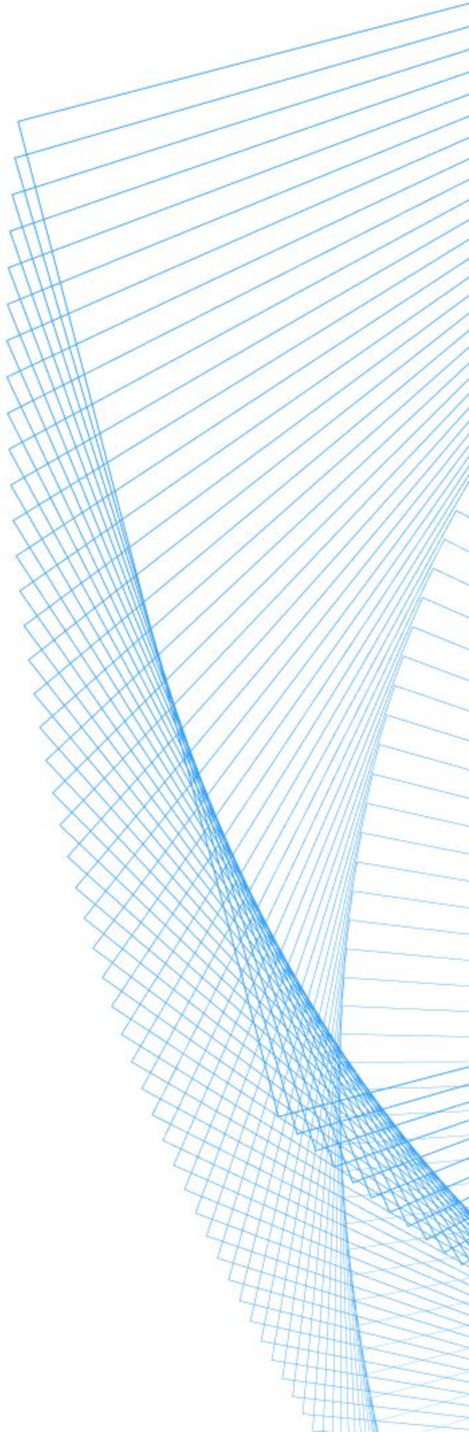
Решение для централизованного управления жизненным циклом секретов. Защищает пароли, ключи API, сертификаты, SSH-ключи, токены и другие конфиденциальные данные от утечек, а также обеспечивает безопасную доставку секретов в приложения.



Мы **не копируем**
HashiCorp Vault,
мы **переосмысливаем**
хранилище секретов
и **создаём стандарт**
для российского рынка

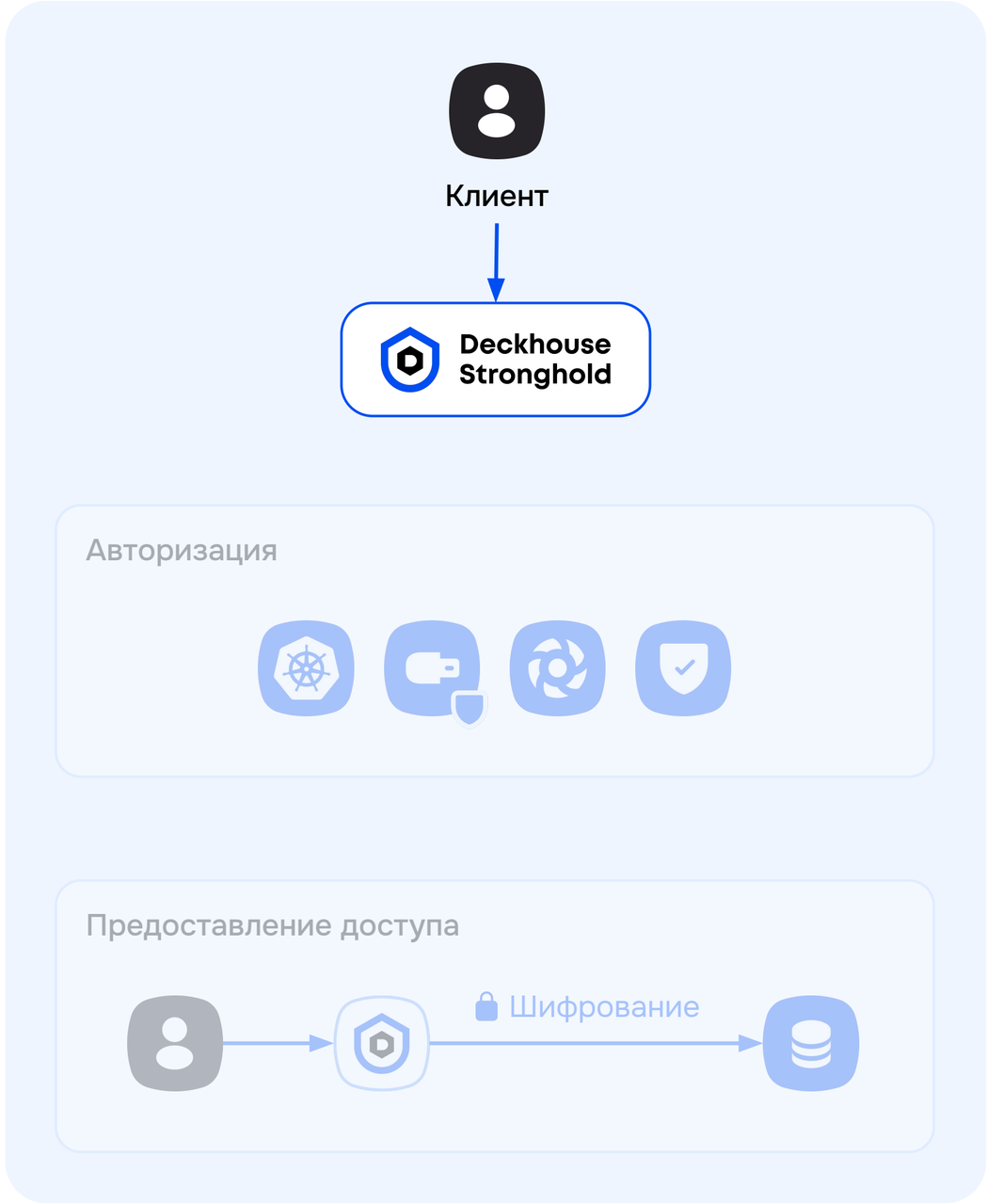
Основные возможности продукта

- Хранение секретов как key-value
- Доступ к хранилищу через API и UI
- Хранение данных в зашифрованном виде
- Разграничение доступа с помощью гибкого набора политик
- Отказоустойчивость «из коробки»
- Совместимость с API HashiCorp Vault
- Полная наблюдаемость жизненного цикла секретов
- Интеграция с внутренними и внешними сервисами



