Механизм секретов базы данных Oracle

Данный механизм секретов является частью механизма секретов базы данных. Если вы еще не читали страницу о бэкенде базы данных, пожалуйста, сделайте это сейчас, так как там объясняется, как настроить бэкенд базы данных и дается обзор функционирования движка.

Oracle - один из поддерживаемых плагинов для механизма секретов баз данных. Он способен динамически генерировать учетные данные на основе настроенных ролей для баз данных Oracle. Он также поддерживает статические роли.

1. Возможности

Плагин для баз данных Oracle не включен в основное дерево кода StarVault и может быть найден в собственном git-репозитории здесь: hashicorp/vault-plugin-database-oracle

Этот плагин не совместим с Alpine Linux из коробки.

Название плагина	**Возможность ротации root**	**Динамические роли**	**Кастомизация имен пользователей**
Customizable (see: Custom Plugins)	Да	Да	Да

2. Настройка

Плагин для баз данных Oracle не входит в основное дерево кода StarVault и может быть найден в собственном git-репозитории здесь: hashicorp/vault-plugin-database-oracle

Для linux/amd64 предварительно собранные двоичные файлы можно найти на странице релизов

Перед запуском плагина вам потребуется установить библиотеку Oracle Instant Client. Их можно загрузить с сайта Oracle. Библиотеки должны быть помещены в путь поиска библиотек по умолчанию или определены в конфигурационных файлах ld.so.conf.

Следующие привилегии необходимы плагину для минимальной функциональности. Дополнительные привилегии могут потребоваться в зависимости от SQL, настроенного в ролях базы данных.

```
GRANT CREATE USER to starvault WITH ADMIN OPTION;
GRANT ALTER USER to starvault WITH ADMIN OPTION;
GRANT DROP USER to starvault WITH ADMIN OPTION;
GRANT CONNECT to starvault WITH ADMIN OPTION;
GRANT CREATE SESSION to starvault WITH ADMIN OPTION;
GRANT SELECT on gv_$session to starvault;
GRANT SELECT on v_$sql to starvault;
GRANT ALTER SYSTEM to starvault WITH ADMIN OPTION;
```

> StarVault требует `ALTER SYSTEM` для завершения пользовательских сессий при отзыве пользователей. Это можно заменить сохраненной процедурой и предоставить ее пользователю-администратору StarVault.

Если вы используете StarVault с включенной функцией mlock, вам нужно включить возможность ipc_lock для бинарных файлов плагина.

---

1. Включите механизм секретов базы данных, если он еще не включен:

   ```
   $ starvault secrets enable database
   Success! Enabled the database secrets engine at: database/
   ```

   По умолчанию механизм секретов будет включаться по имени движка. Чтобы включить механизм секретов по другому пути, используйте аргумент `-path`.

2. Загрузите и зарегестрируйте плагин

   ```
   $ starvault write sys/plugins/catalog/database/oracle-database-plugin \
       sha256="..." \
       command=vault-plugin-database-oracle
   ```

3. Настройте StarVault с помощью соответствующего плагина и информации о подключении:

   ```
   $ starvault write database/config/my-oracle-database \
       plugin_name=oracle-database-plugin \
       connection_url="
   {{username}}/{{password}}@localhost:1521/OraDoc.localhost" \
       allowed_roles="my-role" \
       username="VAULT_SUPER_USER" \
       password="myreallysecurepassword"
   ```

   Если Oracle использует SSL, смотрите пример подключения с использованием SSL.

   Если в используемой вами версии Oracle есть контейнерная база данных, то в поле `connection_url` вам нужно будет подключиться к одной из подключаемых баз данных,

а не к контейнерной базе данных.

4. Настоятельно рекомендуется немедленно сменить пароль пользователя `root`, подробнее см. в разделе Ротация корневых учетных данных. Это гарантирует, что только StarVault сможет получить доступ к `root` пользователю, которого StarVault использует для работы с динамическими и статическими учетными данными.

> **Будьте внимательны**
>
> Пароль пользователя `root` будет недоступен после ротации, поэтому настоятельно рекомендуется создать пользователя для StarVault, не использовать фактического пользователя `root`.

