Методы аутентификации. Логин и пароль (Advanced)

Метод аутентификации userpass-advanced позволяет пользователям аутентифицироваться в StarVault, используя комбинацию имени пользователя и пароля. В отличие от базовой версии, этот метод поддерживает **смену собственного пароля** пользователем, установку флага принудительной смены пароля администратором, генерацию пароля по политике. С настройками политик паролей подробнее можно ознакомиться в руководстве.

Комбинации имени пользователя, пароля и флаг принудительной смены пароля перед входом пользователя настраиваются непосредственно в методе аутентификации через путь **users/**. Этот метод не может считывать имена пользователей и пароли из внешнего источника.

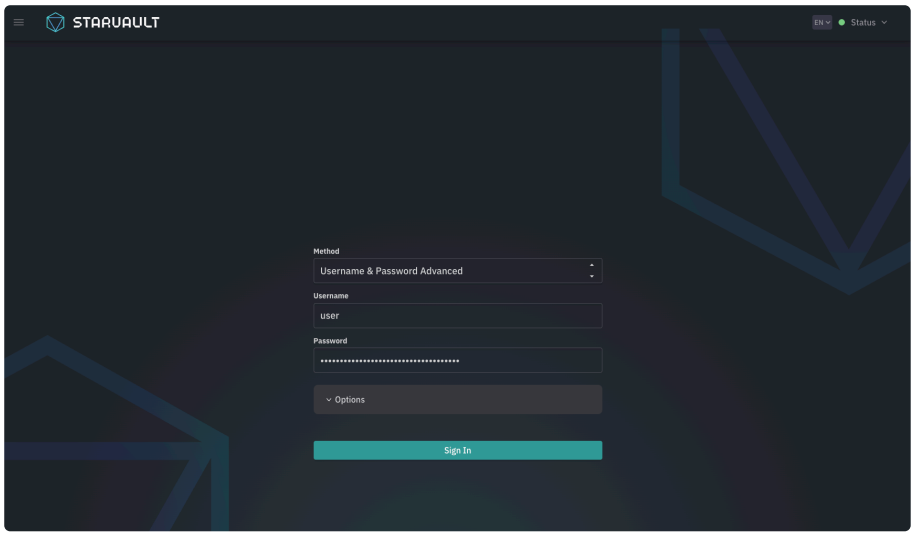
Метод приводит все введенные имена пользователей к нижнему регистру, например, `Mary` и `mary` будут считаться одной и той же записью.

1. Аутентификация с помощью метода Userpass-advanced

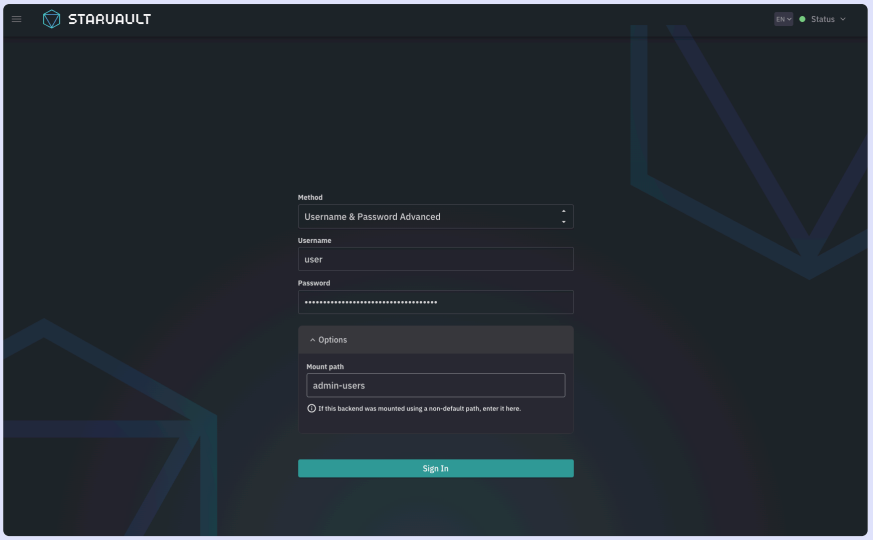
Этот метод аутентификации можно использовать для аутентификации через UI, CLI или API.

UI

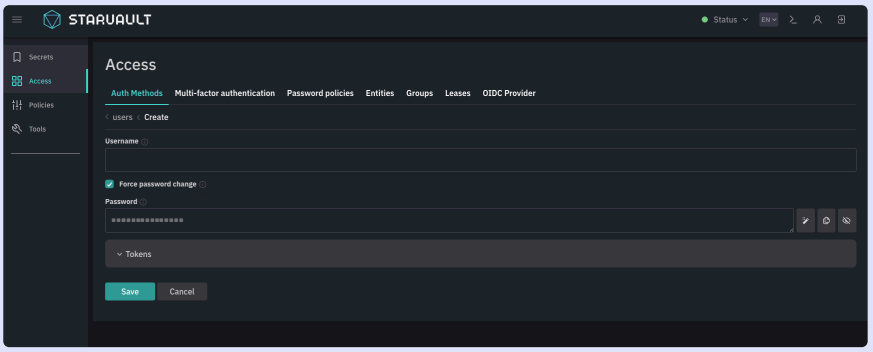
На странице входа выберите **Username & Password Advanced** в качестве метода и в поля **Username** и **Password** введите соответствующие значения.