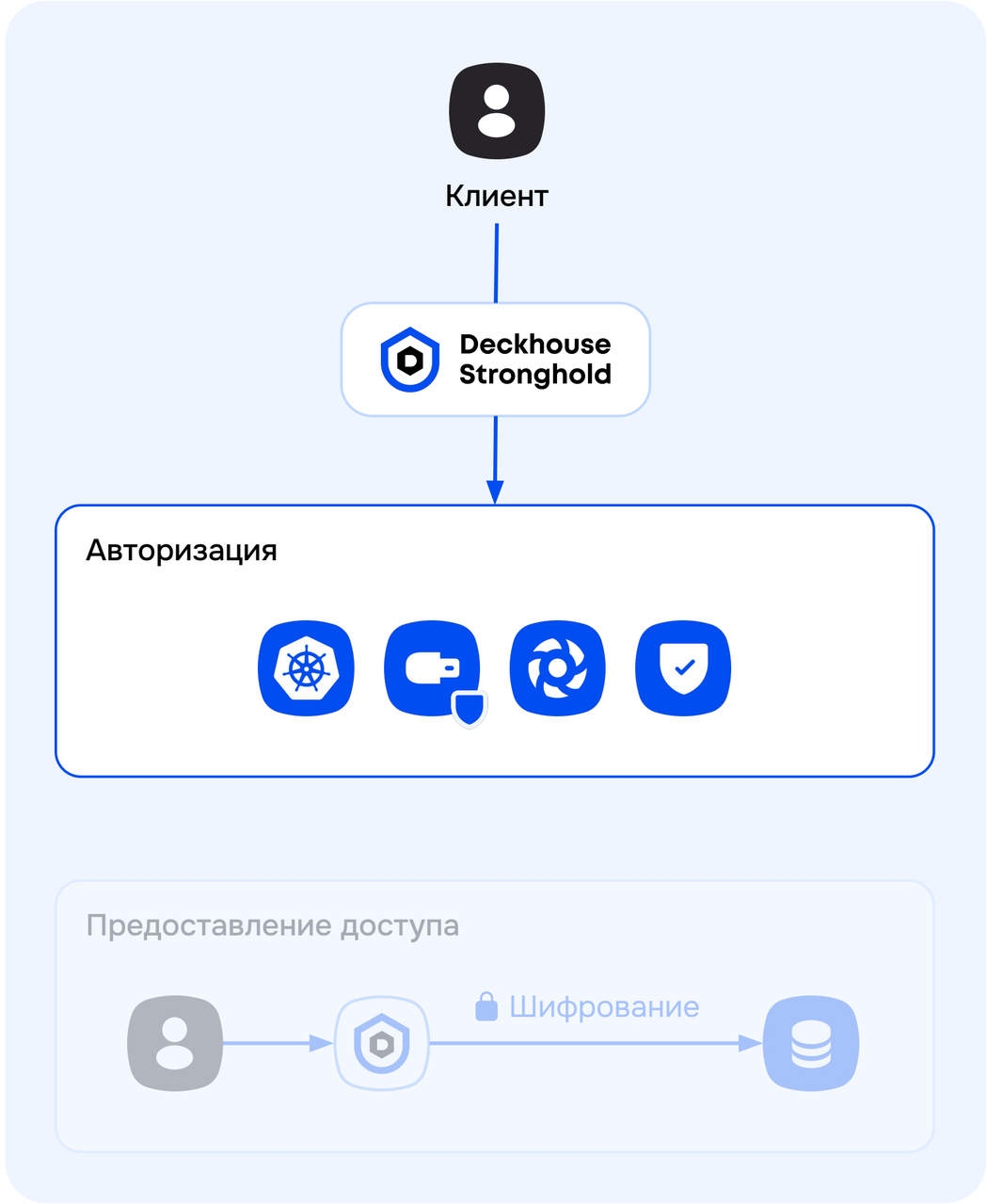
Как работает Stronghold

- Клиент через Stronghold или внешнюю систему **подтверждает**, что обращается **именно он**



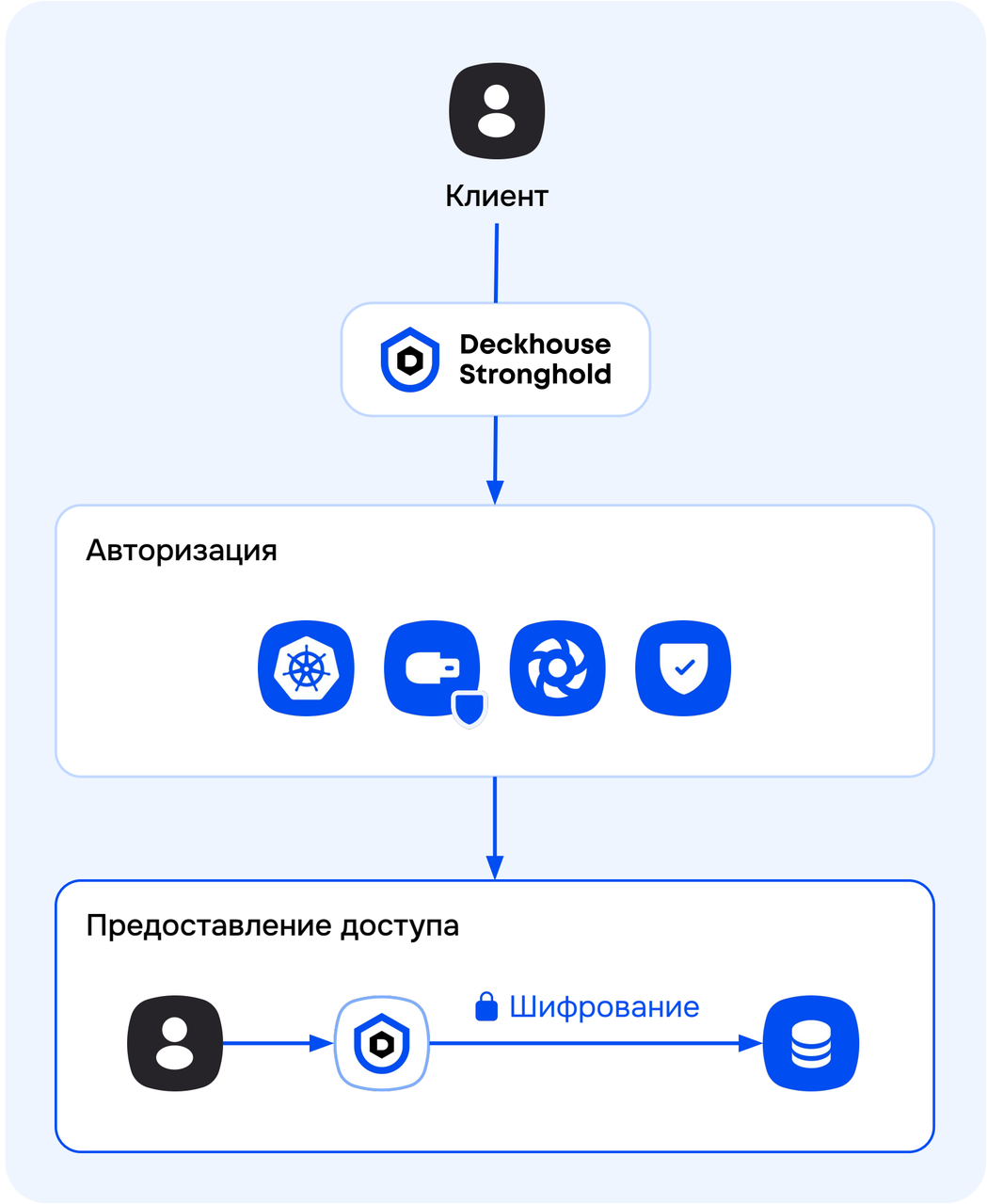
Как работает Stronghold

- Клиент через Stronghold или внешнюю систему **подтверждает**, что обращается **именно он**
- Stronghold **анализирует**, можно ли **предоставить доступ** к запрошенному секрету



