Варианты установки

1. Общие сведения

В данном разделе приведены технические детали установки платформы StarVault. Поддерживаются различные способы развертывания, выбор которых зависит от требований к целевой среде эксплуатации, уровню отказоустойчивости, инфраструктуре и целям использования. Каждый из вариантов установки адаптирован под определенные сценарии: от локальной разработки до масштабируемого кластера с высокой доступностью.

2. Содержание раздела

2025 orionsoft. Все права защищены.

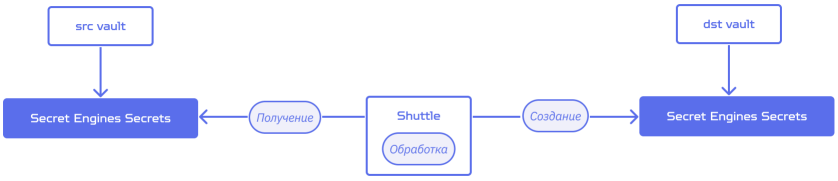
Миграция секретов из Vault в StarVault

В данном разделе описаны методы миграции секретов при переходе с HashiCorp Vault на StarVault.

1. Методы миграции

1.1. StarVault Shuttle

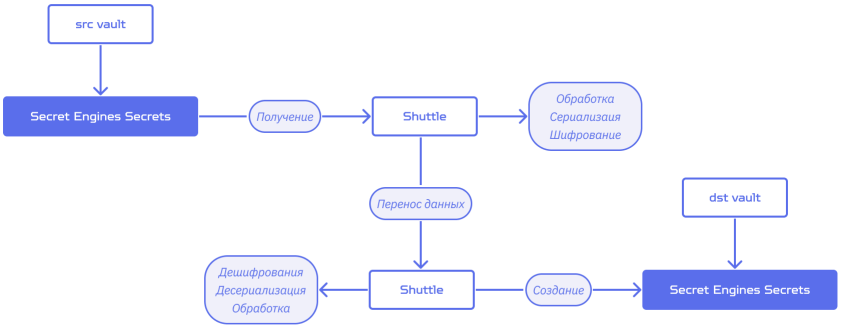
StarVault Shuttle — это утилита, упрощающая процесс миграции секретов между различными инстансами Vault. Миграцию можно выполнять как через командную строку (CLI), так и через веб-интерфейс. Утилита позволяет экспортировать секреты из исходного Vault и импортировать их в StarVault, обеспечивая целостность данных. Встроенные механизмы обработки ошибок и логирования повышают надежность миграции, а поддержка HTTPS гарантирует безопасную передачу данных.



1.1.1. Онлайн и офлайн миграция

StarVault Shuttle поддерживает как онлайн, так и офлайн миграцию данных:

- **Онлайн миграция** происходит в реальном времени в рамках одного сетевого контура.
- **Оффлайн миграция** предусматривает сериализацию и шифрование данных с помощью алгоритма AES-256. Отсутствие привязки к Storage Backend позволяет разворачивать различные конфигурации StarVault независимо от конфигурации исходного Vault.



1.1.2. Предварительные условия

- ✓ **Docker**. Утилита поставляется в контейнеризированном виде, поэтому необходимо наличие установленного Docker.
- ✓ **Минимальная версия Vault: 1.4.7**. Утилита поддерживает работу с Vault начиная с этой версии и выше. Убедитесь, что как исходный Vault, так и целевой StarVault соответствуют этой версии или новее.
- ✓ **Доступ к Vault**. Контейнер с утилитой должен быть запущен в одном сетевом контуре с исходным Vault и целевым StarVault.
- ✓ **Сетевое окружение**. Порты для доступа к Vault (по умолчанию 8200) должны быть открыты, и контейнер должен иметь возможность подключаться к инстансам Vault.