5. Настройте роль, которая сопоставляет имя в StarVault с оператором SQL, который нужно выполнить для создания учетной записи базы данных:

```bash
$ starvault write database/roles/my-role \
    db_name=my-oracle-database \
    creation_statements='CREATE USER {{username}} IDENTIFIED BY "
{{password}}"; GRANT CONNECT TO {{username}}; GRANT CREATE SESSION TO
{{username}};' \
    default_ttl="1h" \
    max_ttl="24h"
```

> `creation_statements` могут быть указаны в файле и интерпретированы StarVault CLI с помощью символа `@`:

```bash
$ starvault write database/roles/my-role \
    creation_statements=@creation_statements.sql \
    ...
```

Дополнительные сведения см. в документации по командам.

2.1. Подключение с помощью SSL

Если сервер Oracle, к которому пытается подключиться StarVault, использует SSL-приемник, плагин базы данных потребует дополнительной настройки с помощью параметра `connection_url`:

```bash
starvault write database/config/oracle \
  plugin_name=vault-plugin-database-oracle \
  connection_url='{{username}}/{{password}}@(DESCRIPTION=(ADDRESS=
(PROTOCOL=tcps)(HOST=<host>)(PORT=<port>))(CONNECT_DATA=(SERVICE_NAME=
<service_name>))(SECURITY=(SSL_SERVER_CERT_DN="<cert_dn>")(MY_WALLET_DIRECTORY=
<path_to_wallet>)))' \
```

```
  allowed_roles="my-role" \
  username="admin" \
  password="password"
```

Например, отличительное имя сертификата SSL-сервера и путь к Oracle Wallet, который будет использоваться для подключения и проверки, можно настроить с помощью:

BASH

```
starvault write database/config/oracle \
  plugin_name=vault-plugin-database-oracle \
  connection_url='{{username}}/{{password}}@(DESCRIPTION=(ADDRESS=
(PROTOCOL=tcps)(HOST=orionsoft.ru)(PORT=1523))(CONNECT_DATA=(SERVICE_NAME=ORCL))
(SECURITY=(SSL_SERVER_CERT_DN="CN=orionsoft.ru,OU=TestCA,O=orionsoft=ru")
(MY_WALLET_DIRECTORY=/etc/oracle/wallets)))' \
  allowed_roles="my-role" \
  username="admin" \
  password="password"
```

2.1.1. Разрешения для кошелька

Кошельки, используемые при подключении по SSL, должны быть доступны на каждом сервере StarVault при использовании кластеров высокой доступности.

Кошелек, используемый StarVault, должен находиться в хорошо известном месте с соответствующими правами доступа к файловой системе. Например, если StarVault работает от имени пользователя starvault, каталог кошелька может быть настроен следующим образом:

BASH

```
mkdir -p /etc/starvault/wallets
cp cwallet.sso /etc/starvault/wallets/cwallet.sso
chown -R starvault:starvault /etc/starvault
chmod 600 /etc/starvault/wallets/cwallet.sso
```

2.2. Использование TNS имен

Файл `tnsnames.ora` и переменная окружения, используемые при подключении по SSL, должны быть доступны на каждом сервере StarVault при использовании кластеров высокой доступности.

StarVault может опционально использовать имена TNS в строке соединения при подключении к базам данных Oracle с помощью файла `tnsnames.ora`. Пример файла `tnsnames.ora` может выглядеть следующим образом:

```
AWSEAST=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = hashicorp.us-east-1.rds.amazonaws.com)
```

```
(PORT = 1523))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SID = ORCL)
  )
  (SECURITY =
      (SSL_SERVER_CERT_DN =
"CN=hashicorp.rds.amazonaws.com/OU=RDS/O=Amazon.com/L=Seattle/ST=Washington/C=US
")
      (MY_WALLET_DIRECTORY = /etc/oracle/wallet/east)
  )
)

AWSWEST=
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCPS)(HOST = hashicorp.us-west-1.rds.amazonaws.com)
(PORT = 1523))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SID = ORCL)
  )
  (SECURITY =
      (SSL_SERVER_CERT_DN =
"CN=hashicorp.rds.amazonaws.com/OU=RDS/O=Amazon.com/L=Seattle/ST=Washington/C=US
")
      (MY_WALLET_DIRECTORY = /etc/oracle/wallet/west)
  )
)
```

Чтобы настроить StarVault на использование имен TNS, установите на сервере StarVault следующую переменную среды:

```
TNS_ADMIN=/path/to/tnsnames/directory
```

Если StarVault выдает ошибку "не удалось открыть файл", проверьте, доступна ли переменная окружения `TNS_ADMIN` на сервере StarVault.