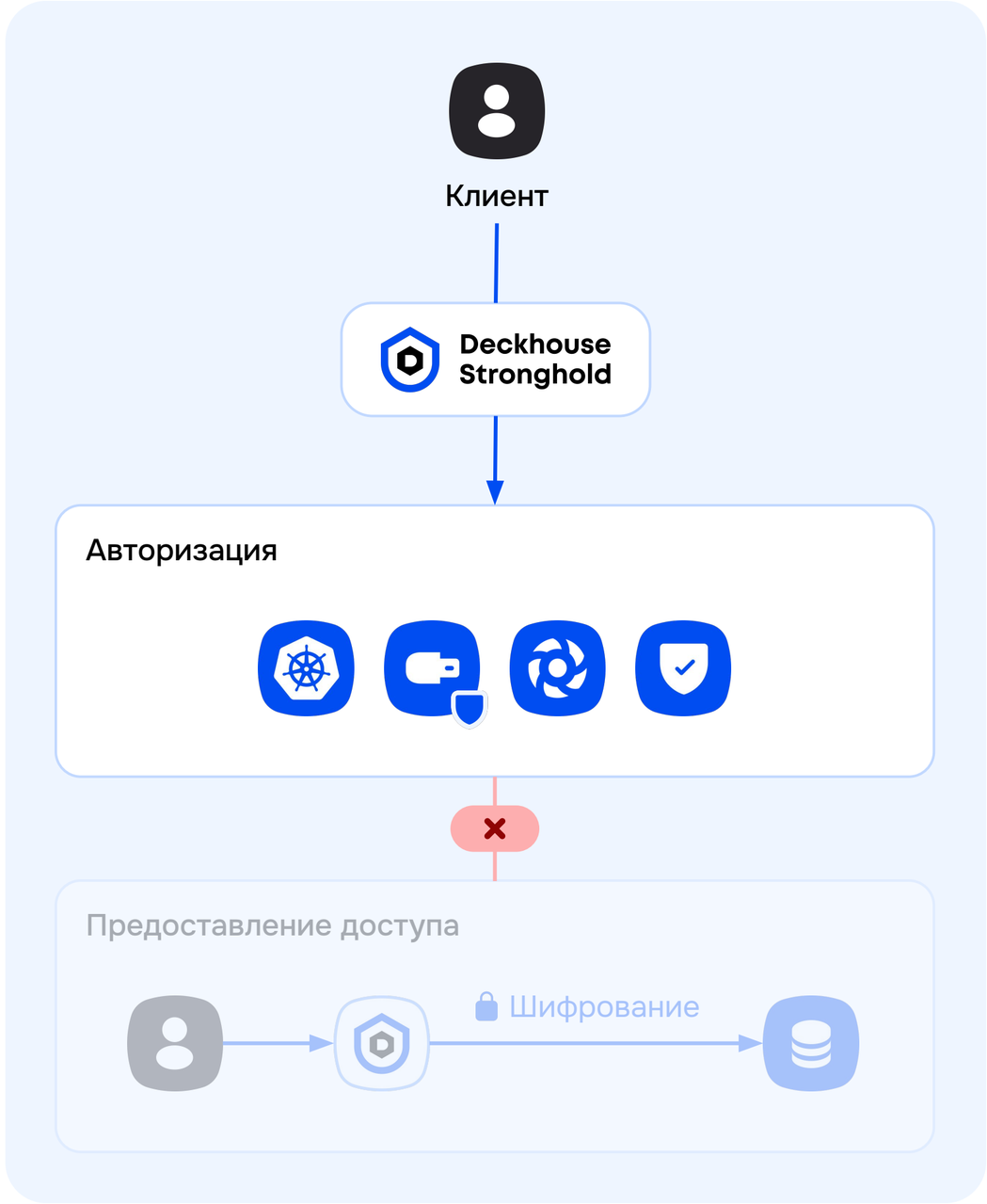
Как работает Stronghold

- Клиент через Stronghold или внешнюю систему **подтверждает**, что обращается **именно он**
- Stronghold **анализирует**, можно ли **предоставить доступ** к запрошенному секрету
- Если **доступ есть**, то операция по чтению или записи секрета **разрешается**



Как работает Stronghold

- Клиент через Stronghold или внешнюю систему **подтверждает**, что обращается **именно он**
- Stronghold **анализирует**, можно ли **предоставить доступ** к запрошенному секрету
- Если **доступ есть**, то операция по чтению или записи секрета **разрешается**.
- Если **доступа нет** — операция **отклоняется**

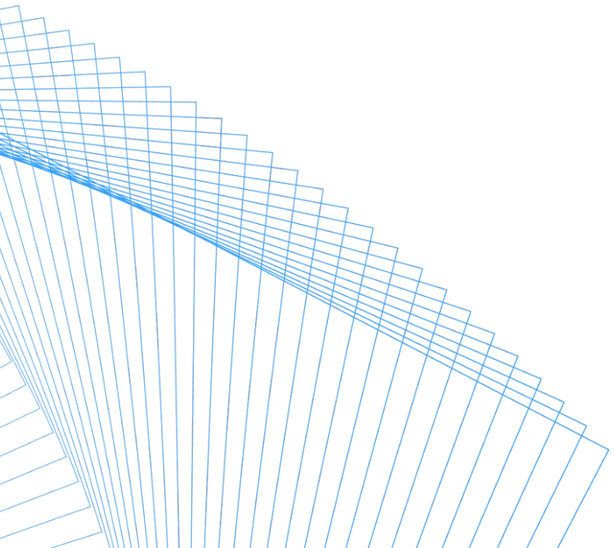


Хранение секретов в Git-репозитории

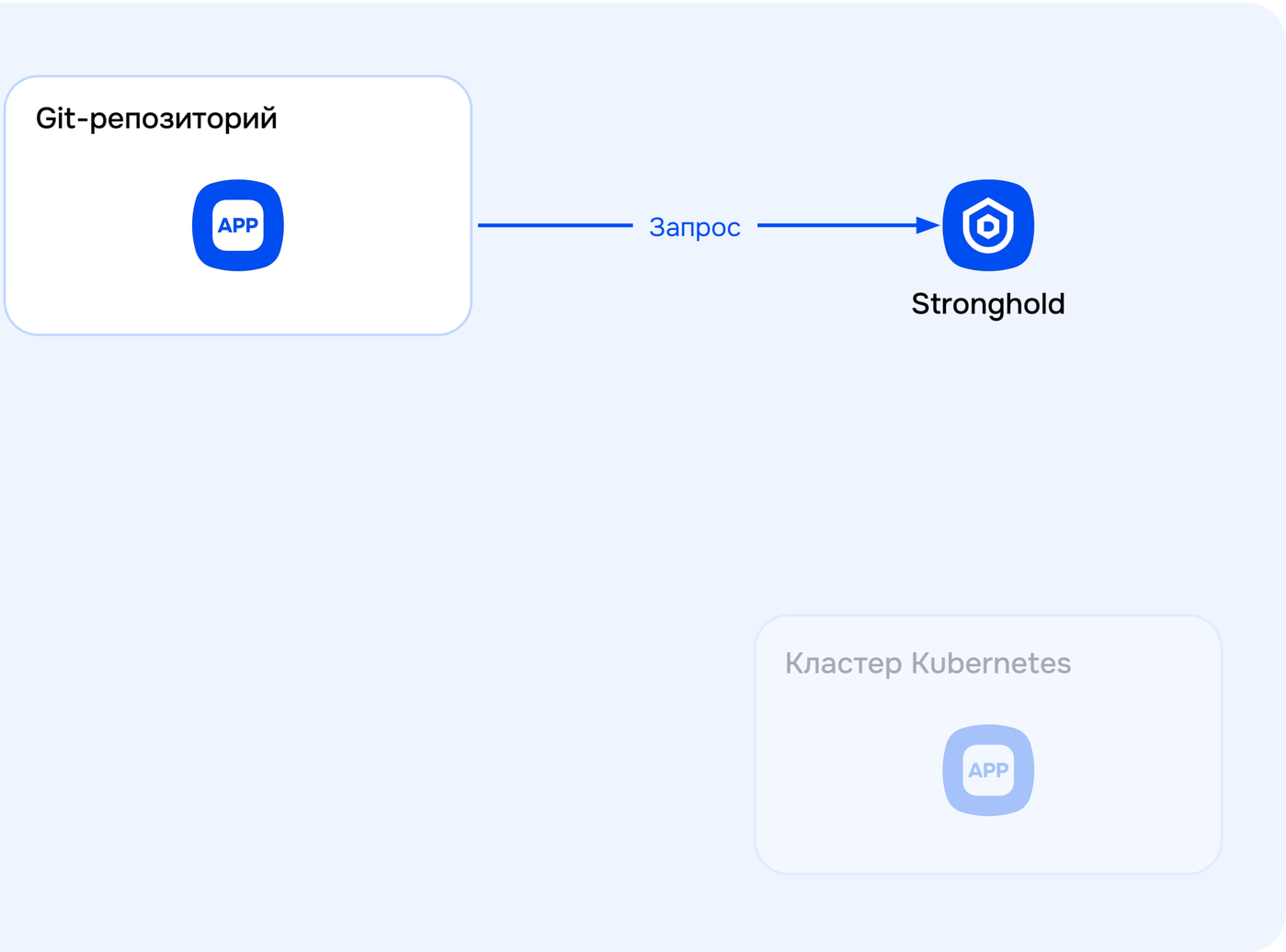
В репозитории в открытом виде хранятся:

- Приложение с паролем
- Токен для развёртывания кластера Kubernetes

```
DB_HOST = "postgres"
DB_PORT = 5432
DB_NAME = "appdb"
DB_USER = "appuser"
DB_PASSWORD = "apppassword"
```

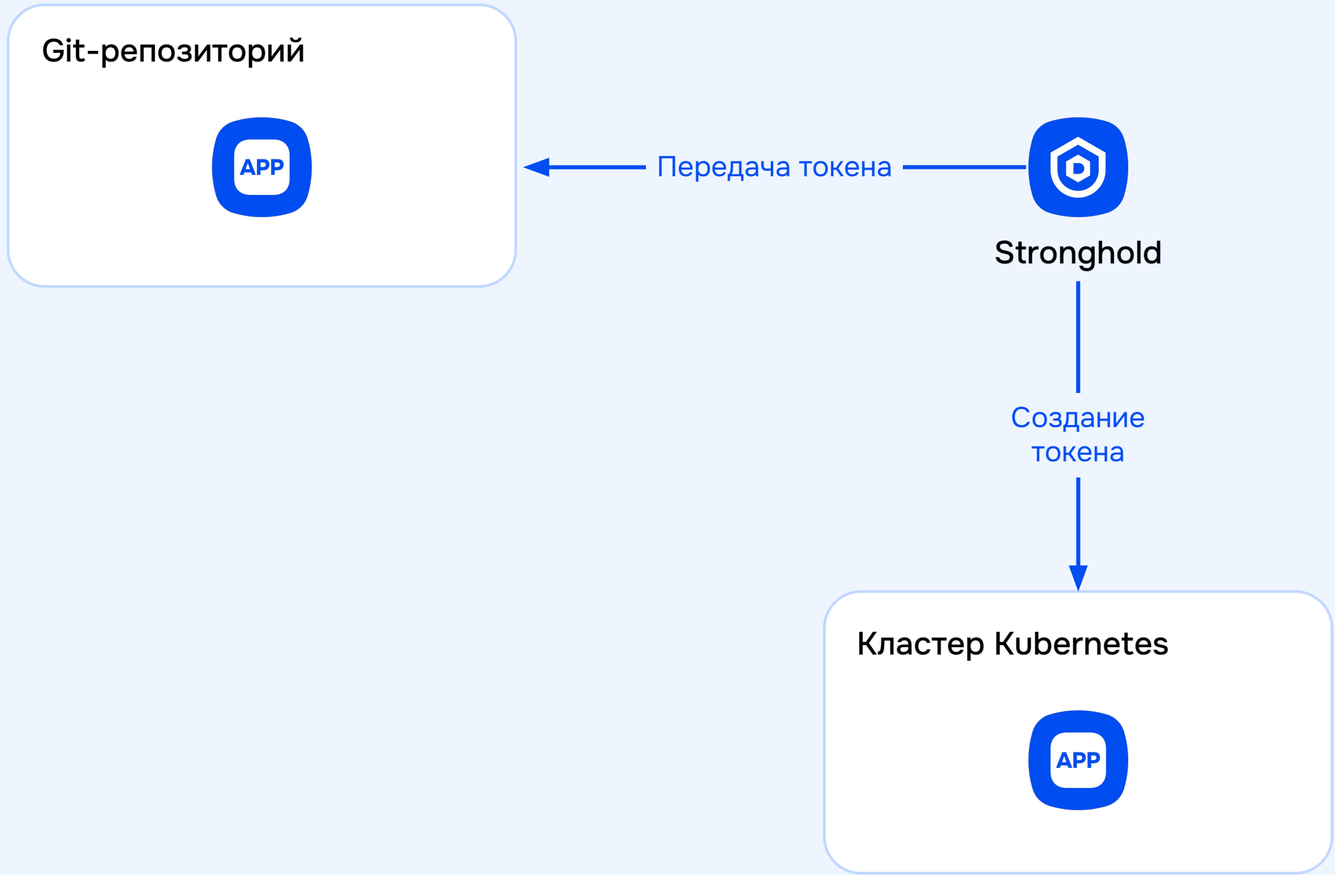


Хранение секретов в Deckhouse Stronghold



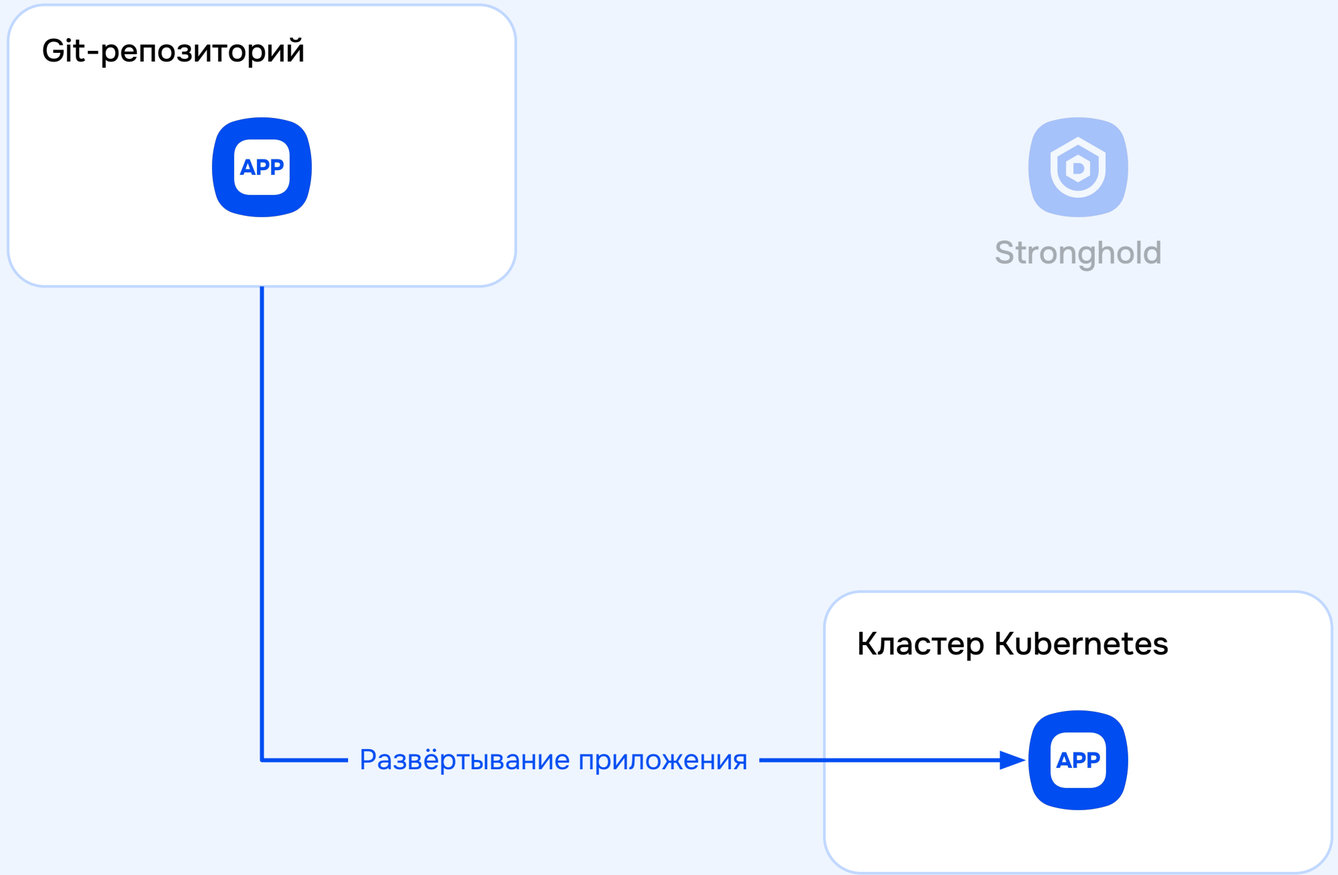
- Проект приложения, размещённый в Git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes

Хранение секретов в Deckhouse Stronghold



- Проект приложения, размещённый в git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes
- Stronghold валидирует запрос, создаёт временный токен и передаёт его проекту в Git-репозиторий

Хранение секретов в Deckhouse Stronghold



- Проект приложения, размещённый в Git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes
- Stronghold валидирует запрос, создаёт временный токен и передаёт его проекту в Git-репозиторий
- Проект с помощью временного токена выполняет развёртывание приложения в кластере Kubernetes (после развёртывания приложения токен удаляется по TTL)

Настроить JWT

Настройка | Параметры

Подробности

URL discovery OIDC

https://token.actions.githubusercontent.com

Роль по умолчанию

JWKS CA PEM

Ввести как текст

Выбрать файл... | Файл не выбран

Выбрать файл на компьютере

JWKS URL

Режим ответа OIDC

Типы ответов OIDC

Добавляйте один элемент на строку.

Добавить

Jwks пары

Пространство имен в состоянии OIDC

Назад к главному меню

Аутентификация

Методы аутентификации

Многофакторная аутентификация

OIDC провайдер

Организация

Пространства имён

Группы

Сущности

Управление

Параметры аренды

root

Назад к главному меню

Аутентификация

Мето

Мног

OIDC

Организация

Пространства имён

Группы

Сущности

Управление

Параметры аренды

root

github

Настроить JWT

Настройка | Параметры

Подробности

URL discovery OIDC

https://token.actions.githubusercontent.com

JWKS CA PEM

Ввести как текст

Выбрать файл... | Файл не выбран

Выбрать файл на компьютере

JWKS URL

Режим ответа OIDC

Типы ответов OIDC

Добавляйте один элемент на строку.

Добавить

Jwks пары

Пространство имен в состоянии OIDC

Create role

Название

demo-deploy

Разрешённые URI перенаправления

Добавляйте один элемент на строку.

Добавить

Связанные аудитории

Добавляйте один элемент на строку.

Добавить

Связанные утверждения

```
{
  "repository": "ximi/demo-app"
}
```

Map of claims/values which must match for login

Тип связанных утверждений

Назад к главному меню

Аутентификация

Методы аутентификации

Многофакторная аутентификация

OIDC провайдер

Организация

Пространства имён

Группы

Сущности

Управление

Параметры аренды

root

roles

Create role

Название

demo-deploy

```
{
  "repository": "ximi/demo-app"
}
```

Назад к главному меню

Аутент

Мето

Многофакторная аутентификация

OIDC провайдер

Организация

Пространства имён

Группы

Сущности

Управ

Пара

Добавляйте один элемент на строку.

Добавить

Связанные аудитории

Добавляйте один элемент на строку.

Добавить

Связанные утверждения

Map of claims/values which must match for login

Тип связанных утверждений

root

Создать политику Acl

Название

kubernetes-deploy-demo

Политика Загрузить файл

```
path "kubernetes/creds/deploy_role" {
  capabilities = ["update"]
}
```

Используйте Alt+Tab (Option+Tab на MacOs) для перехода к следующему полю

Дополнительную ифнормацию про acl-политики вы можете найти здесь.

Создать политику | Отменить

Acl Policies

Создать политику Acl

Назад к главному меню

Полити

ACL

Парс

Название

kubernetes-deploy-demo

Политика

Загрузить файл

```
1 path "kubernetes/creds/deploy_role" {
2   capabilities = ["update"]
3 }
4
```

Используйте Alt+Tab (Option+Tab на MacOs) для перехода к следующему полю

Дополнительную ифнормацию про acl-политики вы можете найти здесь.

Создать политику | Отменить

root

Stronghold
Механизмы секретов
Доступ
Политики
Инструменты
API Explorer

Мониторинг
Хранилище Raft
Расписание бэкапов
Учет клиентов
Запечатать Stronghold

root

Редактировать роль

Роль в Stronghold определяет, что будет сгенерировано для Kubernetes и какие правила будут использоваться для этого. Это не роль Kubernetes.

Генерировать токен используя существующую учетную запись сервиса
Введите учетную запись сервиса, которая уже существует в Kubernetes, и Stronghold будет динамически генерировать токен.

Генерировать токен, учетную запись сервиса и объекты ролей
Введите существующую роль (или ClusterRole) для использования. Stronghold будет генерировать токен, учетную запись сервиса и объекты ролей.

Генерировать целую цепочку объектов Kubernetes
Stronghold будет генерировать целую цепочку — роль, токен, учетную запись сервиса и объекты ролей — на основе правил, которые вы предоставляете.