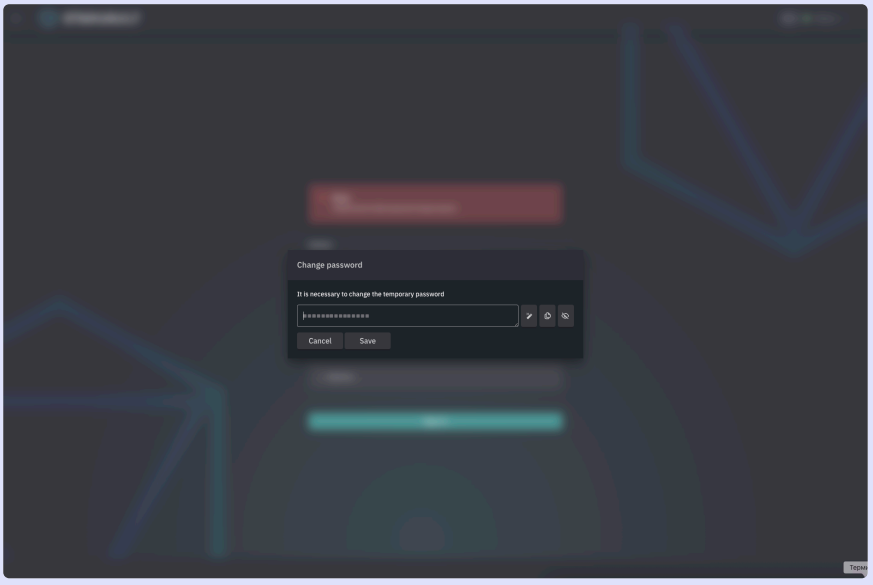
Если метод **Userpass & Password Advanced** был смонтирован по пути, отличном от пути по умолчанию (**auth/userpass**) введите этот путь в поле **Mount path** в разделе **Hide options**:



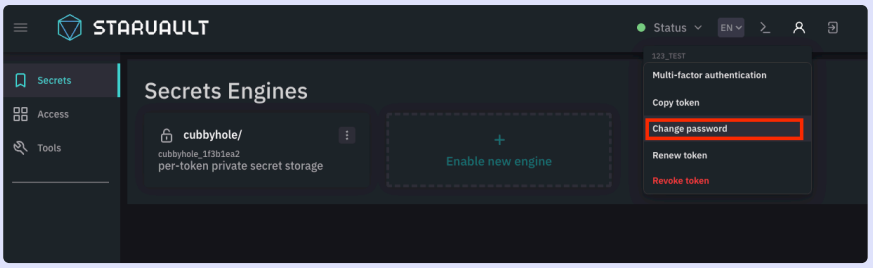
Пусть у созданного пользователя в разделе **users/** был выставлен флаг **Force password change** (предвыбирается автоматически), как показано ниже:



В данном случае при первом входе необходимо выполнить смену пароля:



Ручная смена пароля доступна при нажатии на иконку профиля:



CLI

- Создание пользователя:

```
starvault write auth/<userpass-advanced>/users/<username> \
password="<password>" \
force_password_change=true
```

- Просмотр информации о пользователе:

```
starvault read auth/<userpass-advanced>/users/<username>
```

- Вход в УЗ, если `force_password_change = false`, то выполнить действие Вход после смены пароля.
- Попытка входа, если флаг `force_password_change=true` (по умолчанию). Возвращает ошибку `403 password change required`:

```
starvault write auth/<userpass-advanced>/login/<username> password="
<password>"
```

- Смена временного пароля:

```
starvault write auth/<userpass-advanced>/users/change-temporary-password
\
username="<username>" \
password="<currnet_password>" \
new_password="<new_password>"
```

- Вход после смены пароля:

```
starvault write auth/<userpass-advanced>/login/<username> \
password="<password>"
```

- **Необязательно**. Смена пароля вручную:

```
starvault write auth/<userpass-advanced>/users/password \
old_password="<current_password>" \
```