Используйте псевдоним в параметре `connection_url` в конфигурации базы данных:

```
starvault write database/config/oracle-east \
    plugin_name=vault-plugin-database-oracle \
    connection_url="{{username}}/{{password}}@AWSEAST" \
    allowed_roles="my-role" \
    username="VAULT_SUPER_USER" \
    password="myreallysecurepassword"

starvault write database/config/oracle-west \
    plugin_name=vault-plugin-database-oracle \
```

```
    connection_url="{{username}}/{{password}}@AWSWEST" \
    allowed_roles="my-role" \
    username="VAULT_SUPER_USER" \
    password="myreallysecurepassword"
```

3. Использование

3.1. Динамические учетные данные

После того как механизм секретов настроен и у пользователя/машины есть токен StarVault с соответствующими правами, он может генерировать учетные данные.

1. Сгенерируйте новые учетные данные, считав из конечной точки `/creds` имя роли:

```
$ starvault read database/creds/my-role
```

BASH

Key	Value
`lease_id`	`database/creds/my-role/2f6a614c-4aa2-7b19-24b9-ad944a8d4de6`
`lease_duration`	`1h`
`lease_renewable`	`true`
`password`	`yRUSyd-vPYDg5NkU9kDg`
`username`	`V_VAULTUSE_MY_ROLE_SJJUK3Q8W3BKAYAN8S62_1602543009`

4. API

Полный список настраиваемых параметров можно посмотреть на странице API плагина для баз данных Oracle.

Дополнительную информацию о HTTP API движка секретов баз данных можно найти на странице API движка секретов баз данных.

Механизм секретов базы данных PostgreSQL

PostgreSQL - один из поддерживаемых плагинов для механизма секретов баз данных. Этот плагин генерирует учетные записи базы данных динамически на основе настроенных ролей для базы данных PostgreSQL, а также поддерживает статические роли.

Дополнительную информацию о настройке механизма секретов базы данных см. в документации по механизму секретов базы данных.

Механизм секретов PostgreSQL использует pgx, ту же библиотеку баз данных, что и бэкэнд хранилища PostgreSQL. Параметры строки соединения, включая параметры SSL, можно найти в документации по строке соединения pgx и PostgreSQL.

1. Возможности

Название плагина	Возможность ротации root	Динамические роли	Кастомизация имен пользователей
`postgresql-database-plugin`	Да	Да	Да

2. Настройка

1. Включите механизм секретов базы данных, если он еще не включен:

```
$ starvault secrets enable database
Success! Enabled the database secrets engine at: database/
```

   По умолчанию движок secrets будет включаться по имени движка. Чтобы включить движок secrets по другому пути, используйте аргумент `-path`.

2. Настройте StarVault с помощью соответствующего плагина и информации о подключении:

```
$ starvault write database/config/my-postgresql-database \
    plugin_name="postgresql-database-plugin" \
    allowed_roles="my-role" \
    connection_url="postgresql://{{username}}:
{{password}}@localhost:5432/database-name" \
```

```
    username="vaultuser" \
    password="vaultpass" \
    password_authentication="scram-sha-256"
```

3. Настройте роль, которая сопоставляет имя в StarVault с оператором SQL, чтобы выполнить создание учетной записи базы данных:

```
BASH
$ starvault write database/roles/my-role \
    db_name="my-postgresql-database" \
    creation_statements="CREATE ROLE \"{{name}}\" WITH LOGIN PASSWORD
'{{password}}' VALID UNTIL '{{expiration}}'; \
        GRANT SELECT ON ALL TABLES IN SCHEMA public TO \"{{name}}\";" \
    default_ttl="1h" \
    max_ttl="24h"
Success! Data written to: database/roles/my-role
```

3. Использование

После того как механизм секретов настроен и у пользователя/машины есть токен StarVault с соответствующими правами, он может генерировать учетные данные.