Настройки роли

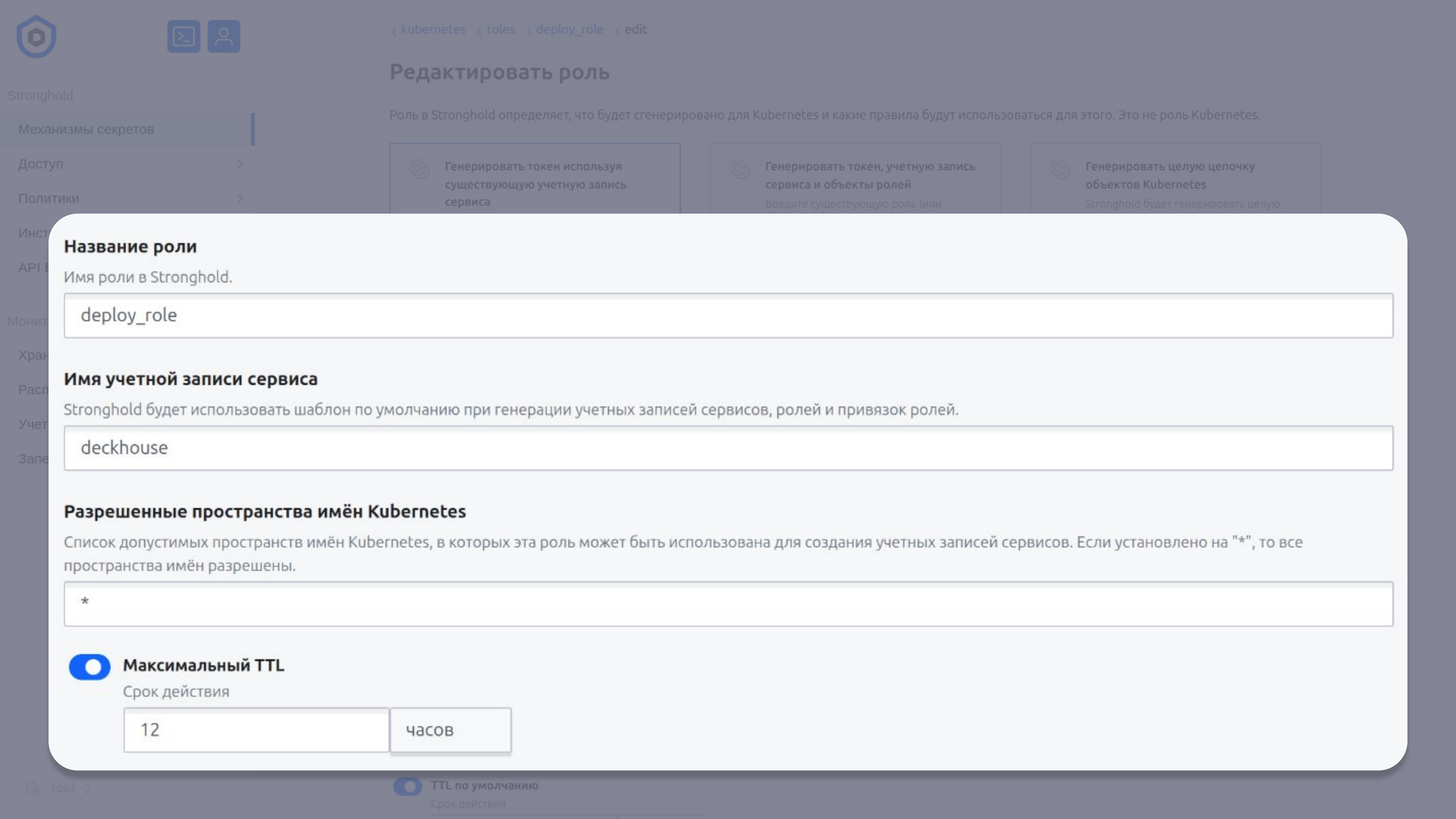
Название роли
Имя роли в Stronghold.
deploy_role

Имя учетной записи сервиса
Stronghold будет использовать шаблон по умолчанию при генерации учетных записей сервисов, ролей и привязок ролей.
deckhouse

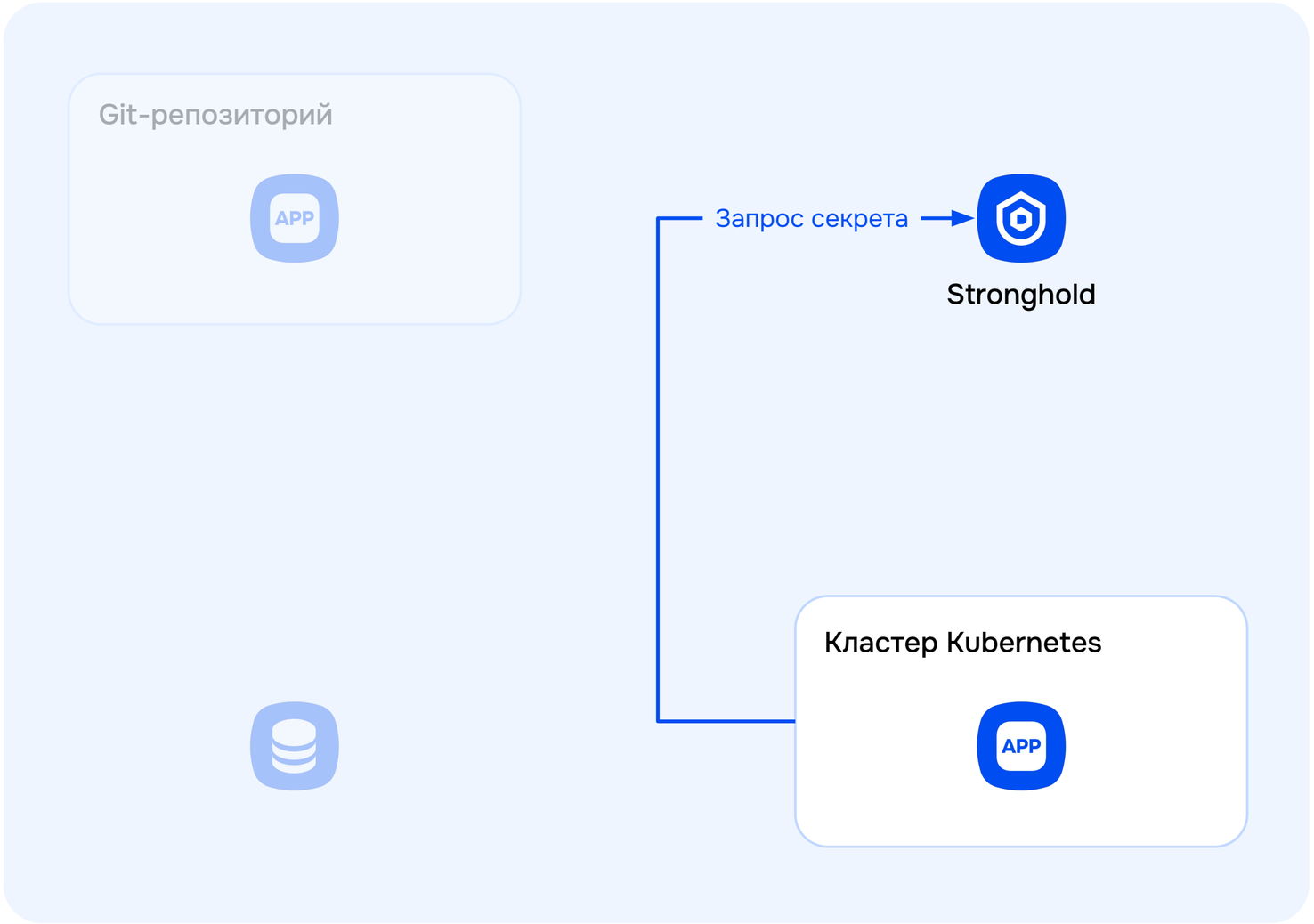
Разрешенные пространства имён Kubernetes
Список допустимых пространств имён Kubernetes, в которых эта роль может быть использована для создания учетных записей сервисов. Если установлено на "*", то все пространства имён разрешены.
*