1.1.3. Использование

1.1.3.1. Запуск

Утилита доступна в виде образа контейнера (для получения аутентификационных данных обратитесь в службу поддержки StarVault). Чтобы загрузить последнюю версию утилиты, выполните следующую команду:

```bash
docker pull hub.nova-platform.io/starvault/starvault-shuttle:latest
```

Для запуска утилиты используйте следующую команду:

```bash
docker run -d --name starvault-shuttle -p 5000:5000 hub.nova-platform.io/starvault/starvault-shuttle:latest
```

1.1.3.2. Миграция через CLI

Чтобы начать взаимодействие через CLI, войдите внутрь запущенного контейнера:

```bash
docker exec -it starvault-shuttle /bin/sh
```

После входа используйте команду shuttle-cli, чтобы запустить процесс миграции в интерактивном режиме:

```bash
shuttle-cli
```

После запуска утилита предложит ввести необходимые данные для миграции:

- **Исходный Vault** - URL, unseal keys и токен
- **Целевой Vault** - URL, unseal keys и токен

- **Тип миграции** - онлайн или оффлайн. В случае с оффлайн миграцией утилита уточнит проводите ли вы экспорт или импорт секретов.
- **Путь миграции** - Утилита запросит, хотите ли вы перенести все секреты или только секреты с определенного пути. Вы можете ввести путь в формате secret/data/my-app или выбрать миграцию всех секретов.

Пример диалога при запуске `shuttle-cli`:

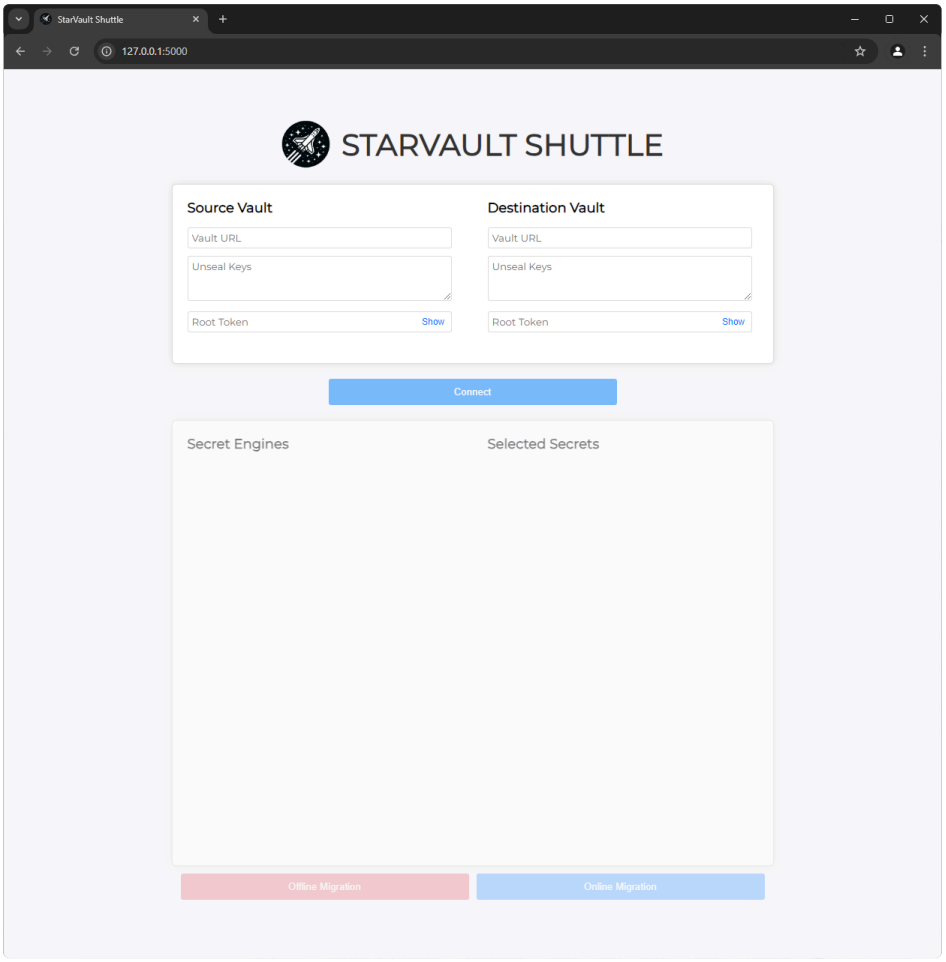
```bash
Welcome to Shuttle CLI.

Enter the source Vault URL: https://source-vault.example.com:8200
Enter the source Vault unseal keys: XXXXXXXXX,XXXXXXXXX,XXXXXXXXX
Enter the source Vault token: XXXXXXXXXXXX

Are you using online migration? (y/n): y

Enter the target Vault URL: https://source-vault.example.com:8200
Enter the target Vault unseal keys: XXXXXXXXX,XXXXXXXXX,XXXXXXXXX
Enter the target Vault token: XXXXXXXXXXXX

Establishing connection...
Source Vault connected successfully.
Destination Vault connected successfully.

Would you like to migrate all secrets? (y/n): n
Please specify the path to migrate (e.g., secret/data/path): secret/data/path

Migration in progress...

Migration completed successfully.
Total secrets migrated: 42.
```