1. Сгенерируйте новые учетные данные, считав из конечной точки `/creds` имя роли:

```
BASH
$ starvault read database/creds/my-role
```

Key	Value
`lease_id`	`database/creds/my-role/2f6a614c-4aa2-7b19-24b9-ad944a8d4de6`
`lease_duration`	`1h`
`lease_renewable`	`true`
`password`	`SsnoaA-8Tv4t34f41baD`
`username`	`v-vaultuse-my-role-x`

4. API

Полный список настраиваемых параметров можно посмотреть на странице API плагина базы данных PostgreSQL.

Более подробную информацию о HTTP API движка секретов баз данных можно найти на странице API движка секретов баз данных.

Параметры конфигурации

Серверы StarVault настраиваются с использованием конфигурационного файла. Этот файл может иметь формат HCL или JSON.

Активация проверки прав доступа к файлам через переменную окружения **VAULT_ENABLE_FILE_PERMISSIONS_CHECK** позволяет StarVault проверять, принадлежат ли директория конфигурации и файлы пользователю, который запускает StarVault. Также проверяется отсутствие прав на запись или выполнение для группы или других пользователей. StarVault позволяет операторам указывать пользователя и права доступа к директории плагинов и исполняемым файлам с помощью параметров `plugin_file_uid` и `plugin_file_permissions` в конфигурации, если оператору необходимо установить иные значения. По умолчанию эта проверка отключена.

Пример конфигурации показан ниже:

```
ui            = true
cluster_addr  = "https://127.0.0.1:8201"
api_addr      = "https://127.0.0.1:8200"
disable_mlock = true

storage "raft" {
  path = "/path/to/raft/data"
  node_id = "raft_node_id"
}

listener "tcp" {
  address       = "127.0.0.1:8200"
  tls_cert_file = "/path/to/full-chain.pem"
  tls_key_file  = "/path/to/private-key.pem"
}

telemetry {
  statsite_address = "127.0.0.1:8125"
  disable_hostname = true
}
```

После изменения файла конфигурации необходимо перезапустить сервис **starvault** для применения новых параметров:

BASH

```
systemctl restart starvault
```

1. Обзор параметров

storage

Обязательный блок параметров. Настройка бэкенда хранилища, где будут сохраняться данные StarVault. Для работы StarVault в режиме высокой доступности (HA) необходимо, чтобы бэкенд поддерживал семантику координации. Если бэкенд хранения поддерживает координацию в режиме HA, параметры бэкенда HA также могут быть указаны в этом блоке параметров. В противном случае, следует настроить отдельный параметр `ha_storage` с бэкендом, поддерживающим HA, вместе с соответствующими параметрами HA. Подробнее о параметрах бэкенда хранилища см. в разделе блок конфигурации storage.

ha_storage

Необязательный блок параметров. Настройка бэкенда хранилища, где будет происходить координация StarVault в режиме высокой доступности (HA). Это должен быть бэкенд, поддерживающий HA. Если параметр не установлен, попытка запустить HA будет выполнена на бэкенде, указанном в параметре `storage`. Этот параметр не требуется, если бэкенд хранилища поддерживает координацию HA и если специфические параметры HA уже указаны в блоке `storage`.

listener

Обязательный блок параметров. Настройка параметров прослушивания запросов API StarVault. Подробнее см. в разделе блок конфигурации listener.

user_lockout

Необязательный блок параметров. Настройка поведения блокировки пользователя при неудачном входе в систему. Подробнее см. в разделе Блокировка пользователей.

cluster_name

Необязательный строковый параметр. Указывает идентификатор для кластера StarVault. Если это значение не указано, StarVault сгенерирует его.

cache_size

Необязательный строковый параметр. Указывает размер кэша чтения, используемого физической подсистемой хранения. Значение указывается в количестве записей, поэтому общий размер кэша зависит от размера хранимых записей. По умолчанию **131072**.

disable_cache

Необязательный логический параметр. Отключает все кэши в StarVault, включая кэш чтения, используемый физической подсистемой хранения. Оказывает сильное влияние на производительность. По умолчанию **false**.

disable_mlock

Необязательный логический параметр. Отключает возможность сервера выполнять системный вызов **mlock**. **mlock** предотвращает выгрузку памяти на диск. Отключение **mlock** не рекомендуется, если не используется интегрированное хранилище. При отключении **mlock** следует соблюдать дополнительные меры безопасности, описанные ниже. Этот параметр также может быть задан через переменную окружения **VAULT_DISABLE_MLOCK**.