Максимальный TTL
Срок действия
12 часов

TTL по умолчанию
Срок действия

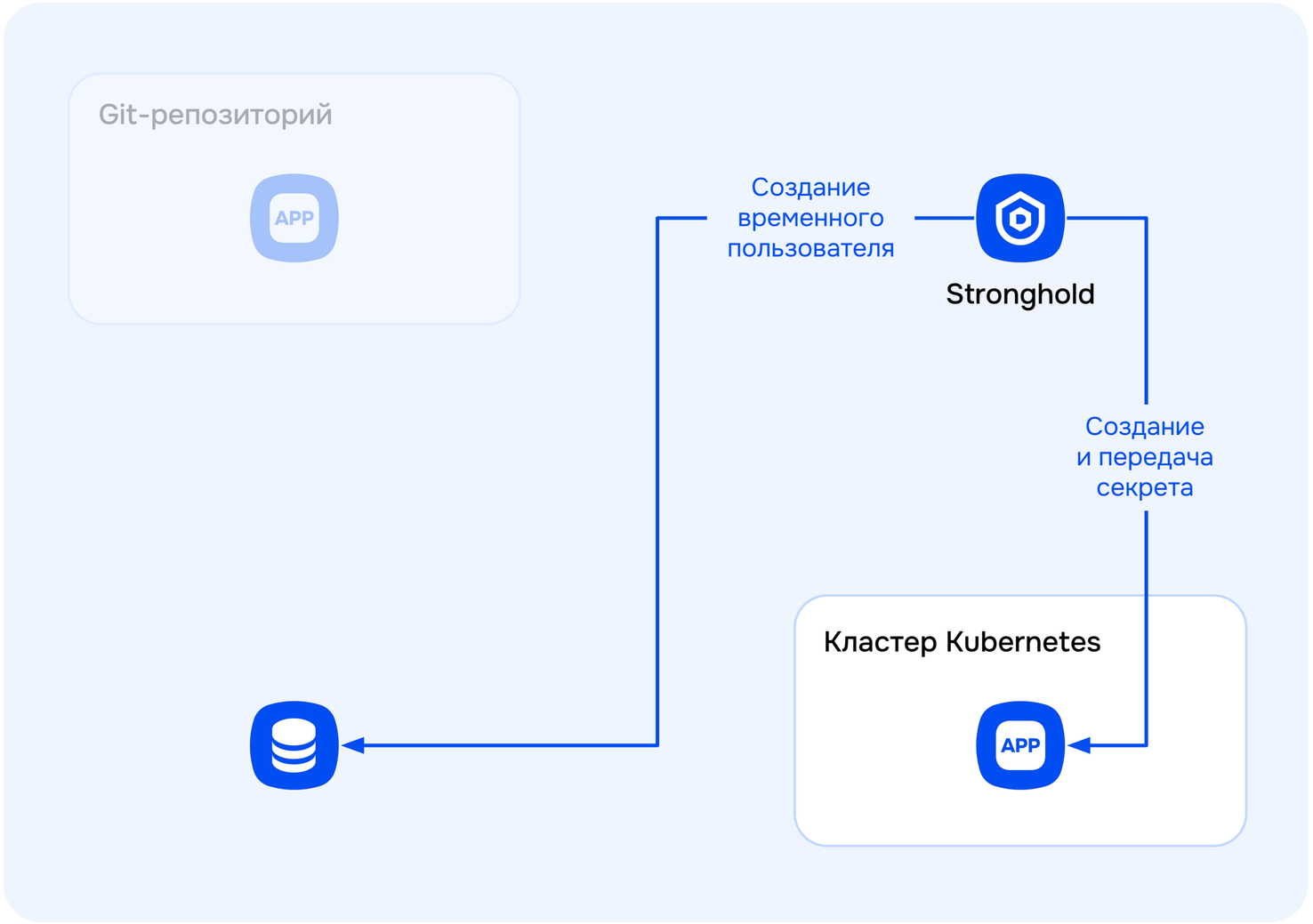

kubernetes
roles
deploy_role
edit
Редактировать роль
Роль в Stronghold определяет, что будет сгенерировано для Kubernetes и какие правила будут использоваться для этого. Это не роль Kubernetes.
Генерировать токен используя существующую учетную запись сервиса
Генерировать токен, учетную запись сервиса и объекты ролей
Введите существующую роль (или
Генерировать целую цепочку объектов Kubernetes
Stronghold будет генерировать целую
Stronghold
Механизмы секретов
Доступ
Политики
Название роли
Имя роли в Stronghold.
deploy_role
Имя учетной записи сервиса
Stronghold будет использовать шаблон по умолчанию при генерации учетных записей сервисов, ролей и привязок ролей.
deckhouse
Разрешенные пространства имён Kubernetes
Список допустимых пространств имён Kubernetes, в которых эта роль может быть использована для создания учетных записей сервисов. Если установлено на "*", то все пространства имён разрешены.
*
Максимальный TTL
Срок действия
12
часов
root
TTL по умолчанию
Срок действия

Хранение секретов в Deckhouse Stronghold



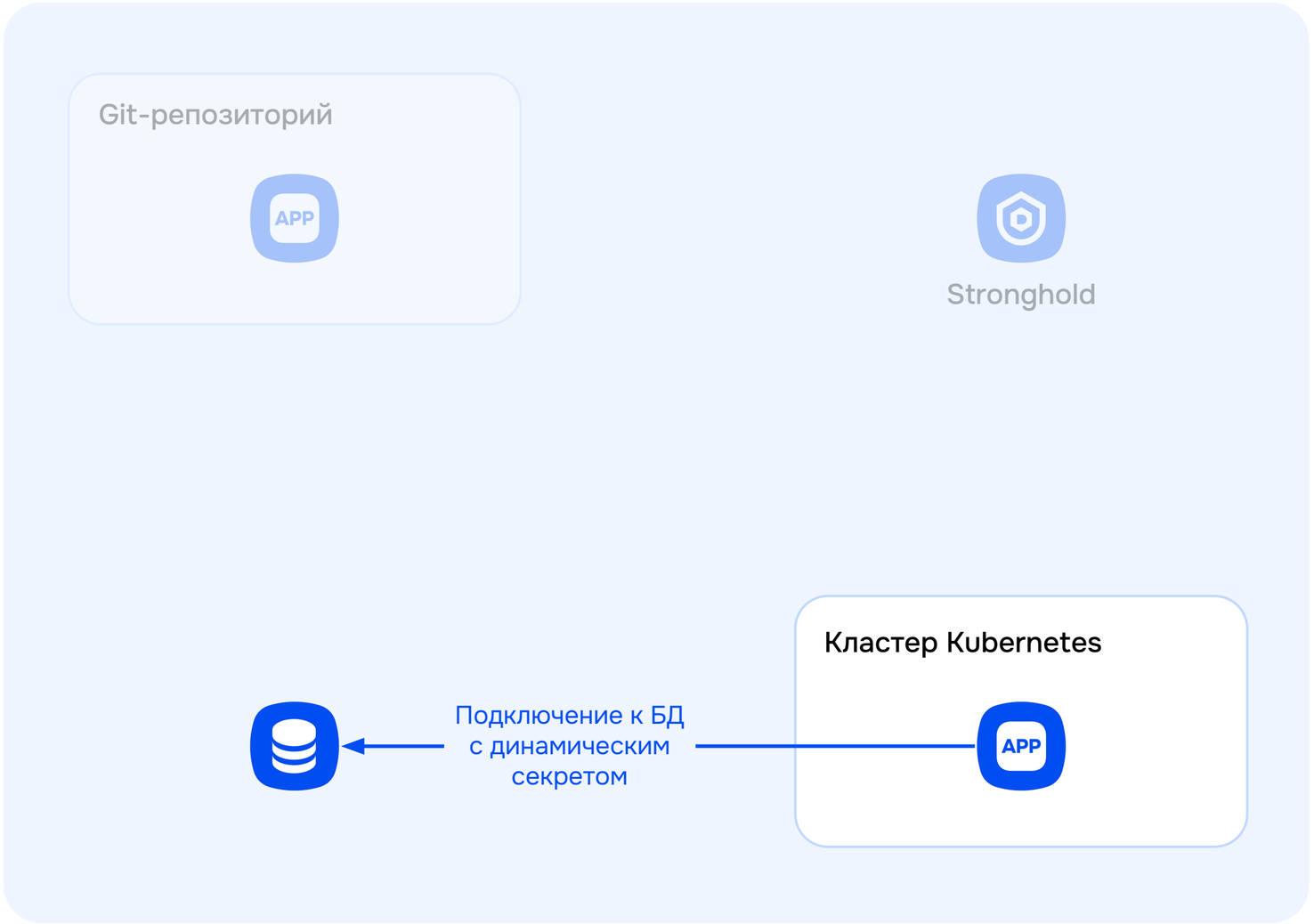
- Проект приложения, размещённый в Git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes
- Stronghold валидирует запрос, создаёт временный токен и передаёт его проекту в Git-репозиторий
- Проект с помощью временного токена выполняет развёртывание приложения в кластере Kubernetes (после развёртывания приложения токен удаляется по TTL)
- Модуль Kubernetes аутентифицируется в Stronghold и запрашивает секрет для подключения к БД