Для выхода из контейнера используйте команду `exit`.

1.1.3.3. Миграция через веб-интерфейс

Для удобства пользователей утилита предоставляет веб-интерфейс, позволяющий выполнять миграцию секретов через веб-браузер.

После запуска контейнера веб-интерфейс будет доступен по адресу `http://localhost:5000`

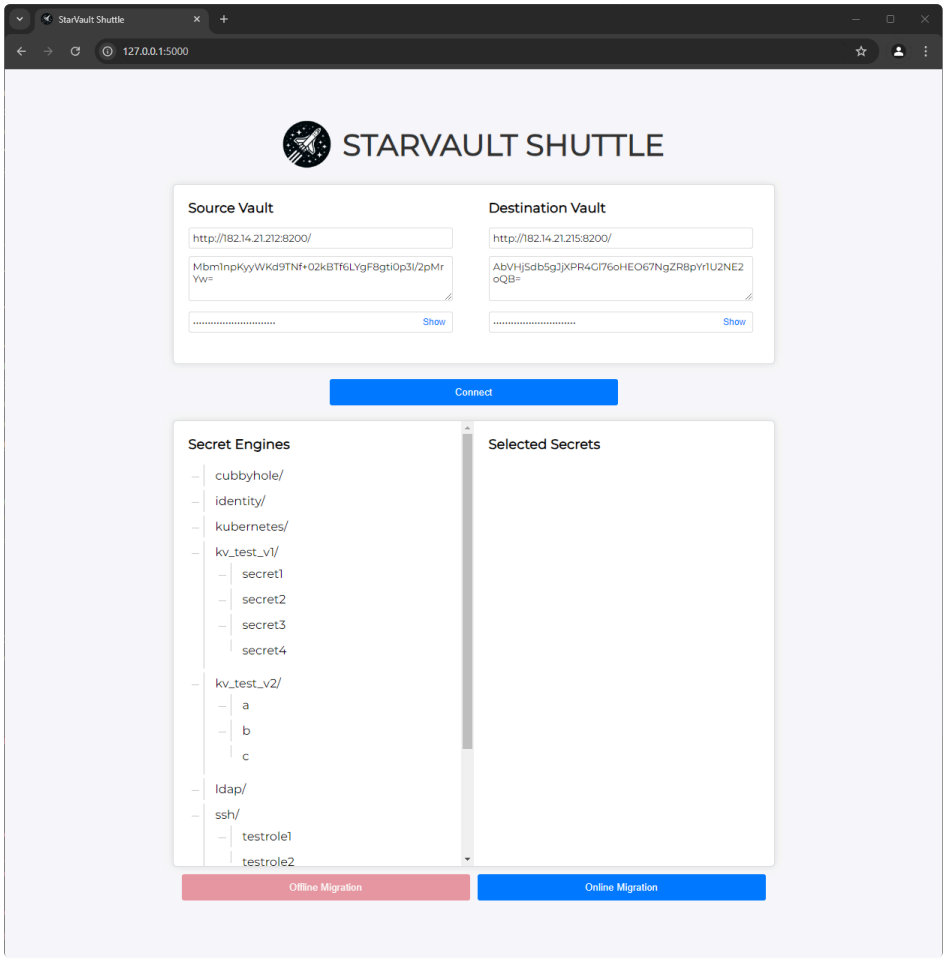


Шаг 1: Подключение

