VK WorkSpace

Корпоративный мессенджер VK Teams

Инструкция по настройке Single Sign-On аутентификации

© VK WorkSpace, 2024 г. Все права защищены

Оглавление

Назначение документа

В данной инструкции представлено описание процесса настройки Single Sign-On аутентификации по протоколам SAML и OIDC, а также SSO-аутентификации по протоколу Kerberos в Microsoft Active Directory.

Документ предназначен для использования администраторами организации.

Дополнительная документация

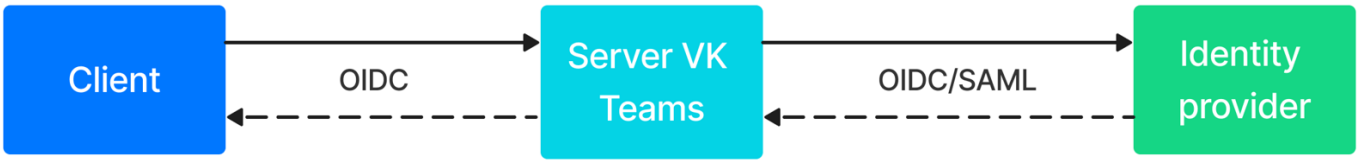
Архитектура и описание системы — в документе представлена информация о сервисах VK Teams, обеспечивающих функциональность SSO-аутентификации, а также расположение log-файлов данных сервисов. Не является частью публичной документации, обратитесь к представителю VK Tech, чтобы ознакомиться с документом.

Предварительные условия для настройки SSO аутентификации

Клиентские платформы в рамках запроса аутентификации должны поддерживать аутентификацию через внешнего провайдера аутентификации, в результате которой сервер отдаст ответ, содержащий email и atoken (ключ, необходимый для инициализации сессии и получения идентификатора сессии — aimsid), используемые далее при старте сессии мессенджера.

Необходимо отключить блокировку всплывающих окон в браузере, так как поддержка OIDC реализована через всплывающие окна.

Функциональное описание



Внешний провайдер аутентификации (Identity Provider) — провайдер осуществляющий аутентификацию и поддерживающий протоколы аутентификации SAML и OpenID Connect. Провайдером является SAML IDP или OIDC Authentication server.

В процессе аутентификации пользователя сервер VK Teams перенаправляет пользователя на провайдера аутентификации. Клиент переходит по указанным redirect'ам, пользователь вводит аутентификационные данные. Далее клиент начинает новую сессию мессенджера и пользуется идентификатором сессии (aimsid) во всех запросах. При отзыве access_token'а aimsid инвалидируется.

На клиентских приложениях aimsid хранится во внутреннем хранилище ОС в зашифрованном виде (за исключением web-версии), в соответствии с таблицей:

Платформа	**Технология для хранения aimsid**
Web	Cookie
MacOS	Симметрично зашифрован в локальном файле конфигурации
Windows	Симметрично зашифрован в локальном файле конфигурации
Linux	Симметрично зашифрован в локальном файле конфигурации

Платформа	Технология для хранения aimsid
iOS	Keychain
Android	Encrypted Shared Preferences

По окончании процесса аутентификации все управление токенами и взаимодействие с провайдером аутентификации осуществляется на стороне сервера VK Teams.

Хранением, обновлением, проверкой токенов занимается сервис Tokeeper. Сервис хранит данные, используя БД Tarantool. БД Tarantool отслеживает токены, период жизни которых истек, и автоматически удаляет их из базы.

Настройка взаимодействия Client — Server VK Teams представлена в разделе Шаг 1. Настройка подсистемы авторизации сервера VK Teams.

Настройка взаимодействия Server VK Teams — Identity Provider представлена в Шагах 2-5.

Ограничения Keycloak

Сервис Keycloak не взаимодействует с внешним провайдером аутентификации после авторизации. Соответственно, не сможет инвалидировать свою сессию при инвалидации сессии пользователя на внешнем провайдере аутентификации.

Однако, если инвалидировать сессию пользователя в сервисе Keycloak, клиент VK Teams разлогинит пользователя.

Одновременная работа с несколькими провайдерами аутентификации

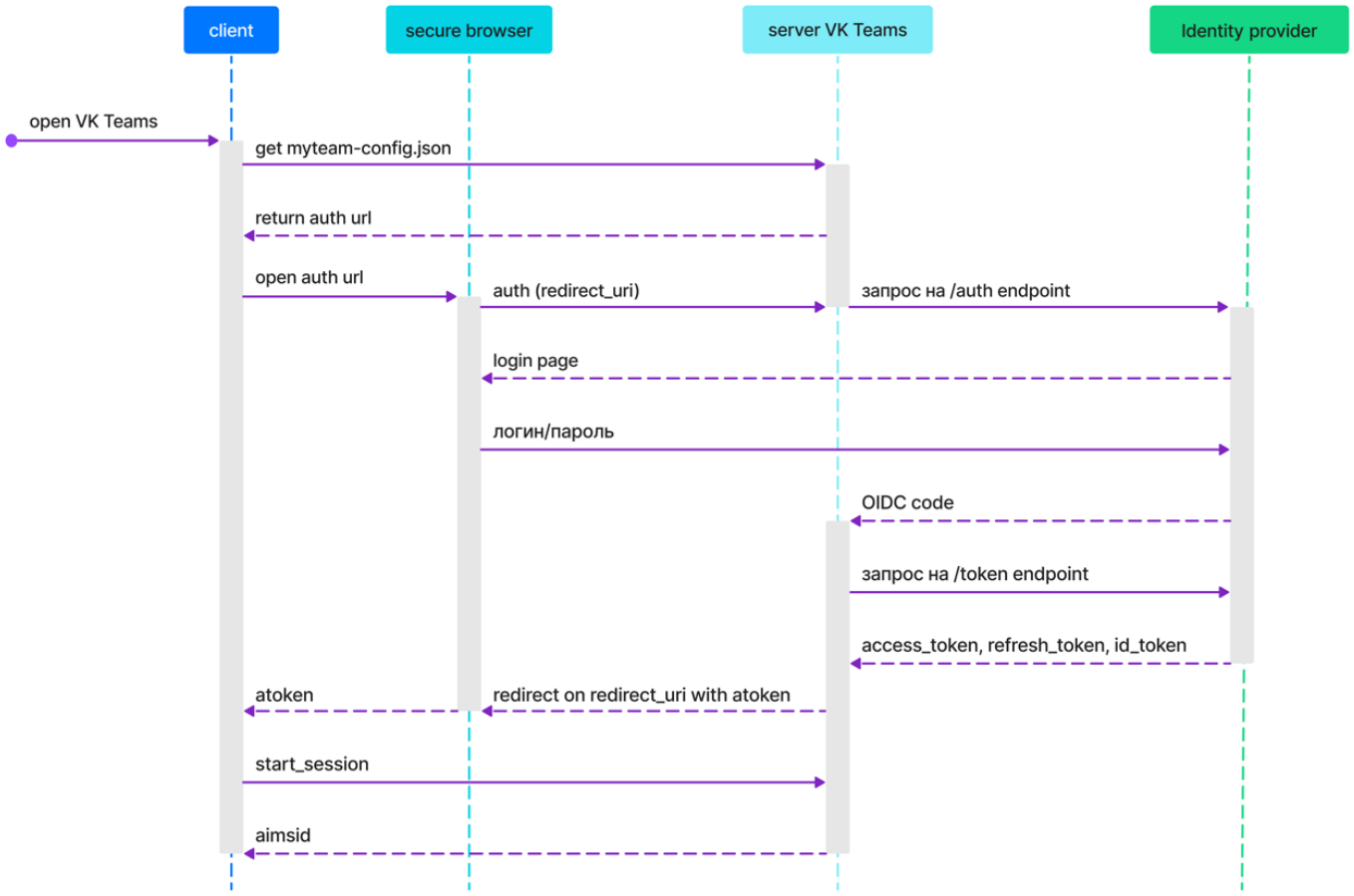
SSO аутентификация поддерживает аутентификацию через несколько провайдеров.

