Установка в среде выполнения контейнеров

Предварительные требования

- Установлена среда выполнения контейнеров, например, docker или podman.
- Установлен, в зависимости от среды исполнения, Docker Compose или Podman Compose.

> Обычно для Docker compose-оболочка устанавливается вместе со средой выполнения.
>
> Для podman в rh-based дистрибутивах выполните следующее:
>
> 1. Установите репозиторий epel:
>
> ```
> dnf install -y epel-release
> ```
>
> 2. Установите podman-compose:
>
> ```
> dnf install -y podman-compose
> ```

Порядок действий

1. Добавьте разрешающее правило на сетевом экране для TCP-порта 8200, например для firewalld:

```
firewall-cmd --add-port=8200/tcp --permanent
firewall-cmd --reload
```

> В примере используется порт 8200, но вы можете использовать любой другой порт.
> Конфигурационный файл после установки расположен по пути **/etc/starvault.d/starvault.hcl**.

2. Создайте каталог для хранения данных и конфигурации:

```
mkdir -p /opt/starvault/{data,tls}
```

3. Создайте самодписанный сертификат или пропустите текущий этап если будет использоваться сертификат выданный вашим Центром сертификации:

```
openssl req -nodes -x509 -sha256 -newkey rsa:4096 \
-keyout /opt/starvault/tls/starvault.key \
-out /opt/starvault/tls/starvault.crt \
-days 356 \
-subj "/C=RU/O=Orionsoft/CN=starvault" \
```

```
-extensions san \
-config <( \
echo '[req]'; \
echo 'distinguished_name=req'; \
echo '[san]'; \
echo 'subjectAltName=DNS:localhost,DNS:starvault,IP:127.0.0.1')
```

StarVault может быть запущен без использования SSL. Если вам не требуется шифрование, то пропустите текущий этап.

4. Подготовьте конфигурационной файл **/opt/starvault/config.hcl**:

BASH

```
cat << EOF > /opt/starvault/config.hcl
ui = true
storage "file" {
    path = "/opt/starvault/data"
}
listener "tcp" {
    address = "0.0.0.0:8200"
    tls_cert_file = "/opt/starvault/tls/starvault.crt"
    tls_key_file = "/opt/starvault/tls/starvault.key"
}
EOF
```

Для запуска StarVault без SSL замените параметры в конфигурационной файле `tls_cert_file` и `tls_key_file` на `tls_disable = "true"`.

5. Создайте Compose манифест:

BASH

```
cat << EOF > /opt/starvault/compose.yml
services:
  starvault:
    image: hub.orionsoft.ru/public/starvault:v1.2.0
    volumes:
      - /opt/starvault/data:/opt/starvault/data
      - /opt/starvault/tls:/opt/starvault/tls
      - /opt/starvault/config.hcl:/opt/starvault/config.hcl
    ports:
      - 8200:8200
      - 8201:8201
    cap_add:
      - IPC_LOCK
    entrypoint: starvault server -config /opt/starvault/config.hcl
EOF
```

Для запуска с помощью **podman в режиме rootless** нужно явно задать права на доступ к прокинутым томам для пользователя, от имени которого будет запускаться контейнер (в примере ниже используется **admin**):

1. Смените владельца каталога для starvault:

```
chown admin:admin -R /opt/starvault
```

2. Добавьте параметры доступа к списку подключаемых томов в compose.yml:

```
- /opt/starvault/data:/opt/starvault/data:z
- /opt/starvault/tls:/opt/starvault/tls:z
- /opt/starvault/config.hcl:/opt/starvault/config.hcl:z
```

6. Поднимите контейнер:
   - Для docker:

```
docker compose --file /opt/starvault/compose.yml up
```