Отключение **mlock** не рекомендуется, если только системы, на которых работает StarVault, используют только зашифрованный swap или не используют swap вообще. StarVault поддерживает блокировку памяти только в UNIX-подобных системах, поддерживающих системный вызов **mlock()** (Linux, FreeBSD и т. д.). В таких системах как, например, Windows, NaCL, Android отсутствуют механизмы для предотвращения записи всего адресного пространства памяти процесса на диск, поэтому данная функция автоматически отключается на не поддерживаемых платформах.

Отключение **mlock** настоятельно рекомендуется при использовании интегрированного хранилища, поскольку **mlock** плохо совместим с файлами, отображаемыми в памяти, такими как те, что создаются BoltDB, используемым Raft для отслеживания состояния. При использовании **mlock** файлы, отображаемые в память, загружаются в резидентную память, что приводит к загрузке всего набора данных StarVault в оперативную память и может вызвать проблемы с нехваткой памяти, если объем данных StarVault превышает доступный объем ОЗУ. В этом случае, несмотря на то что данные внутри BoltDB остаются зашифрованными в режиме покоя, swap следует отключить, чтобы предотвратить выгрузку других конфиденциальных данных StarVault, находящихся в памяти, на диск.

В Linux, чтобы дать исполняемому файлу StarVault возможность использовать системный вызов **mlock** без запуска процесса от имени root, выполните команду:

```bash
sudo setcap cap_ipc_lock=+ep $(readlink -f $(which starvault))
```

Поскольку каждый плагин запускается как отдельный процесс, вам нужно сделать то же самое для каждого плагина в вашей директории plugins.

Если вы используете дистрибутив Linux с современной версией systemd, вы можете добавить следующую директиву в раздел конфигурации "[Service]":

```
LimitMEMLOCK=infinity
```

plugin_directory

Необязательный строковый параметр. Каталог, из которого разрешено загружать плагины. Для успешной загрузки плагинов StarVault должен иметь разрешение на чтение файлов в этой директории, а значение не может быть символической ссылкой. По умолчанию **""**.

plugin_tmpdir

Необязательный строковый параметр. Каталог, в котором StarVault может создавать временные файлы для поддержки взаимодействия Unix-сокета с контейнеризированными плагинами. Если значение не задано, StarVault будет использовать каталог по умолчанию для временных файлов. Обычно не требуется, если вы не используете контейнеризированные плагины и StarVault не разделяет временную папку с другими процессами, например, при использовании параметра **PrivateTmp** в **systemd**. Этот параметр также можно указать с помощью переменной окружения **VAULT_PLUGIN_TMPDIR**. По умолчанию **""**.

plugin_file_uid

Необязательный целочисленный параметр. Идентификатор пользователя (Uid) директорий плагинов и исполняемых файлов плагинов, если они принадлежат пользователю, отличному от того, кто запускает StarVault. Этот параметр необходимо устанавливать только в том случае, если проверка прав доступа к файлам включена через переменную окружения **VAULT_ENABLE_FILE_PERMISSIONS_CHECK**.

plugin_file_permissions

Необязательный строковый параметр. Строка восьмеричных прав доступа для директорий плагинов и исполняемых файлов плагинов, если установлены права на запись или выполнение для группы или других пользователей. Этот параметр необходимо устанавливать только в том случае, если проверка прав доступа к файлам включена через переменную окружения **VAULT_ENABLE_FILE_PERMISSIONS_CHECK**.

telemetry

Необязательный блок параметров. Указывает систему телеметрии для сбора и отправки статистических данных.

default_lease_ttl

Необязательный строковый параметр. Определяет срок действия аренды по умолчанию для токенов и секретов. Значение указывается с использованием суффикса времени, например "30s" (30 секунд) или "1h" (1 час). Это значение не может быть больше, чем `max_lease_ttl`. По умолчанию **768h**.