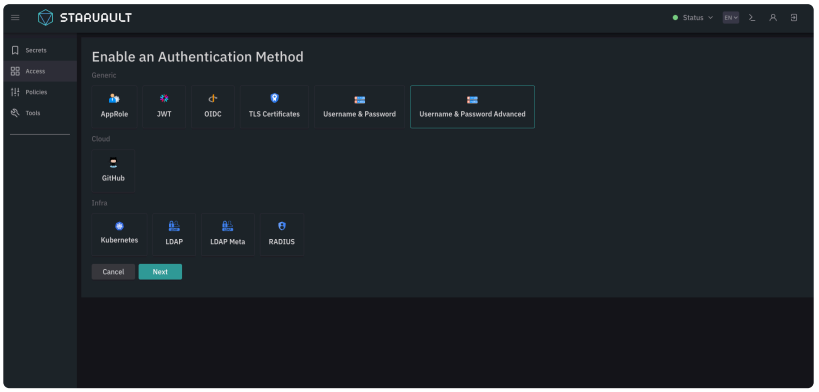
```
new_password="<new_password>"
```

2. Настройка Userpass-advanced

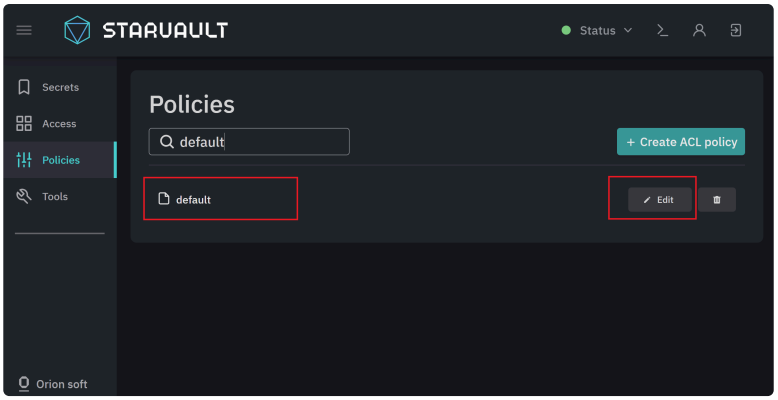
Методы аутентификации должны быть настроены заранее, прежде чем пользователи или машины смогут пройти аутентификацию. Эти шаги обычно выполняются администратором или инструментом управления конфигурацией.

Для настройки Userpass-advanced выполните следующие действия:

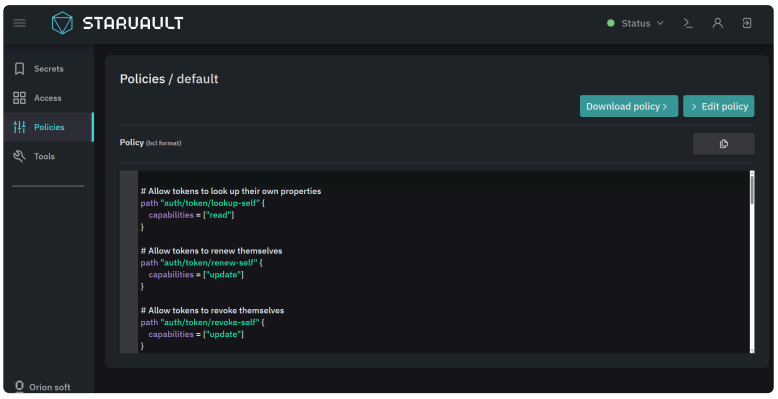
1. Активируйте метод:
    - **Через UI**:
        a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.
        b. На странице **Access** нажмите **[Enable new method]**.
        c. Выберите **Username & Password Advanced** и нажмите **[Next]**.



        d. При необходимости измените путь в поле **Path** и параметры метода в группе **Hide Method Options**. Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации
        e. Нажмите **[Enable Method]**.
        f. Для того, чтобы пользователь мог менять пароль, необходимо отредактировать политику **default**. Для этого выполните шаги:
            i. Перейдите на вкладку **Policies** и с помощью фильтра найдите политику **default**.



ii. Нажмите на кнопку **[Edit policy]** и перейдите к редактированию политики.

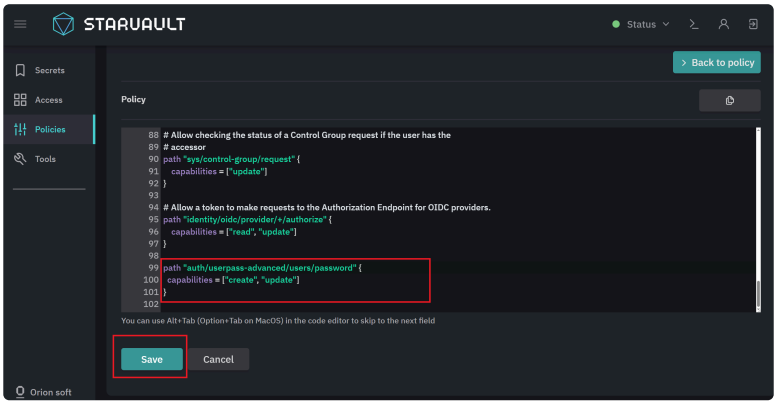


iii. Добавьте в политику следующие строки:

```
path "auth/<userpass-advanced>/users/password" {
  capabilities = ["create", "update"]
}
```

> Впишите в поле path свой путь метода.

iv. Нажмите **[Save]**



- **Через CLI**:

a. Используйте команду для активации метода аутентификации.

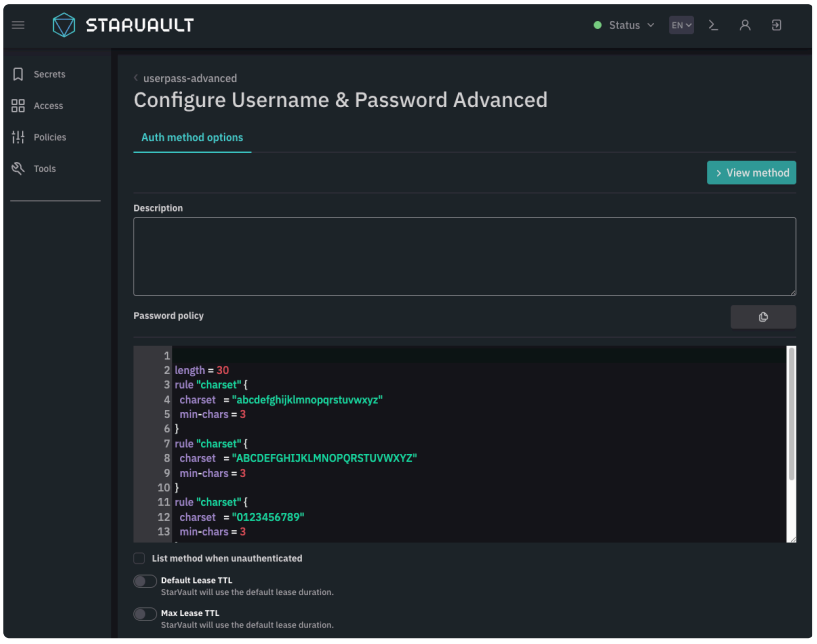
```bash
starvault auth enable userpass-advanced
```

b. Используйте команду для активации метода аутентификации с переопределением path.

```
starvault auth enable -path=<path> userpass-advanced
```

Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации

2. **Необязательно**. Доступно редактирование стандартной парольной политики Password policy в конфгурации метода `auth/configure/userpass-advanced/options`. С настройками политик паролей подробнее можно ознакомиться в руководстве:
   - **Через UI**:
     a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.
     b. На странице **Access** нажмите на карточку метода **Userpass-advanced**.
     c. На странице метода нажмите **[Configuration]**.
     d. В разделе конфигурации метода нажмите **[Configure]**.



   - **Через CLI**:
     a. Используйте команду для редактирования парольной политики.

```
starvault write auth/userpass-advanced/policies/password
policy="length = 30
rule \"charset\" {
  charset = \"abcdefghijklmnopqrstuvwxyz\"
  min-chars = 3
}
rule \"charset\" {
  charset = \"ABCDEFGHIJKLMNOPQRSTUVWXYZ\"
  min-chars = 3
}
```

```
rule \"charset\" {
  charset = \"0123456789\"
  min-chars = 3
}
rule \"charset\" {
  charset = \"-\"
  min-chars = 3
}"
```

3. Создайте пользователя и ассоциируйте его с политикой (подробнее см. в разделе Политики доступа)

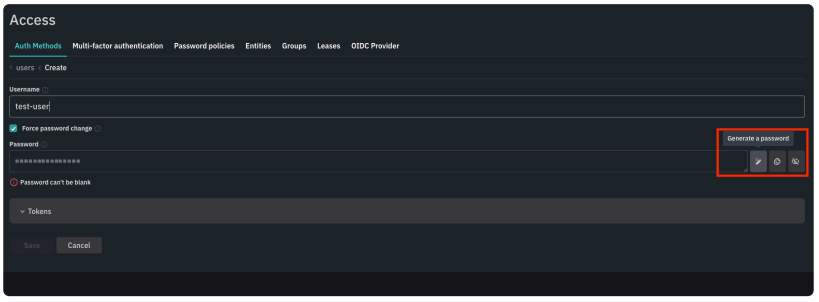
   - **Через UI**:

     a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.

     b. На странице **Access** нажмите на карточку метода **Userpass-advanced**.

     c. На странице метода нажмите **Create user**.

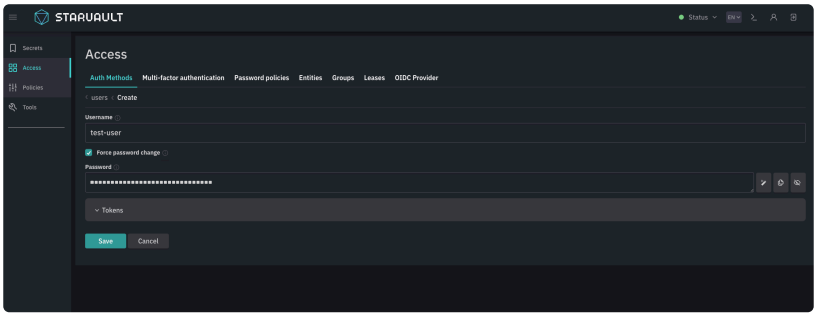
     d. Введите имя пользователя и пароль (его можно сгенерировать).



     e. **Необязательно**. Переопределите параметр **Force password change**, отвечающий за необходимость смены пароля при первом входе пользователя.

     f. В разделе **Hide Tokens** можно настроить дополнительные параметры, в том числе связать токен с политиками.

     g. Нажмите **Save**



     Подробнее о доступных параметрах токена см. в таблице.

   - **Через CLI**:

   Используйте следующую команду для создания пользователя:

```bash
starvault write auth/<userpass:path>/users/<username> \
    password=<password> \
    force_password_change=true \
    [options]
```

3. Управление пользователями

В таблице ниже представлены команды для управления пользователями в методе **Userpass-advanced**.

Команда	Описание
`starvault write auth/<userpass:path>/users/<username> <key=value list>`	Создание нового пользователя или изменение существующего по пути `auth/<userpass:path>/users/<username>`. Параметры учетной записи передаются в формате `key=value`. Для добавления нескольких ключей со значениями, их необходимо перечислить через пробел. Подробнее о доступных параметрах токена см. в таблице.
`starvault read [-format=<string>] auth/<userpass:path>/users/<username>`	Возвращает данные пользователя с именем <username> в методе **Userpass-advanced** по пути `<userpass:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json', 'raw'.
`starvault list [-format=<string>] auth/<userpass:path>/users`	Возвращает список пользователей в методе **Userpass-advanced** по пути `<userpass:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json'.
`starvault delete auth/<userpass:path>/users/<username>`	Удаляет пользователя с именем <username> в методе **Userpass-advanced** по пути `<userpass:path>`.