Выбор нужного провайдера осуществляется на основе useragent'a, переданного параметром в начале процесса аутентификации. Процесс настройки провайдеров представлен ниже.

Secure Browser

- Для iOS — ASWebAuthenticationSession.
- Для Android — Android Custom Tab.
- Для Web — Window.open.
- Для Desktop — открытие в стандартном браузере.

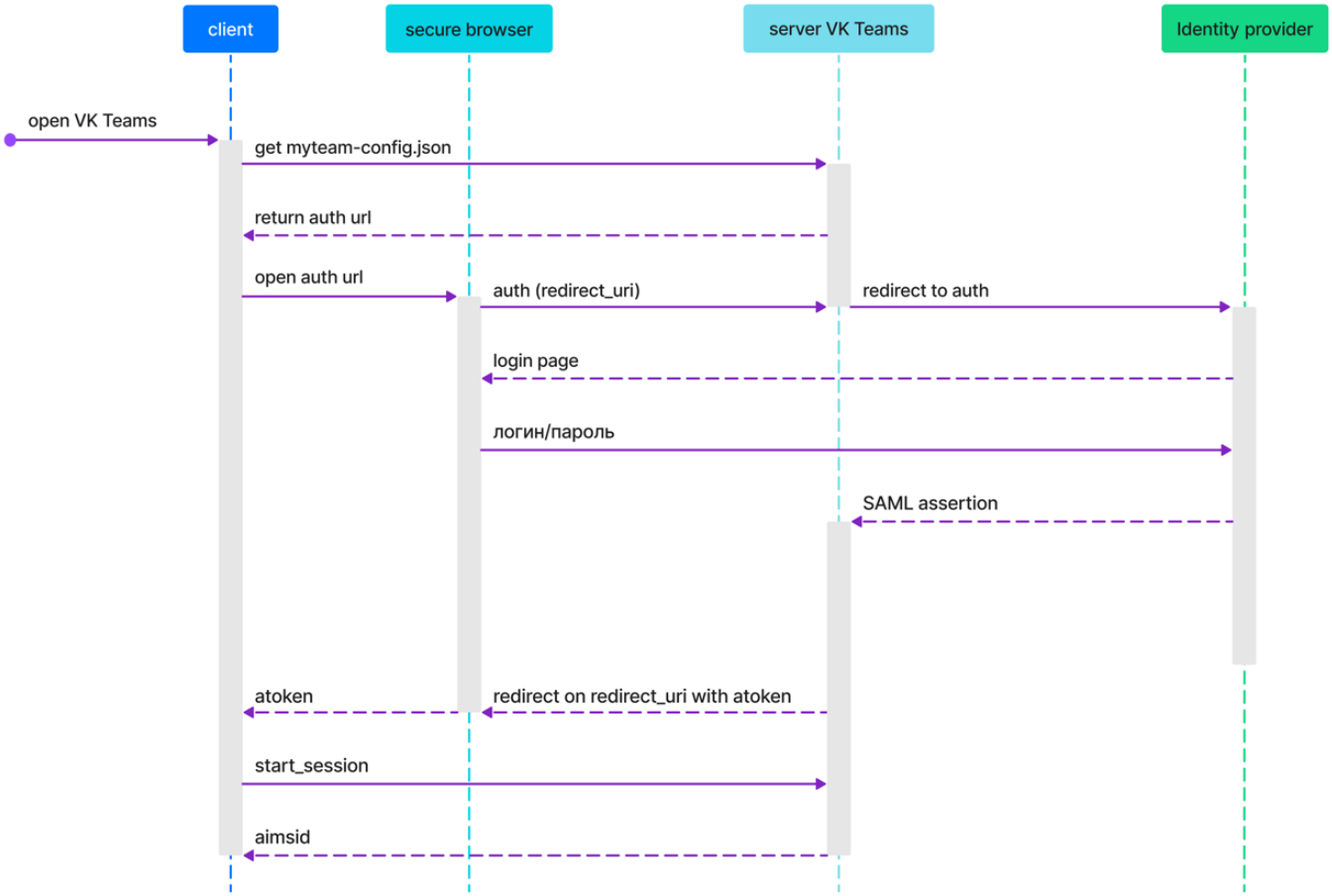
Механизм аутентификации по протоколу OIDC



1. Клиент из файла myteam-config.json получает auth-url для аутентификации (см. описание в разделе).
2. Сервер VK Teams составляет запрос в Identity Provider на /auth endpoint и перенаправляет на него клиента.
3. Пользователь (в secure browser) вводит аутентификационные данные:
    - если пользователем уже вошел в Систему, сработает SSO, и пользователю ничего вводить не потребуется;
    - в случае ошибки логина/пароля — об этом пользователю сообщит Identity Provider внутри secure_browser в окне логина («Invalid username or password») и предложит ввести логин/пароль повторно.
4. Identity Provider перенаправляет на указанный сервером redirect_uri, находящийся на сервере.
5. Сервер VK Teams обрабатывает redirect от Identity Provider:
    - в параметрах запроса получает:
        - state;
        - code.

6. Сервер VK Teams составляет и отправляет запрос в Identity Provider на /token endpoint:
    - в ответ получает необходимые токены и периоды их жизни (access_token, refresh_token, id_token и т.д.)
    - сохраняет их, если ещё нет токенов для этого пользователя в хранилище сервера.
7. Сервер VK Teams завершает действия, необходимые для аутентификации, и осуществляет редирект на переданный redirect_uri, в параметрах передавая результат:
    - code:
        - 20000 – успех;
        - 50000 - server error.
    - reason — передается в случае ошибки;
    - atoken;
    - email;
    - host_time.

Механизм аутентификации по протоколу SAML



1. Клиент из файла myteam-config.json получает auth-url для аутентификации (см. описание в разделе).
2. Сервер VK Teams составляет запрос в Identity Provider на /auth endpoint и перенаправляет на него клиента.
3. Пользователь (в secure browser) вводит аутентификационные данные:
    - если пользователем уже вошел в Систему, сработает SSO, и пользователю ничего вводить не потребуется;
    - в случае ошибки логина/пароля — об этом пользователю сообщит Identity Provider внутри secure_browser в окне логина («Invalid username or password») и предложит ввести логин/пароль повторно.
4. Identity Provider перенаправляет на указанный сервером redirect_uri, находящийся на сервере.
5. Сервер VK Teams обрабатывает redirect от Identity Provider:
    - в параметрах запроса получает:
        - state;
        - code.

6. Сервер VK Teams завершает действия, необходимые для аутентификации, и осуществляет редирект на переданный redirect_uri, в параметрах передавая результат:
    - code:
        - 20000 – успех;
        - 50000 - server error.
    - reason — передается в случае ошибки;
    - atoken;
    - email;
    - host_time.

Настройка SSO аутентификации по протоколам OIDC и SAML

Необходимые шаги для включения SSO аутентификации представлены ниже:

Шаг 1. Настройка подсистемы авторизации сервера VK Teams

1. Перейти в веб-интерфейс сервиса Keycloak:

   - открыть доступ для домена mridme. <DOMAIN> и перейти в браузере на https://mridme.<DOMAIN>

     > **Примечание**
     >
     > По умолчанию имя mridme не заведено в DNS, и в настройках nginx выставлено deny all. Не рекомендуется использовать этот способ доступа без крайней необходимости.