При входе в веб-интерфейс вам будет предложено указать параметры подключения к исходному Vault и целевому StarVault:

- **Исходный Vault** - URL, unseal keys и токен
- **Целевой StarVault** - URL, unseal keys и токен

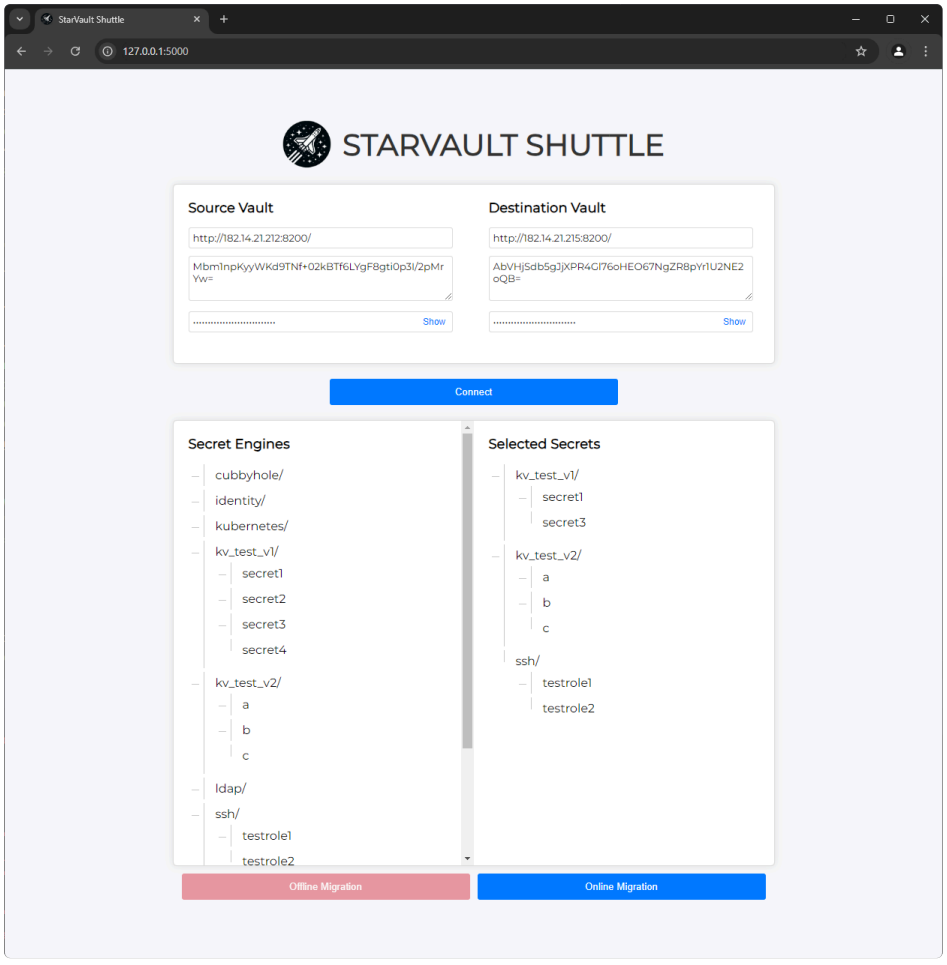
После указания параметров нажмите кнопку **"Connect"**, утилита выполнит подключение к Vault и выведет список секретов, доступных для переноса.