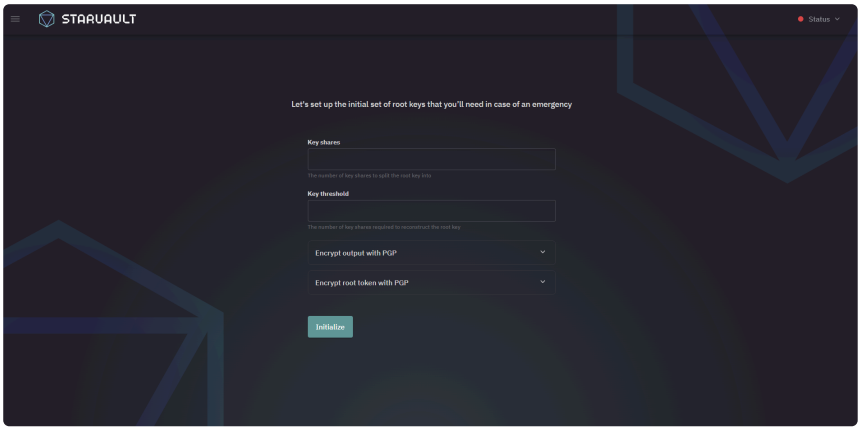
   - Для podman:

```
podman compose --file /opt/starvault/compose.yml up
```

7. Выполните инициализацию:

Инициализация - это процесс настройки StarVault. Она происходит только один раз, когда сервер запускается с новым бэкендом, который никогда ранее не использовался с StarVault.

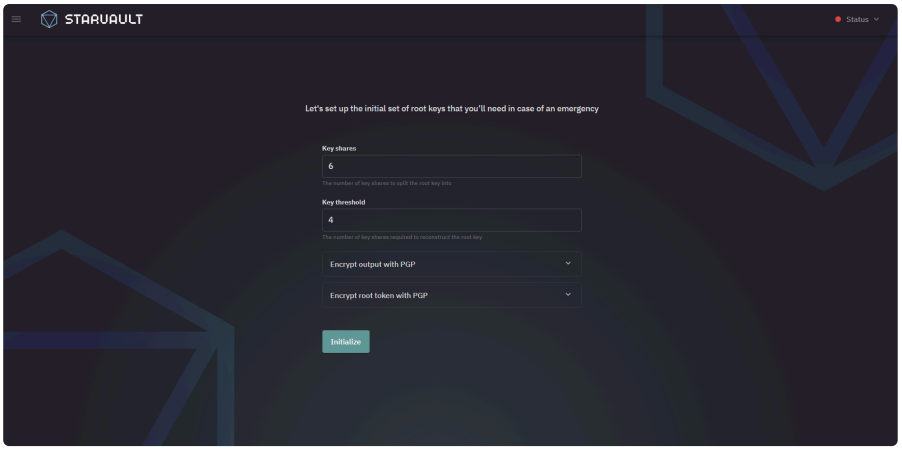
a. В браузере перейдите по адресу **https://SERVER-IP:8200**, где **SERVER-IP** - адрес сервера, на котором развертывается StarVault.



b. В поле **Key shares** введите количество долей ключа на которое будет разделён ключ распечатки

c. В поле **Key threshold** введите количество долей, которого будет достаточно для расшифровывания корневого ключа

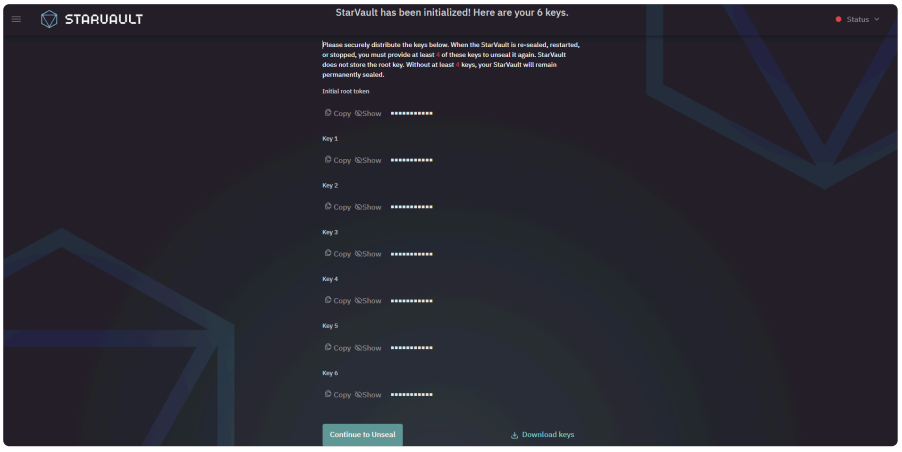
d. Нажмите **[Initialize]** для инициализации StarVault.