     или

   - пробросить локальный порт на сервер:

     ```
     ssh -L 8080:keycloak-http.keycloak.svc.cluster.local:80 centos@<server>
     ```

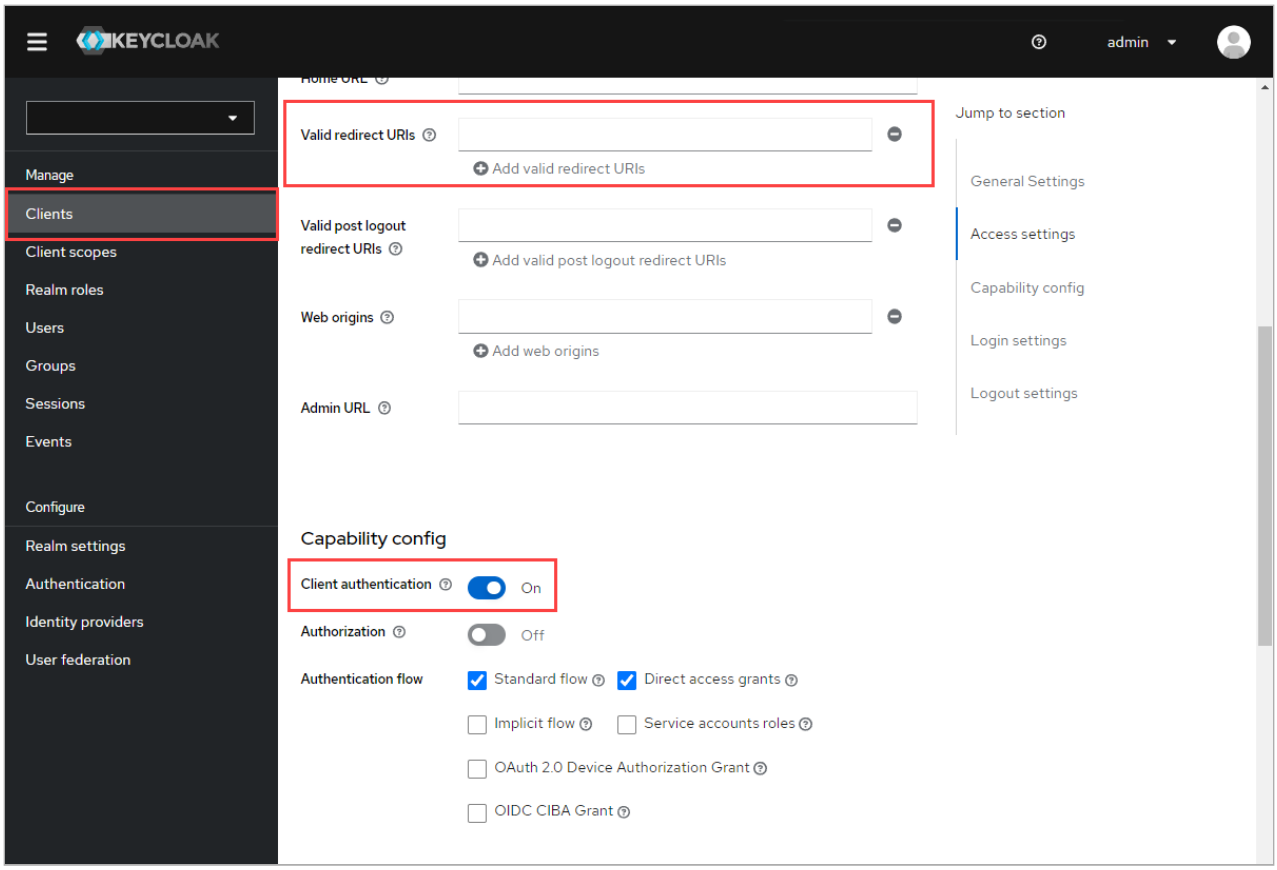
     и перейти в браузере http://127.0.0.1:8080/auth

2. Логин: admin

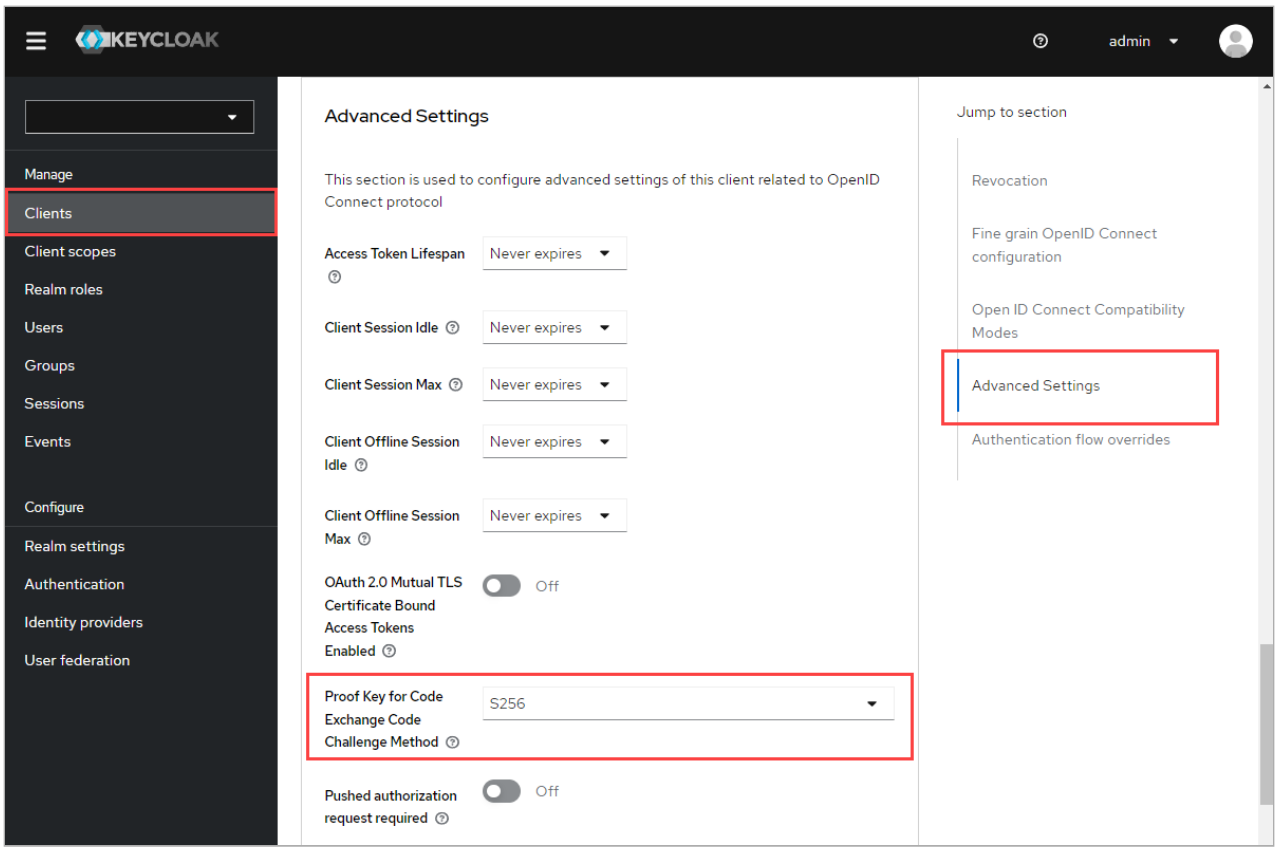
   Пароль: пароль необходимо получить в службе технической поддержки

3. Перейти **Manage** → **Clients** → выбрать **nomailcli** → вкладка **Settings** → установить значения для полей:

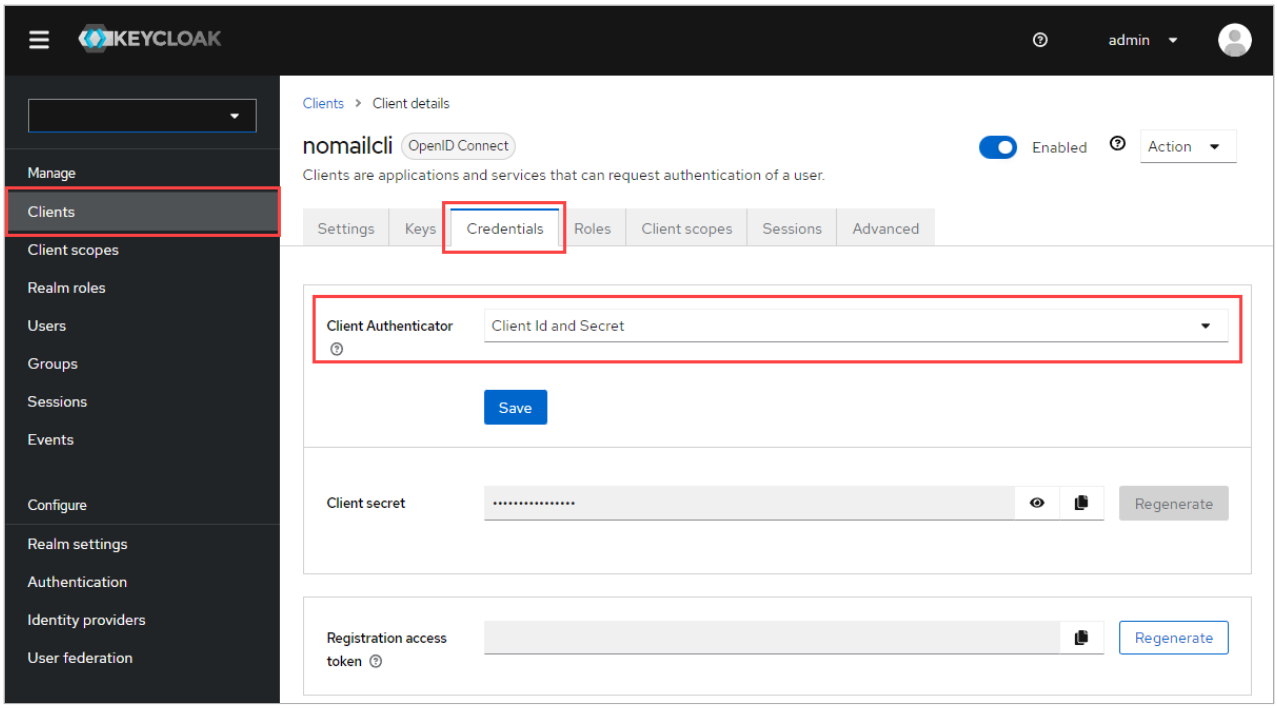
   - **Valid redirect URIs** — https://u.<DOMAIN_EXAMPLE.COM>/api/v87/rapi/auth/oidc/submitCode, где <DOMAIN_EXAMPLE.COM> — ваш домен;
   - **Client authentication** — **On**:



- Вкладка **Advanced** → **Advanced Settings** → поле **Proof Key for Code Exchange Code Challenge Method** → указать **S256**:

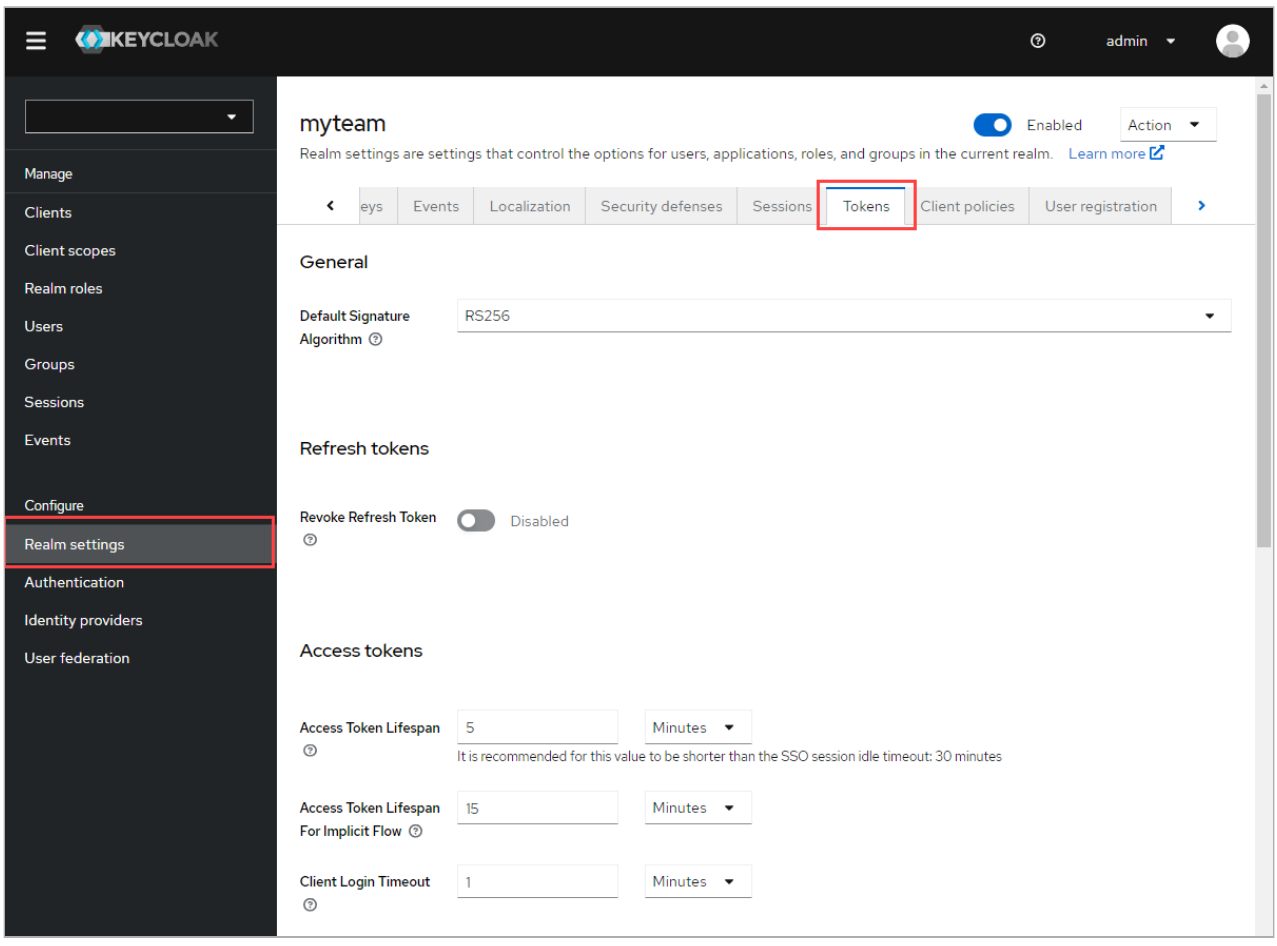


- Вкладка **Credentials** → поле **Client Authenticator** → указать **Client Id and Secret**:

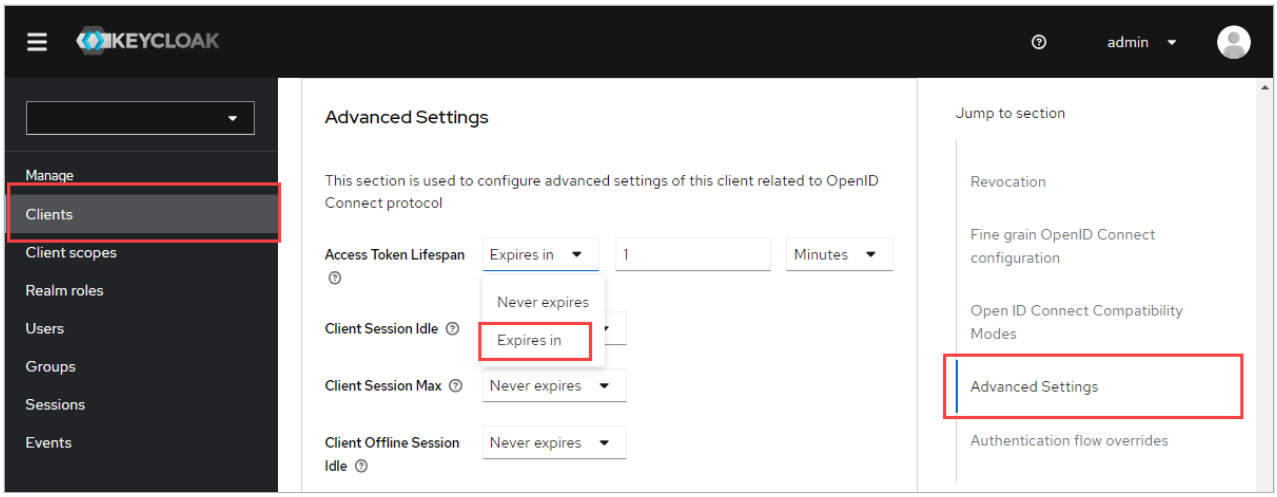


4. Необязательные параметры:

- Перейти **Realm settings** → вкладка **Tokens**:
  можно указать время жизни различных токенов (на весь realm):

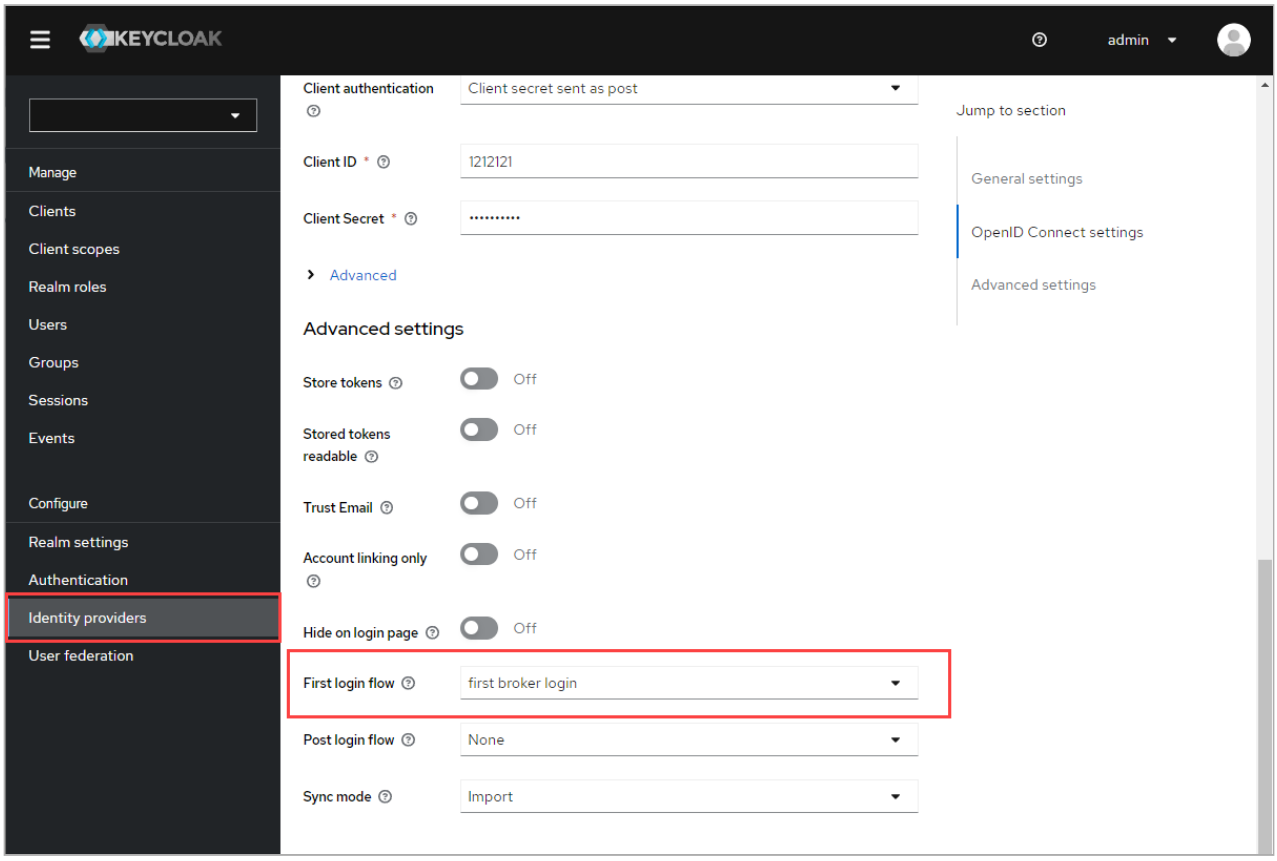


- Перейти **Clients** → выбрать **nomailcli** → вкладка **Advanced** → **Advanced Settings**:
  можно указать время жизни access_token:



5. Проставить поведение при первом логине:

- Перейти **Configure** → вкладка **Identity providers** → выбрать провайдера → в поле **First login flow** указать «**first broker login**»:

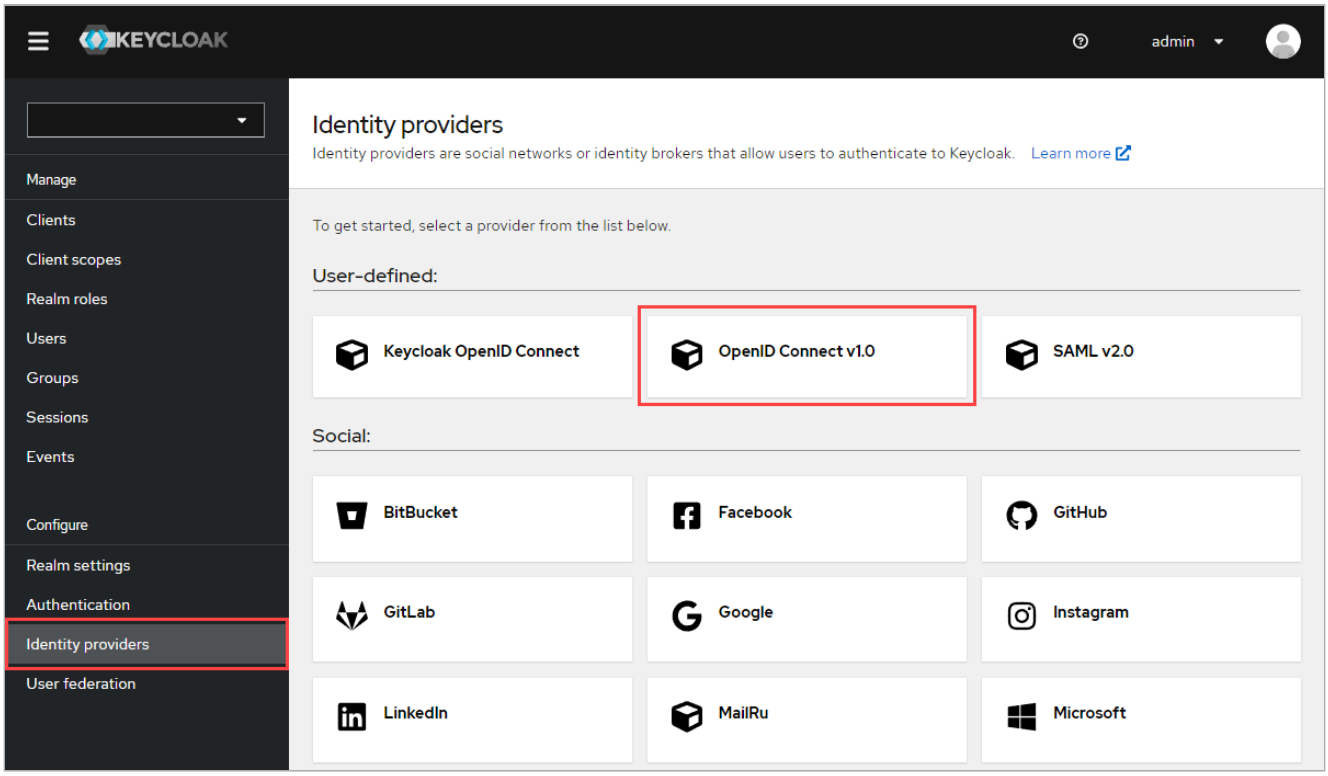


Шаг 2. Добавление провайдера аутентификации

Протокол OIDC

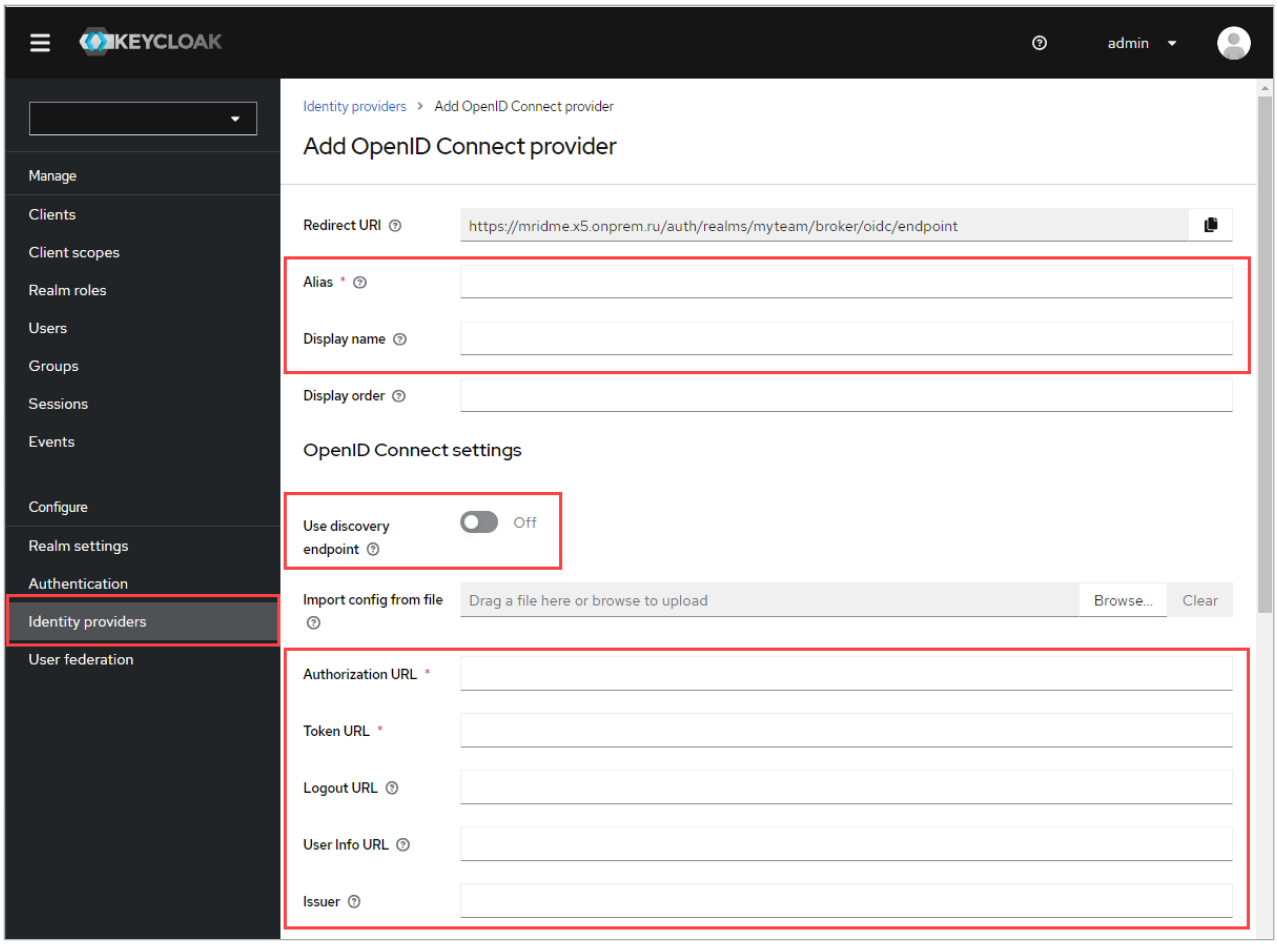
Ниже представлено добавление провайдера аутентификации при использовании протокола OIDC. Если Вы используете протокол SAML, перейдите к разделу Протокол SAML.

1. Проверить доступность адреса mridme.<DOMAIN_TEAMS>.
2. Создать и настроить провайдера аутентификации:
   Перейти **Identity Providers** → выбрать необходимый протокол:

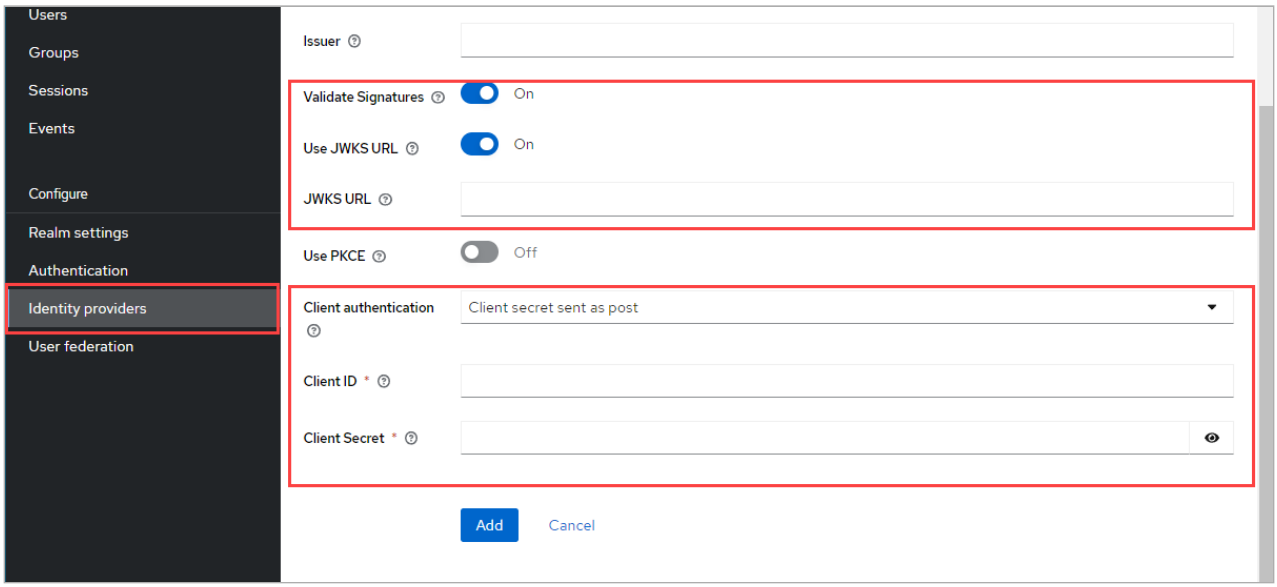


3. Указать следующие значения полей:

   - **Alias** — задать Alias провайдера;
   - **Display name** — задать имя провайдера:
   - **Use discovery endpoint** — **Off**;
   - **Authorization URL** — запросить у администратора Authentication server;
   - **Token URL** — запросить у администратора Authentication server;
   - **Logout URL** — запросить у администратора Authentication server;
   - **User Info URL** — запросить у администратора Authentication server;
   - **Issuer** — запросить у администратора Authentication server:



- **Validate Signatures** — **On**;
- **Use JWKS URL** — **On**;
- **JWKS URL** — запросить у администратора Authentication server;
- **Client authentication** — **Client secret sent as post**;
- **Client ID** — запросить у администратора Authentication server;
- **Client Secret** — запросить у администратора Authentication server:

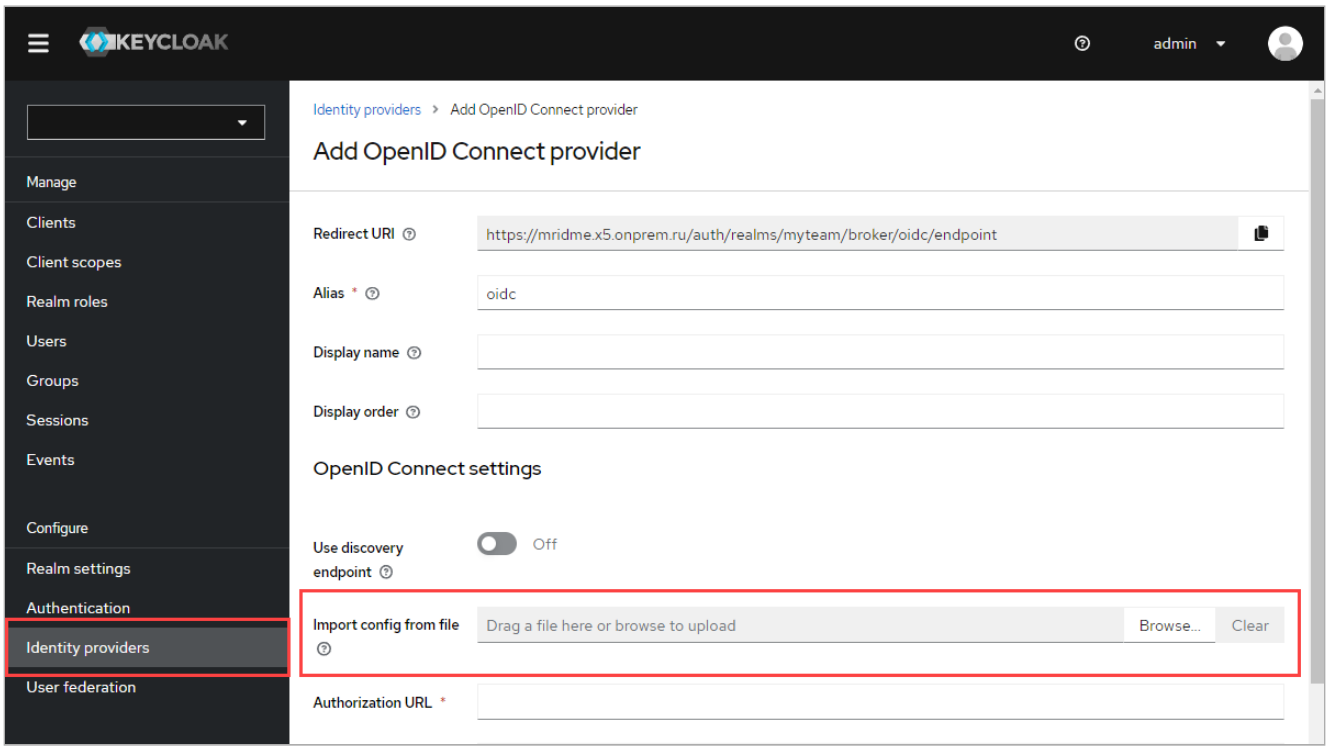


> **Примечание**
>
> Данные поля также можно заполнить, импортировав файл конфигурации. Запросить Import External Config можно у администратора Authentication server.

- Перетащить файл в поле **Import config from file**

  или

- В поле **Import config from file** нажать **Browse** → выбрать файл <Import External Config>:

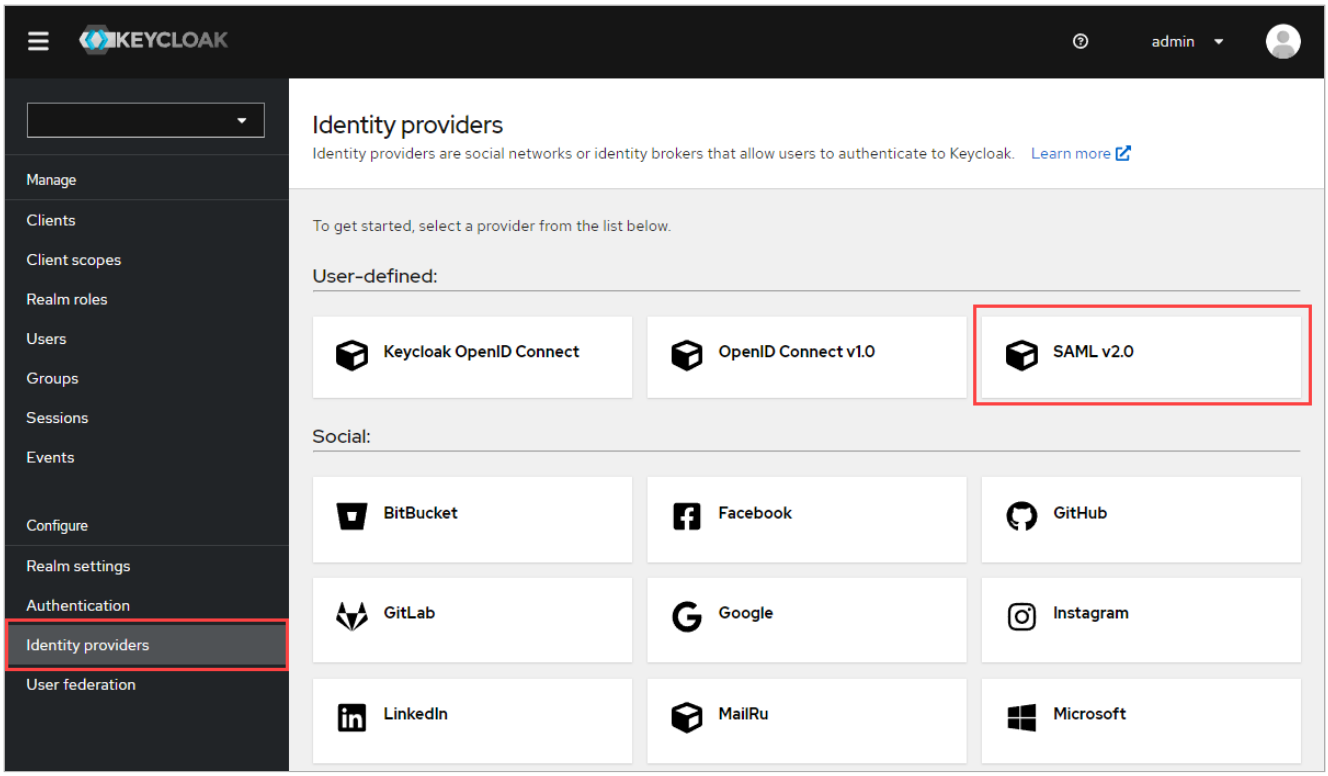


4. Нажать **Save**.