Шаг 2: Выбор секретов

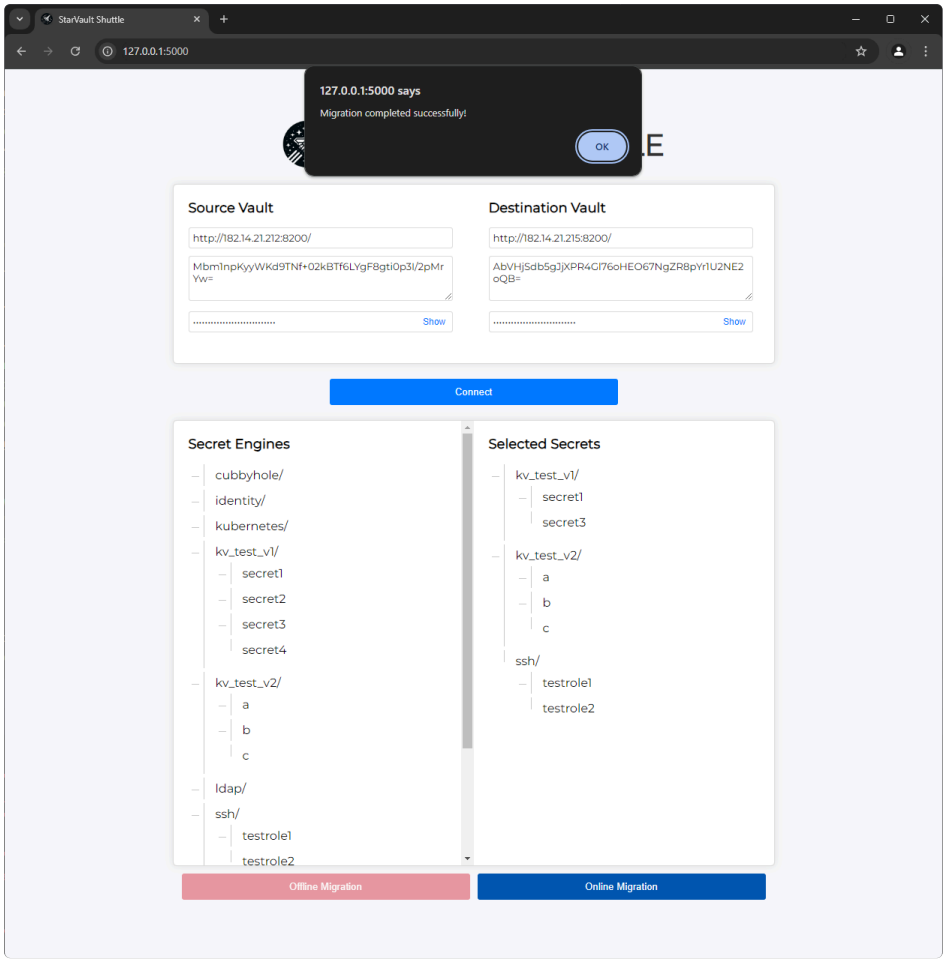
Далее необходимо выбрать секреты для миграции. Утилита выводит секреты в формате дерева в поле Secret Engines. При выборе родительского элемента в список на миграцию также будут включены все его дочерние элементы.

Если вы выбрали что-то лишнее, то по нажатию на элемент в поле Selected Secrets он будет исключен из списка.



Шаг 3: Запуск миграции

После выбора всех нужных секретов нажмите кнопку **"Online Migration"**. StarVault Shuttle выполнит миграцию и сообщит о результатах в всплывающем окне.



В случае с миграцией в оффлайн режиме вам необходимо указать параметры подключения только для одного из инстансов Vault. Для экспорта секретов укажите параметры только для Source Vault, для экспорта - параметры для Destination Vault. В первом случае при нажатии на кнопку **"Connect"** будет выведен список секретов для экспорта. При этом вы можете защитить экспортируемый файл паролем.