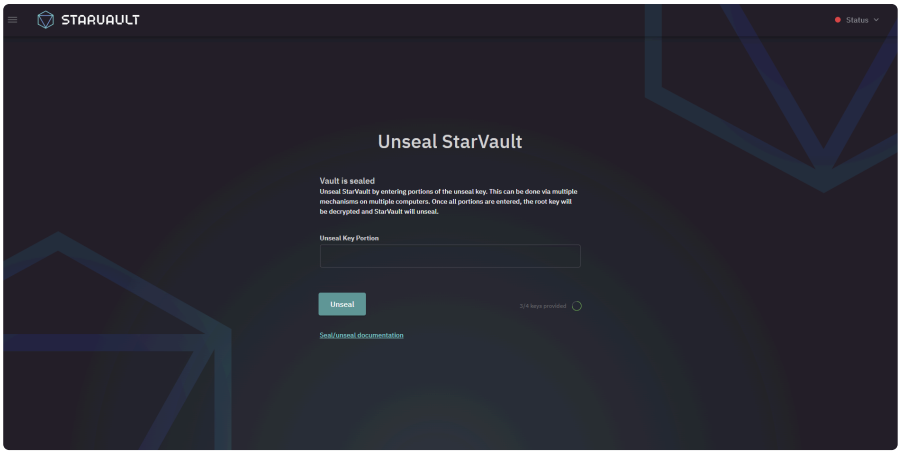
8. После того, как инициализация будет выполнена, на экране будут выведены все части ключа и корневой токен. Сохраните эти данные и нажмите **[Continue to Unseal]** для перехода к распечатыванию хранилища.

> Части ключа и начальный корневой токен крайне важны. Это единственный раз, когда все эти данные известны StarVault, а также единственный раз, когда они должны быть так близко друг к другу.

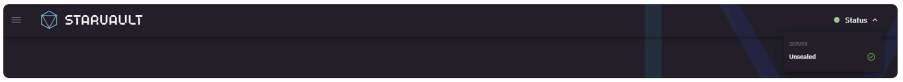
> Для быстрого сохранения вы можете загрузить токен и доли ключа в формате json, нажав **[Download keys]**.



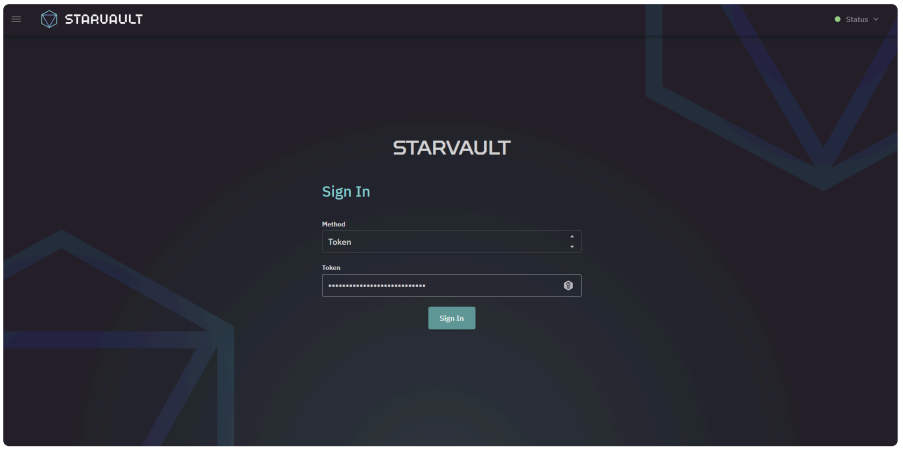
9. Поочередно введите доли ключа распечатки в поле **Unseal Key Portion** в необходимом количестве. После ввода каждой доли нажимайте Unseal