Параметры учетной записи, которые можно передать при создании/изменении пользователя представлены в следующей таблице:

Опция CLI	Поле в UI	Описание
`force_password_change`	Флаг **Force password change**	Логическое значение. Определяет, необходима ли смена пароля при первом входе. Если **force_password_change** установлен в `true`, то для первого входа нужно будет произвести смену пароля.

Опция CLI	Поле в UI	Описание
`token_ttl`	Переключатель **Generated Token's Initial TTL**	Целочисленное или строковое значение. Определяет время жизни (TTL) токенов, выдаваемых для данного пользователя. По истечении этого времени токен будет автоматически отозван, если только он не будет продлен.
`token_max_ttl`	Переключатель **Generated Token's Maximum TTL**	Целочисленное или строковое значение. Максимальное время жизни сгенерированных токенов. Текущее значение этого параметра будет указано при обновлении.
`token_policies`	Поля **Generated Token's Policies**	Список значений, разделенных запятыми. Используется для указания списка политик, которые будут присвоены токенам, выданным при аутентификации с использованием этого пользователя.
`token_bound_cidrs`	Поля **Generated Token's Bound CIDRs**	Список значений, разделенных запятыми. Определяет список CIDR-адресов, для которых будет действителен выданный токен. Это ограничение по IP-адресам устанавливает, что токен может использоваться только с определенных IP-адресов или диапазонов IP-адресов, что повышает безопасность, ограничивая возможность использования токена
`token_explicit_max_ttl`	Переключатель **Generated Token's Explicit Maximum TTL**	Целочисленное или строковое значение. Задает максимальное время жизни (TTL) для токенов, выданных с использованием этого пользователя. Это значение устанавливает жесткий верхний предел времени жизни токена, который не может быть превышен даже при продлении токена.
`token_no_default_policy`	Опция **Do Not Attach 'default' Policy To Generated Tokens**	Логическое значение. Определяет, будет ли новому токену автоматически присваиваться политика по умолчанию. Если `token_no_default_policy` установлен в `true`, то при создании токена политика по умолчанию не будет присвоена.
`token_num_uses`	Поле **Maximum Uses of Generated Tokens**	Целочисленное значение. Максимальное количество раз, которое может быть использован сгенерированный токен (в течение срока его действия). 0 означает неограниченное количество раз. Если требуется, чтобы токен имел возможность создавать дочерние токены, необходимо установить это значение равным 0.

Опция CLI	Поле в UI	Описание
`token_period`	Переключатель **Generated Token's Period**	Целочисленное или строковое значение. Определяет период, в течение которого выданный токен может быть продлен бесконечное количество раз, пока политики, связанные с токеном, остаются неизменными.
`token_type`	Поле **Generated Token's Type**	Строковое значение. Определяет тип токена, который будет выдан при аутентификации с использованием этой роли. Допустимые значения: • `service` - стандартный тип токена, который может быть продлен в соответствии с его политиками и TTL. Токены типа "service" подходят для большинства приложений и сервисов, которые требуют долгосрочного доступа и возможности продления токена. • `batch` - легковесный тип токена, который хранится в памяти и не записывается в хранилище данных StarVault. Токены типа "batch" не могут быть продлены или отозваны индивидуально, и они исчезают при перезагрузке или выключении сервера StarVault. Они подходят для краткосрочных операций и сценариев с большим объемом токенов, где требуется меньшая нагрузка на хранилище. • `default` - Если параметр не задан, будет использоваться тип токена по умолчанию, который в большинстве случаев является типом "service".

Пример:

BASH

```
starvault write auth/admin-users/users/admin \ ①
  password=P@ssw0rd \ ②
  token_policies=admins-kv \ ③
  token_ttl=5h \ ④
  token_num_uses=2 ⑤
```

① Создается пользователь с именем *admin* в методе аутентификации по пути **admin-users**

② Установка пароля пользователя

③ Сопоставление токена пользователя с политикой admins-kv

④ Срок действия токена пользователя задан равным 5 часам

⑤ С этим токеном можно аутентифицироваться 2 раза. Затем токен будет аннулирован.

2025 orionsoft. Все права защищены.

Методы аутентификации. Логин и пароль

Метод аутентификации userpass позволяет пользователям аутентифицироваться в StarVault, используя комбинацию имени пользователя и пароля.

Комбинации имени пользователя и пароля настраиваются непосредственно в методе аутентификации через путь **users/**. Этот метод не может считывать имена пользователей и пароли из внешнего источника.

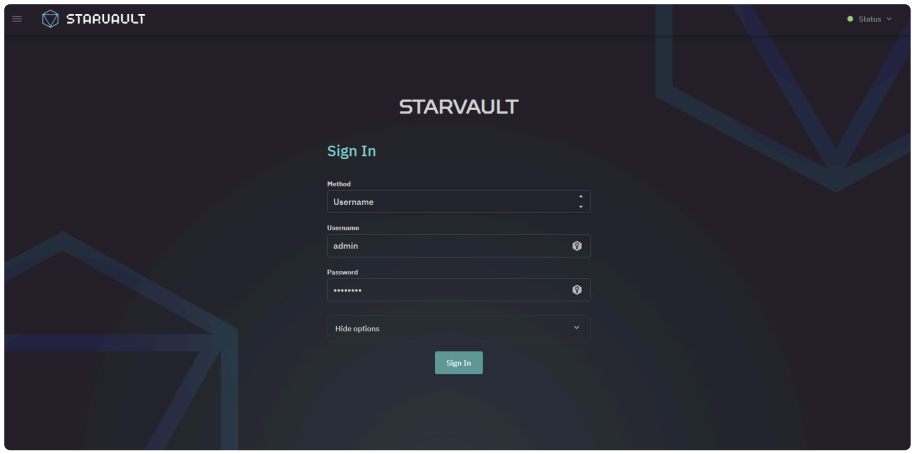
Метод приводит все введенные имена пользователей к нижнему регистру, например, `Mary` и `mary` будут считаться одной и той же записью.