1.1.4. Решение проблем

Если во время работы утилиты возникают ошибки или миграция секретов не происходит должным образом, воспользуйтесь следующими советами для устранения неполадок.

1.1.4.1. Проверка логов

Работа утилиты глубоко логируется, эти логи могут помочь в анализе причин сбоев. Логи сохраняются в контейнере по пути `/shuttle/logs`. Если вам требуется направить логи в службу поддержки для дальнейшего анализа, вы можете выгрузить файл логов следующим образом:

```bash
docker cp starvault-shuttle:/shuttle/logs /path/to/local/directory
```

Также можно просмотреть логи контейнера с помощью команды:

```bash
docker logs starvault-shuttle
```

1.1.4.2. Проблемы с подключением к Vault

Если контейнер не может подключиться к исходному Vault или целевому StarVault, убедитесь, что:

- Указаны корректные значения для подключения.
- Инстансы Vault доступны из контейнера (контейнер должен находиться в одном сетевом контуре с инстансами Vault).
- Открыты необходимые порты (по умолчанию 8200) для взаимодействия с Vault.

Если все указано верно, попробуйте выполнить команду `ping` из контейнера:

```bash
docker exec -it starvault-shuttle ping source-vault.example.com
```

Если инстансы недоступны, проверьте настройки сети

1.1.4.3. Ошибки аутентификации

Если миграция прерывается с ошибкой аутентификации:

- Убедитесь, что указанные токены не истекли.
- Проверьте, что токены имеют достаточные привилегии для чтения и записи секретов.
- Попробуйте сгенерировать новые токены доступа через Vault и передать их утилите.

1.1.4.4. Проблемы с лицензией

Если утилита сообщает об ошибке, связанной с лицензией:

- Убедитесь, что переменная `LICENSE_KEY` передана корректно и содержит действительный лицензионный ключ.
- Проверьте, не истек ли срок действия лицензии.
- Если лицензия все еще действительна, обратитесь в поддержку для получения нового лицензионного ключа.

1.1.4.5. Дополнительные параметры для отладки

Для более подробного вывода логов запустите утилиту в режиме отладки, добавив флаг `--verbose` при запуске через CLI:

```bash
shuttle-cli --verbose
```

1.2. Миграция средствами Vault

1.2.1. Filesystem storage backend

1.2.1.1. Полный перенос хранилища

Для миграции данных, хранящихся с помощью файловой системы, необходимо перенести на целевой хост директорию, содержащую хранимые данные. По умолчанию это `/opt/vault/data`.

```
storage "file" {
  path = "/opt/vault/data"
}
```
BASH

1.2.2. Integrated Storage Backend (Raft)

1.2.2.1. Снапшоты

Сохраните снапшот исходного хранилища:

```
vault login
vault operator raft snapshot save data.snap
```
BASH

Восстановите снапшот на StarVault:

```
vault login
vault operator raft snapshot restore --force data.snap
```
BASH

Так как набор ключей у StarVault после инициализации другой, используйте флаг `--force`. После этого распечатывание хранилища и аутентификация будут происходить по unseal key и root token из исходного Vault.

Аналогичную операцию можно выполнить через веб-интерфейс: *Monitoring* → *Raft Storage* → *Snapshots* → *Download/Restore*. При восстановлении из снапшота необходимо отметить Force Restore.

1.2.3. Consul Storage Backend

1.2.3.1. Подключение нод к существующему кластеру

Если необходимо использовать существующий Consul-кластер, новые ноды StarVault можно подключить к нему, а старые ноды Vault впоследствии вывести из кластера.

```
storage "consul" {
  address = "172.22.1.7:8500"
  path    = "vault/"
}
```
BASH

1.2.3.2. Снапшоты

По аналогии с **Integrated Storage Backend**, данные можно перенести в новый Consul-кластер при помощи снапшотов.