max_lease_ttl

Необязательный строковый параметр. Определяет максимально возможный срок действия аренды для токенов и секретов. Значение указывается с использованием суффикса времени, например "30s" (30 секунд) или "1h" (1 час). Отдельные точки монтирования могут изменить это значение, настроив точку монтирования с помощью флага `max-lease-ttl` в командах **auth** или **secret**. По умолчанию **768h**.

default_max_request_duration

Необязательный строковый параметр. Указывает максимальное стандартное время выполнения запроса, после которого StarVault отменяет запрос. Значение указывается с

использованием суффикса времени, например "30s" (30 секунд) или "1h" (1 час). Это значение может быть переопределено для каждого слушателя (listener) через параметр `max_request_duration`. По умолчанию **90s**.

detect_deadlocks

Необязательный строковый параметр. Строка значений, разделенных запятыми, которая указывает внутренние взаимоисключающие блокировки, за которыми следует наблюдать на предмет потенциальных взаимоблокировок. В настоящее время поддерживаемые значения включают `statelock`, `quotas` и `expiration`, что приведет к записи в лог "POTENTIAL DEADLOCK:", когда попытка блокировки состояния ядра кажется заблокированной. Включение этой функции может негативно сказаться на производительности из-за отслеживания каждой попытки блокировки. По умолчанию **""**.

raw_storage_endpoint

Необязательный логический параметр. Активирует конечную точку **sys/raw**, которая позволяет выполнять дешифрование/шифрование необработанных данных на входе и выходе из защитного барьера. Это конечная точка с высоким уровнем привилегий. По умолчанию **false**.

introspection_endpoint

Необязательный логический параметр. Активирует конечную точку **sys/internal/inspect**, которая позволяет пользователям с root-токеном или привилегиями sudo проводить инспекцию определенных подсистем внутри StarVault. По умолчанию **false**.

ui

Необязательный логический параметр. Активирует встроенный веб-интерфейс пользователя, который доступен на всех слушателях (адрес + порт) по пути **/ui**. Браузеры, обращающиеся к стандартному адресу API StarVault, будут автоматически перенаправлены туда. Этот параметр также может быть задан через переменную окружения **VAULT_UI**. По умолчанию **false**. Подробнее см. в разделе блок конфигурации ui.

pid_file

Необязательный строковый параметр. Путь к файлу, в котором должен храниться идентификатор процесса (PID) сервера StarVault.

enable_response_header_hostname

Необязательный логический параметр. Активирует добавление HTTP-заголовка во все HTTP-ответы StarVault: `X-Vault-Hostname`. Этот заголовок будет содержать имя узла StarVault, который обработал HTTP-запрос. Эта информация предоставляется по мере возможности и ее наличие не гарантируется. Если эта опция включена и заголовок `X-Vault-Hostname` отсутствует в ответе, это означает, что произошла какая-то ошибка при извлечении имени хоста из операционной системы. По умолчанию **false**.

enable_response_header_raft_node_id

Необязательный логический параметр. Активирует добавление HTTP-заголовка во все HTTP-ответы StarVault: `X-Vault-Raft-Node-ID`. Если StarVault участвует в кластере Raft (то есть использует интегрированное хранилище), этот заголовок будет содержать идентификатор узла Raft, который обработал HTTP-запрос. Если узел StarVault не участвует в кластере Raft, этот заголовок будет опущен, независимо от того, включена ли эта опция или нет. По умолчанию **false**.

log_level

Необязательный строковый параметр. Уровень подробности журнала. Поддерживаемые значения (в порядке убывания подробности): `trace`, `debug`, `info`, `warn` и `error`. Это значение также можно задать с помощью переменной окружения **VAULT_LOG_LEVEL**. По умолчанию **info**.

При SIGHUP (sudo kill -s HUP pid starvault), если указано правильное значение, StarVault обновит существующий уровень журнала, отменяя (даже если он указан) как флаг CLI, так и переменную окружения.