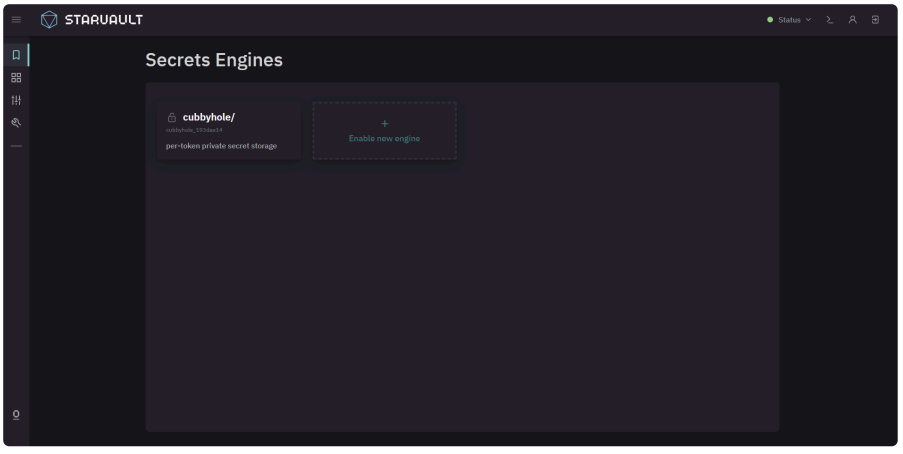
10. После ввода необходимого количества долей, хранилище будет распечатано, на что указывает статус:



11. На экране логина убедитесь, что выбран метод аутентификации **Token** и введите корневой токен, полученный после инициализации. Нажмите **[Sign In]** для входа.



12. При успешном входе вы увидите Secrets Engines.



Установка в режиме высокой доступности (HA)

StarVault поддерживает мультисерверный режим для обеспечения высокой доступности. Этот режим защищает от сбоев благодаря работе нескольких серверов StarVault. Режим высокой доступности включается автоматически при использовании хранилища данных, которое его поддерживает.

Вы можете определить, поддерживает ли хранилище данных режим высокой доступности ("HA"), запустив сервер и увидев, выводится ли рядом с информацией о хранилище данных сообщение "(HA available)". Если это так, то StarVault будет автоматически использовать режим HA. Эта информация также доступна на странице Конфигурация.

Чтобы достичь высокой доступности, один из узлов StarVault получает блокировку в системе хранения данных. Этот узел затем принимает роль активного узла, в то время как остальные узлы переходят в режим ожидания. Если резервные узлы получают запросы, они будут перенаправлять их или переадресовывать клиентов в соответствии с настройками и текущим состоянием кластера.

1. Использование бэкенда хранилища Raft

Развертывание StarVault в режиме высокой доступности с интегрированным хранилищем Raft требует не менее трёх серверов StarVault. В ином случае не набирается кворум и распечатывание хранилища невозможно.

Предварительные требования:

- На сервер установлена поддерживаемая операционная система (RedOS, Astra Linux, AlmaLinux);
- На сервер скопирован пакет с дистрибутивом StarVault.
- Наличие сертификатов для каждого узла в кластере Raft, а также сертификата корневого центра сертификации.

1.1. Предварительная подготовка инфраструктуры

В данном сценарии будет описан процесс построения кластера StarVault, состоящий из следующих компонентов:

- 3 узлов StarVault: 1 активный, 2 резервных.

Данная архитектура **не является** рекомендованной для применения в продуктивной среде и предназначена для демонстрации и тестирования.

Рекомендуемая архитектура описана в разделе Интегрированное хранилище Raft устройства StartVault.