1. Аутентификация с помощью метода Userpass

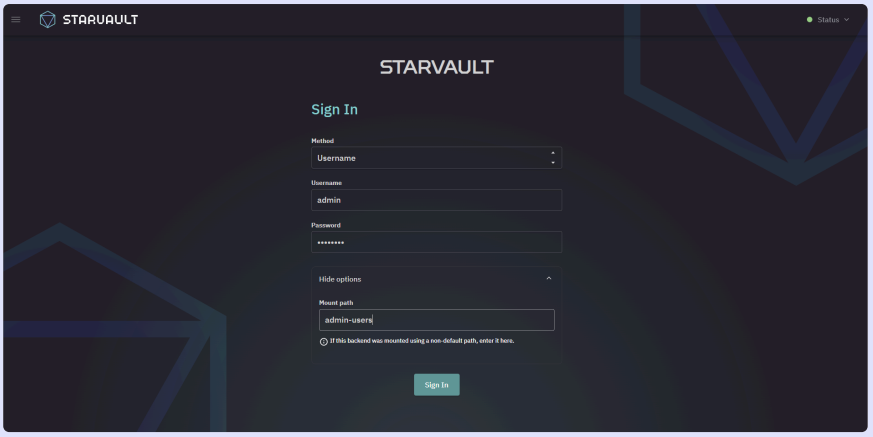
Этот метод аутентификации можно использовать для аутентификации через UI, CLI или API.

UI

На странице входа выберите **Username** в качестве метода и в поля **Username** и **Password** введите соответствующие значения.



Если метод **Userpass** был смонтирован по пути, отличном от пути по умолчанию (**auth/userpass**) введите этот путь в поле **Mount path** в разделе **Hide options**:



CLI

Используйте команду `starvault login`:

```
starvault login -method=userpass \ ①
  username=admin \ ②
  password=P@ssw0rd \ ③
  [-path=<path>] ④
```

① Указывает команде использовать метод **Userpass**

② Указывает пользователя для входа

③ Указывает пароль для входа

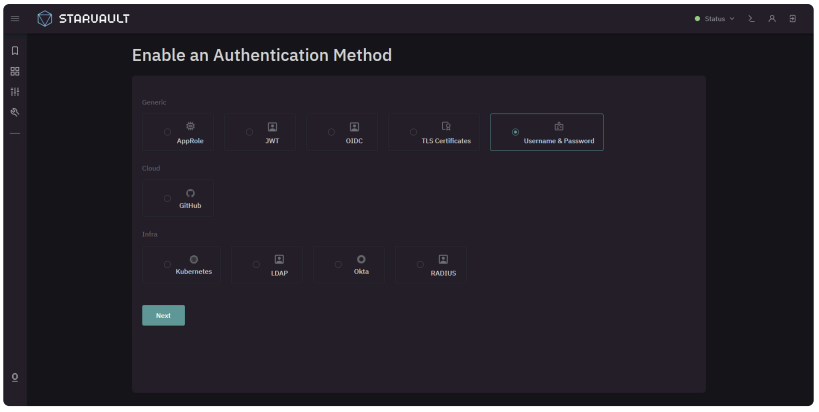
④ Если метод **Userpass** был смонтирован по пути, отличном от пути по умолчанию (**auth/userpass**) необходимо ввести актуальный путь.

2. Настройка Userpass

Методы аутентификации должны быть настроены заранее, прежде чем пользователи или машины смогут пройти аутентификацию. Эти шаги обычно выполняются администратором или инструментом управления конфигурацией.

Для настройки Userpass выполните следующие действия:

1. Активируйте метод:
   - Через UI:
     a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.
     b. На странице **Access** нажмите **[Enable new method]**.
     c. Выберите **Username & Password** и нажмите **[Next]**.



d. При необходимости измените путь в поле **Path** и параметры метода в группе **Hide Method Options**. Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации.

e. Нажмите **[Enable Method]**.

- Через CLI:

a. Используйте команду для активации метода аутентификации.

```
starvault auth enable [options] userpass ①
```

① Подробнее о параметрах см. в разделе Общие параметры для методов аутентификации

2. Создайте пользователя и ассоциируйте его с политикой (подробнее см. в разделе Политики):

- Через UI:

a. Аутентифицируйтесь в пользовательском интерфейсе с достаточными правами.