Хранение секретов в Deckhouse Stronghold



- Проект приложения, размещённый в Git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes
- Stronghold валидирует запрос, создаёт временный токен и передаёт его проекту в Git-репозиторий
- Проект с помощью временного токена выполняет развёртывание приложения в кластере Kubernetes (после развёртывания приложения токен удаляется по TTL)
- Модуль Kubernetes аутентифицируется в Stronghold и запрашивает секрет для подключения к БД
- Stronghold валидирует запрос, создаёт временного пользователя в БД, секрет для доступа в БД и возвращает его приложению

Хранение секретов в Deckhouse Stronghold



- Проект приложения, размещённый в Git-репозитории, аутентифицируется в Stronghold и запрашивает токен для развёртывания в Kubernetes
- Stronghold валидирует запрос, создаёт временный токен и передаёт его проекту в Git-репозиторий
- Проект с помощью временного токена выполняет развёртывание приложения в кластере Kubernetes (после развёртывания приложения токен удаляется по TTL)
- Модуль Kubernetes аутентифицируется в Stronghold и запрашивает секрет для подключения к БД
- Stronghold валидирует запрос, создаёт временного пользователя в БД, секрет для доступа в БД и возвращает его приложению
- С помощью временного секрета приложение подключается к БД

Edit role demo

Название

demo

Источник имени псевдонима

serviceaccount_uid

Аудитория

Имена привязанных сервисных аккаунтов

Добавляйте один элемент на строку.

demo

Добавить

Выбор пространства имен привязанной учетной записи службы

Пространства имен привязанных сервисных аккаунтов

Добавляйте один элемент на строку.

demo-ns

Добавить

Токены

Сохранить Отменить

Назад к главному меню

Аутентификация

Методы аутентификации

Многофакторная аутентификация

OIDC провайдер

Организация

Пространства имён

Группы

Сущности

Управление

Параметры аренды

root

Назад к главному меню

Аутентификация

Методы аутентификации

Многофакторная аутентификация

OIDC провайдер

Управление

Параметры аренды

root

roles

Edit role demo

Название

demo

Источник имени псевдонима

serviceaccount_uid

Аудитория

Имена привязанных сервисных аккаунтов

Добавляйте один элемент на строку.

demo

Добавить

Выбор пространства имен привязанной учетной записи службы

Пространства имен привязанных сервисных аккаунтов

Добавляйте один элемент на строку.

demo-ns

Добавить

Токены

Сохранить Отменить

Создать роль

Название роли

demo

Имя подключения

Подключение к базе данных, с помощью которого будут созданы учетные данные

myapp-postgres

Тип роли

dynamic

Настройки роли

Время жизни созданных учетных данных (TTL)

Срок действия

30 минут

Максимальное время жизни созданной учетной записи (Max TTL)

Срок действия

1 дней

Операторы

Запросы для создания

Добавляйте один элемент на строку.

```
CREATE ROLE "{{name}}" WITH LOGIN PASSWORD '{{password}}' VALID UNTIL '{{expiration}}';
```

Добавить

database

Создать роль

Название роли

demo

Имя подключения

Подключение к базе данных, с помощью которого будут созданы учетные данные

myapp-postgres

Расписание бэкапов

Учет клиентов

Запечатать Stronghold

Настройки роли

Время жизни созданных учетных данных (TTL)

Срок действия

30 минут

Максимальное время жизни созданной учетной записи (Max TTL)

Срок действия

1 дней

Операторы

Запросы для создания

Добавляйте один элемент на строку.

CREATE ROLE "{{name}}" WITH LOGIN PASSWORD '{{password}}' VALID UNTIL '{{expiration}}';

Добавить

root

Создать политику Acl

Название

detabase-cred-access

Политика Загрузить файл

```
1 path "database/creds/demo" {
2   capabilities = ["read"]
3 }
4
```

Используйте Alt+Tab (Option+Tab на MacOs) для перехода к следующему полю

Дополнительную ифнормацию про acl-политики вы можете найти здесь.

Создать политику | Отменить

Назад к главному меню

Политики

ACL политики

Парольные политики

root

Назад к главному меню

Политики

ACL политики

Пар

Acl Policies

Создать политику Acl

Название

detabase-cred-access

Политика Загрузить файл

```
1 path "database/creds/demo" {
2   capabilities = ["read"]
3 }
4
```

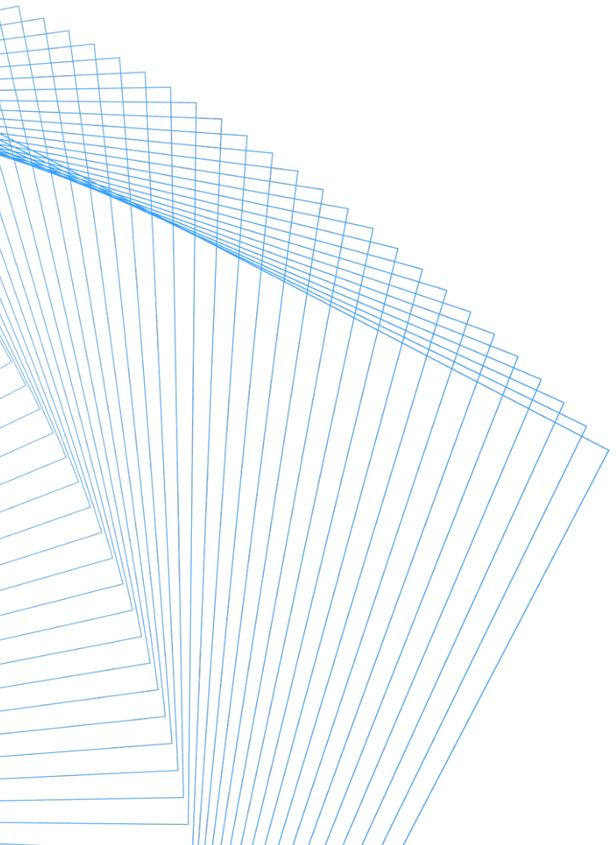
Используйте Alt+Tab (Option+Tab на MacOs) для перехода к следующему полю

Дополнительную ифнормацию про acl-политики вы можете найти здесь.

Создать политику | Отменить

root

Выводы



01 Все секреты хранятся централизованно, в зашифрованном виде и **выдаются только авторизованным сервисам или пользователям**. Это существенно снижает риск утечек и упрощает контроль доступа

02 Динамические секреты не существуют заранее и создаются **только в момент обращения** приложения. Такой пароль или токен выдаётся на ограниченное время и автоматически отзывается после завершения работы

03 Благодаря этому секреты не хранятся постоянно в инфраструктуре, не попадают в репозитории, конфигурационные файлы или образы контейнеров и, соответственно, **не обнаруживаются сканерами уязвимостей**

В чём мы и убедились, проанализировав результаты сканирования TRON ASOC

Deckhouse
ФЛАНТ

TRON ASOC

Спасибо за внимание!

Мы готовы ответить на ваши вопросы!

Чек-лист по безопасному управлению секретами

Оставить заявку на консультацию

contact@deckhouse.ru

+7 (495) 721-10-27

deckhouse.ru

o.novikov@tronasoc.ru

+7 (916) 979-23-47

tronasoc.ru