Данная инструкция предполагает взаимодействие между серверами с использованием FQDN. Для этого необходимо настроить DNS-сервер с соответствующими A-записями. В целях демонстрации далее будут использоваться следующие сопоставления FQDN и IP-адресов узлов:

ID узла	FQDN	IP-адрес
raft-node-1	raft-node-1.vlab.local	10.252.11.10
raft-node-2	raft-node-2.vlab.local	10.252.11.11
raft-node-3	raft-node-3.vlab.local	10.252.11.12

1.2. Подготовка необходимых сертификатов

Для настройки TLS необходимо наличие набора сертификатов и ключей, расположенных в каталоге **/opt/starvault/tls**:

- Сертификат корневого центра сертификации, подписавшего сертификат StarVault TLS. В данном сценарии он будет иметь имя **starvault-ca.pem**.
- Сертификаты узлов Raft. В данном сценарии в кластер будет добавлено три узла, для которых будут созданы следующие сертификаты:
  - **node-1-cert.pem**
  - **node-2-cert.pem**
  - **node-3-cert.pem**
- Закрытые ключи сертификатов узлов:
  - **node-1-key.pem**
  - **node-2-key.pem**
  - **node-3-key.pem**

В этом сценарии создадим корневой сертификат, а также набор самоподписанных сертификатов для каждого узла.

Хотя самоподписанные сертификаты можно использовать для экспериментов с развертыванием и запуском StarVault, мы настоятельно рекомендуем заменить их сертификатами, созданными и подписанными соответствующим центром сертификации.

Порядок действий

1. На первом узле перейдите в каталог **/opt/starvault/tls/** (если каталог не существует - создайте его):

```bash
mkdir -p /opt/starvault/tls
cd /opt/starvault/tls/
```

2. Сгенерируйте ключ для корневого сертификата:

```bash
openssl genrsa 2048 > starvault-ca-key.pem
```

3. Выпустите корневой сертификат:

```bash
openssl req -new -x509 -nodes -days 3650 -key starvault-ca-key.pem -out
starvault-ca.pem

Country Name (2 letter code) [XX]:RU
State or Province Name (full name) []:
Locality Name (eg, city) [Default City]:Moscow
Organization Name (eg, company) [Default Company Ltd]:Orionsoft
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server hostname) []:vlab.local
```

Атрибуты сертификата указаны для примера.

4. Создайте конфигурационные файлы, содержащие subjectAltName (SAN), для выпуска сертификатов узлов. Например, для узла **raft-node-1** файл будет выглядеть следующим образом:

```bash
cat << EOF > node-1.cnf
[v3_ca]
subjectAltName = @alt_names
[alt_names]
DNS.1 = raft-node-1.vlab.local ①
IP.1 = 10.252.11.10 ②
IP.2 = 127.0.0.1
EOF
```

① FQDN соответствующего узла

② IP-адрес соответствующего узла

SAN должен содержать корректный FQDN и IP-адрес соответствующего узла.

Конфигурационный файл должен быть создан для каждого узла, который планируется добавить в кластер.

5. Для каждого узла сформируйте файл запроса:

```
BASH
openssl req -newkey rsa:2048 -nodes -keyout node-1-key.pem -out node-1-
csr.pem -subj "/CN=raft-node-1.vlab.local"
openssl req -newkey rsa:2048 -nodes -keyout node-2-key.pem -out node-2-
csr.pem -subj "/CN=raft-node-2.vlab.local"
openssl req -newkey rsa:2048 -nodes -keyout node-3-key.pem -out node-3-
csr.pem -subj "/CN=raft-node-3.vlab.local"
```

6. Выпустите сертификаты на основании запросов:

```
BASH
openssl x509 -req -set_serial 01 -days 3650 -in node-1-csr.pem -out node-1-
cert.pem -CA starvault-ca.pem -CAkey starvault-ca-key.pem -extensions v3_ca
-extfile ./node-1.cnf
openssl x509 -req -set_serial 01 -days 3650 -in node-2-csr.pem -out node-2-
cert.pem -CA starvault-ca.pem -CAkey starvault-ca-key.pem -extensions v3_ca
-extfile ./node-2.cnf
openssl x509 -req -set_serial 01 -days 3650 -in node-3-csr.pem -out node-3-
cert.pem -CA starvault-ca.pem -CAkey starvault-ca-key.pem -extensions v3_ca
-extfile ./node-3.cnf
```

7. После выпуска сертификатов можно поступить двумя способами, в зависимости от того, каким образом планируется присоединять узлы к кластеру:
   - Для автоматического подключения узлов скопируйте на каждый узел:
     - Файл сертификата этого узла.
     - Файл ключа этого узла.
     - Файл корневого сертификата.