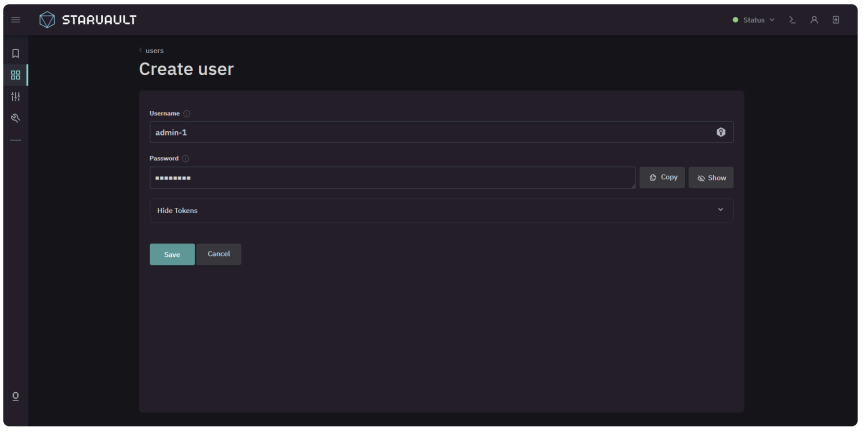
b. На странице **Access** нажмите на карточку метода **Userpass**.

c. На странице метода нажмите **[Create user]**.

d. Введите имя пользователя и пароль.

e. В разделе **Hide Tokens** можно настроить дополнительные параметры, в т.ч связать токен с политиками.

f. Нажмите **[Save]**.



Подробнее о доступных параметрах токена см. в таблице

- Через CLI:

  a. Используйте команду для активации метода аутентификации.

```
starvault write auth/<userpass:path>/users/<username> \ ① ②
    password=<password> \ ③
    [options] ④
```

① <userpass:path> - точка монтирования метода. По умолчанию **userpass**

② <username> - имя пользователя. Обязательный параметр

③ <password> - пароль для пользователя. Обязательный параметр

④ Дополнительные параметры токена пользователя. Подробнее о доступных параметрах токена см. в таблице

3. Управление пользователями

В таблице ниже представлены команды для управления пользователями в методе **Userpass**.

Команда	**Описание**
`starvault write auth/<userpass:path>/users/<username> <key=value list>`	Создание нового пользователя или изменение существующего по пути `auth/<userpass:path>/users/<username>`. Параметры учетной записи передаются в формате `key=value`. Для добавления нескольких ключей со значениями, их необходимо перечислить через пробел. Подробнее о доступных параметрах токена см. в таблице
`starvault read [-format=<string>] auth/<userpass:path>/users/<username>`	Возвращает данные пользователя с именем <username> в методе **Userpass** по пути `<userpass:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json', 'raw'.
`starvault list [-format=<string>] auth/<userpass:path>/users`	Возвращает список пользователей в методе **Userpass** по пути `<userpass:path>`. С помощью опции `-format=<string>` можно указать формат представления данных. Допустимые значения: `table` (по умолчанию), `yaml`, 'json'.
`starvault delete auth/<userpass:path>/users/<username>`	Удаляет пользователя с именем <username> в методе **Userpass** по пути `<userpass:path>`.

Параметры учетной записи, которые можно передать при создании/изменении пользователя представлены в следующей таблице:

Опция CLI	**Поле в UI**	**Описание**
`token_ttl`	Переключатель **Generated Token's Initial TTL**	Целочисленное или строковое значение. Определяет время жизни (TTL) токенов, выдаваемых для данного пользователя. По истечении этого времени токен будет автоматически отозван, если только он не будет продлен.
`token_max_ttl`	Переключатель **Generated Token's Maximum TTL**	Целочисленное или строковое значение. Максимальное время жизни сгенерированных токенов. Текущее значение этого параметра будет указано при обновлении.
`token_policies`	Поля **Generated Token's Policies**	Список значений, разделенных запятыми. Используется для указания списка политик, которые будут присвоены токенам, выданным при аутентификации с использованием этого пользователя.
`token_bound_cidrs`	Поля **Generated Token's Bound CIDRs**	Список значений, разделенных запятыми. Определяет список CIDR-адресов, для которых будет действителен выданный токен. Это ограничение по IP-адресам устанавливает, что токен может использоваться только с определенных IP-адресов или диапазонов IP-адресов, что повышает безопасность, ограничивая возможность использования токена
`token_explicit_max_ttl`	Переключатель **Generated Token's Explicit Maximum TTL**	Целочисленное или строковое значение. Задает максимальное время жизни (TTL) для токенов, выданных с использованием этого пользователя. Это значение устанавливает жесткий верхний предел времени жизни токена, который не может быть превышен даже при продлении токена.
`token_no_default_policy`	Опция **Do Not Attach 'default' Policy To Generated Tokens**	Логическое значение. Определяет, будет ли новому токену автоматически присваиваться политика по умолчанию. Если `token_no_default_policy` установлен в `true`, то при создании токена политика по умолчанию не будет присвоена.