Не все части журнала StarVault могут динамически изменять уровень журнала таким образом; в частности, плагины secrets/auth в настоящее время не обновляются динамически.

log_format

Необязательный строковый параметр. Формат журнала. Поддерживаются следующие значения: `standard` и `json`. Его также можно указать через переменную окружения **VAULT_LOG_FORMAT**. По умолчанию **standard**.

log_file

Необязательный строковый параметр. Абсолютный путь, где StarVault должен сохранять сообщения журнала в дополнение к другим существующим выводам, таким как **journald/stdout**. Пути, которые заканчиваются разделителем пути, используют имя файла по умолчанию, **vault.log**. Пути, которые не заканчиваются расширением файла, используют расширение по умолчанию **.log**. Если файл журнала перезаписывается, StarVault добавляет текущую временную метку к имени файла в момент перезаписи. Например:

Значение параметра `log_file`	Текущий файл журнала	Файл журнала после ротации
/var/log/	**/var/log/starvault.log**	**/var/log/starvault-{timestamp}.log**
/var/log/my-diary	**/var/log/my-diary.log**	**/var/log/my-diary-{timestamp}.log**
/var/log/my-diary.txt	**/var/log/my-diary.txt**	**/var/log/my-diary-{timestamp}.txt**

log_rotate_duration

Необязательный строковый параметр. Указывает максимальную продолжительность записи в файл журнала, после которой он должен быть перезаписан. Должно быть указано значение продолжительности, например, `30s`. По умолчанию **24h**.

log_rotate_bytes

Необязательный целочисленный параметр. Указывает количество байт, которое может быть записано в файл журнала перед его перезаписью. Если не указано, то количество байт, которое может быть записано в файл журнала, не ограничено.

log_rotate_max_files

Необязательный целочисленный параметр. Указывает максимальное количество старых файлов журнала, которые следует сохранять. По умолчанию установлено значение **0** (файлы никогда не удаляются). Установите значение **-1**, чтобы удалять старые файлы журнала при создании нового.

experiments

Необязательный массив значений. Список экспериментальных функций, которые следует активировать для этого узла. Экспериментальные функции НЕ должны использоваться в производственной среде, и связанные с ними API могут претерпевать изменения, несовместимые с предыдущими версиями, между релизами. Дополнительные экспериментальные функции также могут быть указаны через переменную окружения **VAULT_EXPERIMENTS** в виде списка значений, разделённых запятыми.

imprecise_lease_role_tracking

Необязательный логический параметр. Позволяет пропустить подсчет аренды по ролям, если не включены квоты на основе ролей. Когда параметр `imprecise_lease_role_tracking` установлен в значение **true** и включена новая квота на основе ролей, последующий подсчет аренд начинается с **0**. Параметр `imprecise_lease_role_tracking` влияет на квоты подсчета аренды на основе ролей, но уменьшает задержки, если квоты на основе роли не используются.

2. Параметры высокой доступности

Следующие параметры используются для бэкендов, поддерживающих высокую доступность.

api_addr

Необязательный строковый параметр. Указывает адрес (полный URL), который будет анонсироваться другим серверам StarVault в кластере для перенаправления клиентов. Это значение также используется для бэкендов плагинов. Этот параметр также может быть задан через переменную окружения **VAULT_API_ADDR**. В общем случае, это должен быть полный URL, который указывает на значение адреса слушателя. Этот адрес

может быть динамически определен с помощью шаблона go-sockaddr, который разрешается во время выполнения.

cluster_addr

Необязательный строковый параметр. Указывает адрес для анонсирования другим серверам StarVault в кластере для перенаправления запросов. Этот параметр также может быть задан через переменную окружения VAULT_CLUSTER_ADDR. Это полный URL, подобно api_addr, но StarVault будет игнорировать схему (все участники кластера всегда используют TLS с приватным ключом/сертификатом). Этот адрес может быть динамически определен с помощью шаблона go-sockaddr, который разрешается во время выполнения.

disable_clustering

Необязательный логический параметр. Указывает, включены ли функции кластеризации, такие как переадресация запросов. Если установить значение **true** для одного узла хранилища, эти функции будут отключены только в том случае, если этот узел является активным узлом. Этот параметр нельзя установить в **true**, если типом хранилища является raft. По умолчанию **false**.

3. Содержание раздела