     Например:

     ```
     scp ./node-2-key.pem ./node-2-cert.pem ./starvault-ca.pem  raft-node-
     2.vlab.local:/opt/starvault/tls
     scp ./node-3-key.pem ./node-3-cert.pem ./starvault-ca.pem  raft-node-
     3.vlab.local:/opt/starvault/tls
     ```

     Если каталог **/opt/starvault/tls** на целевых узлах отсутствует - создайте его.

   - Для ручного подключения узлов скопируйте сертификаты и ключи узлов, а также корневой сертификат на устройство, с которого будете подключаться к WEB-UI StarVault для добавления узлов.

1.3. Развертывание кластера с Raft

▶ Способ 1. С автоматическим добавлением узлов в кластер

▶ Способ 2. С ручным добавлением узлов в кластер

Установка в Kubernetes

StarVault можно развернуть в Kubernetes с помощью официального Helm-чарта. Helm-чарт позволяет пользователям развернуть StarVault в различных конфигурациях:

- **Dev** — одиночный сервер StarVault в оперативной памяти для тестирования StarVault
- **Standalone** — одиночный сервер StarVault, сохраняющий данные в томе с помощью бэкенда файлового хранилища (по умолчанию)
- **High-Availability** (HA) — кластер серверов StarVault, использующих бэкенд хранения высокой доступности

1. Сценарии использования

- **Запуск сервиса StarVault**. Кластер серверов StarVault может работать непосредственно в Kubernetes и использоваться приложениями как внутри Kubernetes, так и внешними по отношению к Kubernetes, если они умеют взаимодействовать с сервером по сети.
- **Доступ к секретам и их хранение**. Приложения, которые используют сервис StarVault, запущенный в Kubernetes, могут получать доступ к секретам StarVault и хранить их с помощью различных механизмов секретов и методов аутентификации.
- **Запуск сервиса StarVault в режиме высокой доступности**. Благодаря атрибутам affinity у подов, высокодоступному бэкенду хранения и автоматическому распечатыванию, StarVault в Kubernetes может стать сервисом высокой доступности.
- **Шифрование как сервис**. Приложения, которые работают с сервисом StarVault в Kubernetes, могут использовать модуль работы с транзитными секретами по модели «шифрование как сервис», тем самым переложив на StarVault задачу шифрования данных перед сохранением.
- **Журналы аудита StarVault**. Операторы могут прикрепить к кластеру StarVault постоянный том для хранения журналов аудита.
- StarVault может работать непосредственно в Kubernetes. Поэтому любой другой инструмент созданный для Kubernetes, может использовать StarVault.

2. Начало работы с StarVault и Kubernetes

Протестировать работу StarVault с Kubernetes в различных средах можно разными способами.

2.1. Краткое сравнение интеграций

Существует три интеграции, которые помогают рабочим нагрузкам Kubernetes беспрепятственно получать секреты из StarVault, при этом не нужно изменять приложение, чтобы подключаться к StarVault напрямую. У каждой интеграции разные задачи. Ниже кратко описаны преимущества интеграций StarVault.

2.1.1. StarVault Secrets Operator

- Удобен для разработчиков приложений. Рабочие нагрузки могут монтировать секреты Kubernetes без каких-либо специфических для StarVault настроек.
- Сниженная нагрузка на StarVault. Секреты синхронизируются для каждого CRD [Custom Resource Definition], а не для каждого пода, который эти секреты потребляет.
- Улучшенная доступность секретов StarVault. Секреты Kubernetes выступают в качестве долговременного кэша секретов StarVault внутри кластера.