Сохраните снапшот исходного хранилища (укажите адрес и порт HTTP-агента Consul):

```bash
consul snapshot save -http-addr=172.20.1.6:8500 consul_data.snap
```

Восстановите снапшот на StarVault:

```bash
consul snapshot restore -http-addr=172.20.1.7:8500 consul_data.snap
```

1.2.4. External Storage Backend

Этот раздел обобщает остальные внешние хранилища, поскольку принцип для них одинаков - подключение новых нод StarVault к существующему хранилищу. Если необходимо использовать новое внешнее хранилище, миграцию данных необходимо осуществлять средствами самого хранилища. Рассмотрим перенос данных для Vault + PostgreSQL.

1.2.4.1. Подключение к существующей базе данных

В конфигурации StarVault определите хранилище, идентичное исходному Vault. После этого оба Vault будут взаимодействовать с одной базой данных.

```bash
storage "postgresql" {
  connection_url = "postgres://user:pass@123.123.123.123:5432/vault"
}
```

1.2.4.2. Миграция средствами PostgreSQL

В качестве примера используются два инстанса Vault + PostgreSQL со следующими конфигурациями:

Исходный Vault

```bash
storage "postgresql" {
connection_url = "postgres://user:pass@172.20.1.10:5432/vault"
}
```

StarVault

```bash
storage "postgresql" {
connection_url = "postgres://user:pass@172.20.1.11:5432/vault"
}
```

Перенесите данные с помощью pg_dump. Сначала создайте резервную копию исходной базы данных:

```bash
su - postgres
pg_dump -Fc vault > /tmp/data.dump
```

Далее восстановите резервную копию на базе данных StarVault:

```bash
su - postgres
pg_restore --if-exists -d vault /tmp/data.dump
```

2025 orionsoft. Все права защищены.

Установка в тестовом режиме

Чтобы запустить StarVault в качестве сервера в тестовом режиме (dev-режиме) выполните команду: `starvault server -dev`. Сервер в тестовам режиме не требует дополнительных настроек, и локальный `starvault` CLI будет аутентифицирован для общения с ним. Это позволяет легко экспериментировать с StarVault или запустить экземпляр StarVault для разработки. Каждая функция StarVault доступна в режиме "dev". Флаг `-dev` изменяет многие настройки на небезопасные настройки по умолчанию.

> Не запускайте сервер в тестовом режиме в боевой среде. Dev-сервер небезопасен и теряет данные при каждом перезапуске (поскольку хранит данные в памяти). Dev режим предназначен только для разработки или экспериментов.

1. Свойства

Свойства dev-сервера (некоторые из них могут быть переопределены с помощью флагов командной строки или путем указания конфигурационного файла):

- Инициализация и распечатывание — сервер будет автоматически инициализирован и распечатан. Для распечатывания не нужно использовать оператор `starvault operator unseal`. Сервер сразу готов к использованию.
- Хранение в памяти — все данные хранятся в памяти, в зашифрованном виде. Сервер StarVault не требует никаких разрешений на файлы.
- Привязка к локальному адресу без TLS — сервер по умолчанию прослушивает адрес `127.0.0.1:8200` без TLS.
- Автоматическая аутентификация — сервер сохраняет токен корневого доступа, доступ к `starvault` CLI сразу предоставлен. Если подключаетесь к StarVault через API, вам нужно будет пройти аутентификацию, используя токен распечатывания.
- Одиночный ключ для распечатывания — сервер инициализируется с одним ключом для распечатывания. Хранилище уже распечатано, но для экспериментов с запечатыванием/разпечатыванием, потребуется только один выпущенный ключ.
- Хранилище значений ключей смонтировано — механизм секретов v2 KV установлен по адресу `secret/`. Имейте в виду, что v2 KV отличается от v1 KV. Чтобы использовать v1, примените флаг `-dev-kv-v1`.

2. Пример использования

Dev-сервер следует использовать для экспериментов с функциями StarVault. Например, различные методы авторизации, механизмы секретов, устройства аудита и т.д.

Помимо экспериментов, dev-сервер очень легко автоматизировать для сред разработки.

2025 orionsoft. Все права защищены.