Опция CLI	Поле в UI	Описание
`token_num_uses`	Поле **Maximum Uses of Generated Tokens**	Целочисленное значение. Максимальное количество раз, которое может быть использован сгенерированный токен (в течение срока его действия). 0 означает неограниченное количество раз. Если требуется, чтобы токен имел возможность создавать дочерние токены, необходимо установить это значение равным 0.
`token_period`	Переключатель **Generated Token's Period**	Целочисленное или строковое значение. Определяет период, в течение которого выданный токен может быть продлен бесконечное количество раз, пока политики, связанные с токеном, остаются неизменными.
`token_type`	Поле **Generated Token's Type**	Строковое значение. Определяет тип токена, который будет выдан при аутентификации с использованием этой роли. Допустимые значения: • `service` - стандартный тип токена, который может быть продлен в соответствии с его политиками и TTL. Токены типа "service" подходят для большинства приложений и сервисов, которые требуют долгосрочного доступа и возможности продления токена. • `batch` - легковесный тип токена, который хранится в памяти и не записывается в хранилище данных StarVault. Токены типа "batch" не могут быть продлены или отозваны индивидуально, и они исчезают при перезагрузке или выключении сервера StarVault. Они подходят для краткосрочных операций и сценариев с большим объемом токенов, где требуется меньшая нагрузка на хранилище. • `default` - Если параметр не задан, будет использоваться тип токена по умолчанию, который в большинстве случаев является типом "service".

Пример:

```
starvault write auth/admin-users/users/admin \ ①
  password=P@ssw0rd \ ②
  token_policies=admins-kv \ ③
```

```
  token_ttl=5h \ ④
  token_num_uses=2 ⑤
```

① Создается пользователь с именем *admin* в методе аутентификации по пути **admin-users**

② Установка пароля пользователя

③ Сопоставление токена пользователя с политикой admins-kv

④ Срок действия токена пользователя задан равным 5 часам

⑤ С этим токеном можно аутентифицироваться 2 раза. Затем токен будет аннулирован.

2025 orionsoft. Все права защищены.

Рекомендации по автоматизации

Если ресурсы созданы непосредственно из API или командной строки StarVault, то принудительно применять к ним соответствующие политики управления может быть непросто. Для управления крупными развертываниями StarVault рекомендуется систематизировать управление ресурсами с помощью Terraform и Terraform-провайдера StarVault.

Terraform применяет политику и управляет, используя подход "инфраструктура как код" (IaC), что позволяет программно управлять такими ресурсами StarVault как методы аутентификации, плагины, пространства имен и политики. Если, например, определенные политики ACL или Sentinel должны применяться к каждому пространству имен StarVault, то Terraform обеспечит корректное применение при создании каждого нового пространства имен.

1. Рекомендации по Terraform

- **Работая через Terraform, избегайте чтения/записи долгосрочных статических секретов в StarVault.** Данные, прочитанные в Terraform или записанные из него, сохраняются в файле состояния Terraform и всех сгенерированных файлах плана.
- **Шифруйте файл состояния Terraform.** Защищайте файл состояния с помощью надежного шифрованного бэкенда.
- **Соблюдайте принцип минимальных привилегий.** Ограничьте доступ на чтение/запись к файлу состояния Terraform.
- **Ограничьте возможности прямого управления ресурсами StarVault.** Используйте политики Sentinel, чтобы ограничить разрешения на управление теми ресурсами, которыми нужно управлять через Terraform.
- **Используйте краткосрочные учетные данные.** Учетные данные сохраняются в файле состояния Terraform. Использование краткосрочных учетных данных снижает риск утечки в случае компрометации файла состояния.

2. Рекомендации по StarVault

- **Используйте динамические учетные данные, сохраняемые в StarVault, для разных провайдеров облачных услуг.** При использовании динамических учетных данных, сохраняемых в StarVault, не нужно создавать уникальные динамические учетные данные под каждого провайдера облачных услуг, что позволяет централизованно управлять

конфиденциальными данными с помощью StarVault и создавать лишь временные учетные данные под различных облачных провайдеров.

- **Используйте атрибут пространства имен в ресурсах и источниках данных.** Использование имеющегося у ресурса атрибута "пространство имен" вместо псевдонима провайдера упрощает настройку и избавляет от необходимости иметь несколько блоков для разных провайдеров.
- **Точечно используйте функции токенов.** Используйте минимально необходимые возможности токенов StarVault для управления ресурсами StarVault. Например, для чтения данных из источника данных KV и обнаружения дрейфа нужна только функция `read`, в то время как управление ресурсом KV требует функции `create` или `update` в зависимости от того, существует ли уже ресурс или нет, а удаление ресурса KV из конфигурации Terraform требует функции `delete`.
- Перенесите существующие ресурсы в Terraform. Если ресурсы StarVault созданы вне процесса выделения ресурсов Terraform, перенесите неуправляемые ресурсы.
- **При работе со StarVault всегда, когда возможно, используйте провайдер динамических учетных данных.** Провайдер динамических учетных данных по мере необходимости генерирует краткосрочные учетные данные, поэтому статические учетные данные уже не так нужны, что улучшает безопасность интеграции.
- **Не передавайте `address` или `token` в блок конфигурации провайдера.** Если используете провайдер динамических учетных данных, Terraform заполняет переменную окружения `VAULT_ADDR` значением `address`. Далее Terraform использует эти переменные окружения, чтобы извлечь значение для `token`.
- **Не задавайте учетные данные StarVault в коде напрямую.** Если нет возможности использовать провайдер динамических учетных данных, настройте провайдер StarVault с помощью переменных окружения.

Динамические учетные данные влияют на поведение дочерних токенов

Если используете динамические учетные данные вместе с Terraform-провайдером StarVault, это означает, что Terraform управляет жизненным циклом токена StarVault. В результате StarVault не создает дочерний токен и нет возможности использовать аргументы провайдера для управления дочерним токеном StarVault (например, аргумент `token_name`).

2025 orionsoft. Все права защищены.