Протокол SAML

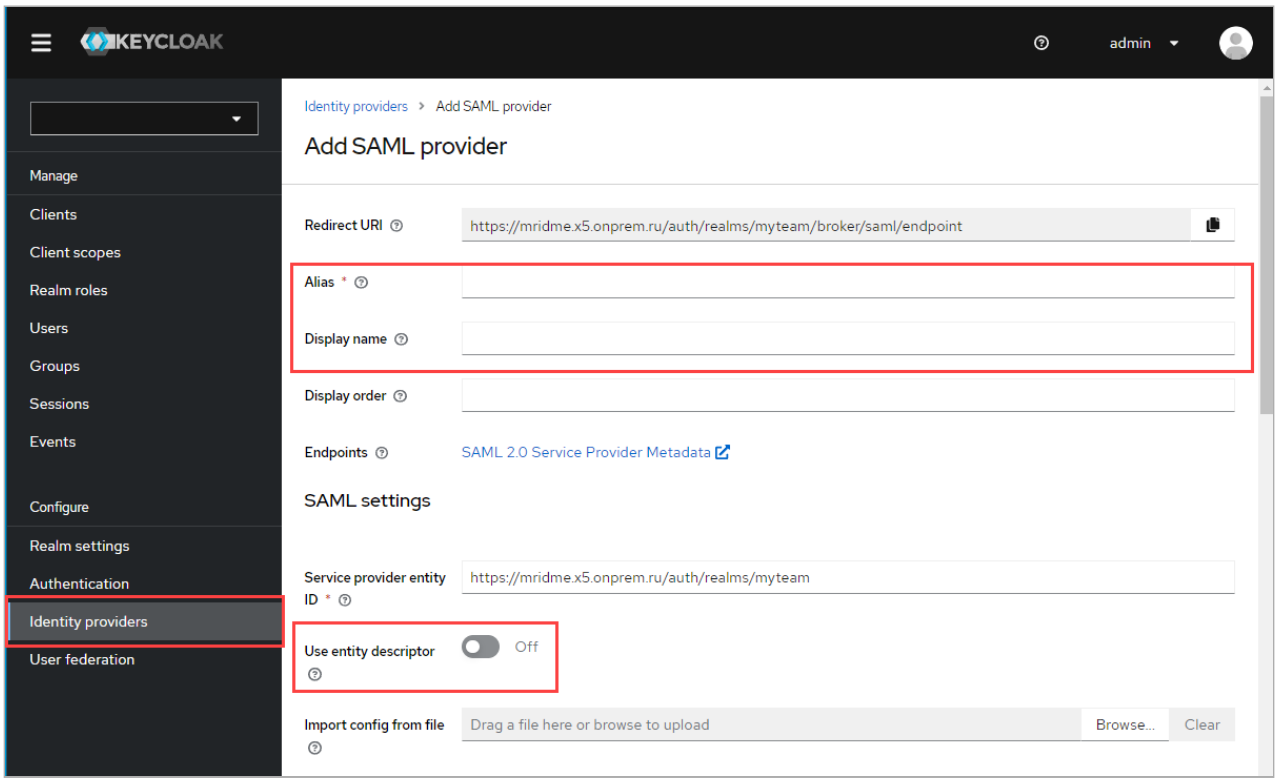
Ниже представлено добавление провайдера аутентификации при использовании протокола SAML. Если Вы используете протокол OIDC, перейдите к разделу Шаг 3. Регистрация провайдеров аутентификации в сервисах VK Teams.

1. Проверить доступность адреса mridme.<DOMAIN_TEAMS>.
2. Создать и настроить провайдера аутентификации:
   Перейти **Identity Providers** → выбрать необходимый протокол:



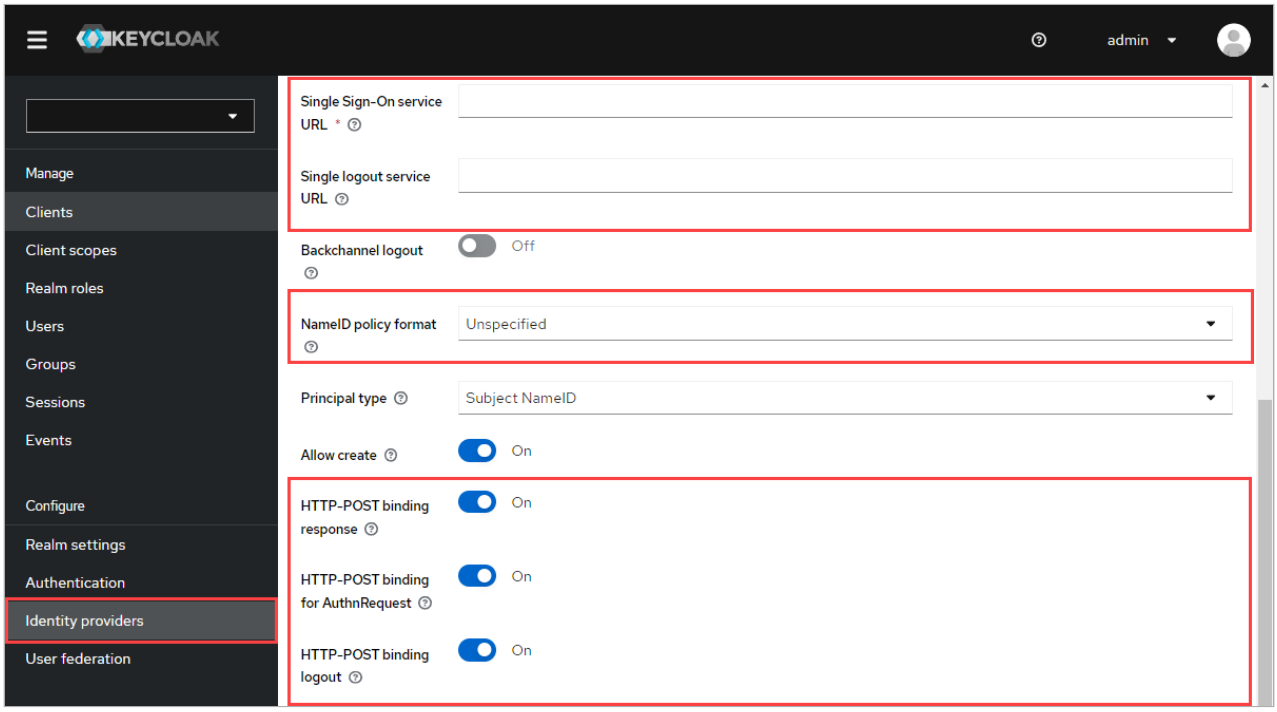
3. Указать следующие значения полей:

- **Alias** — задать Alias провайдера;
- **Display name** — задать имя провайдера;
- **Use entity descriptor** — **Off**:

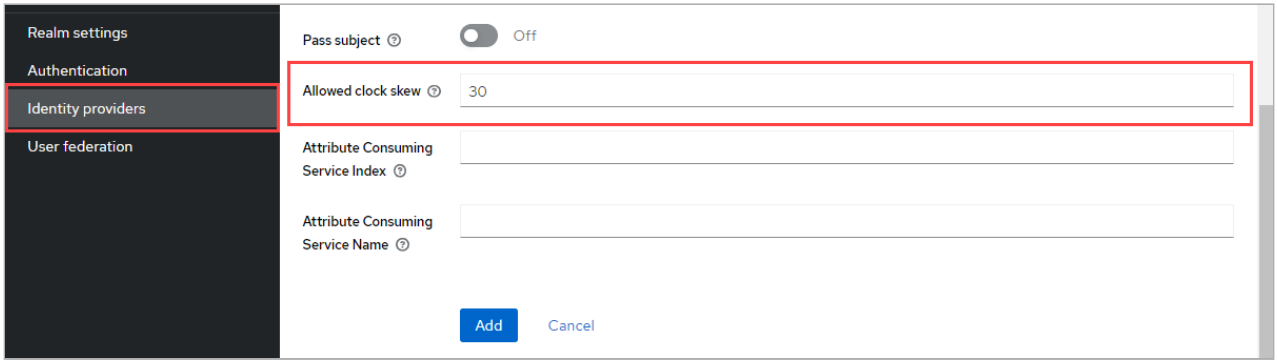


- **Single Sign-On service URL** — запросить у администратора IDP-сервера;
- **Single logout service URL** — запросить у администратора IDP-сервера;
- **NameID policy format** — **Unspecified**;
- **HTTP-POST binding response** — **On**;

- **HTTP-POST binding for AuthnRequest — On**;
- **HTTP-POST binding logout — On**:



- **Allowed clock skew — 30**:

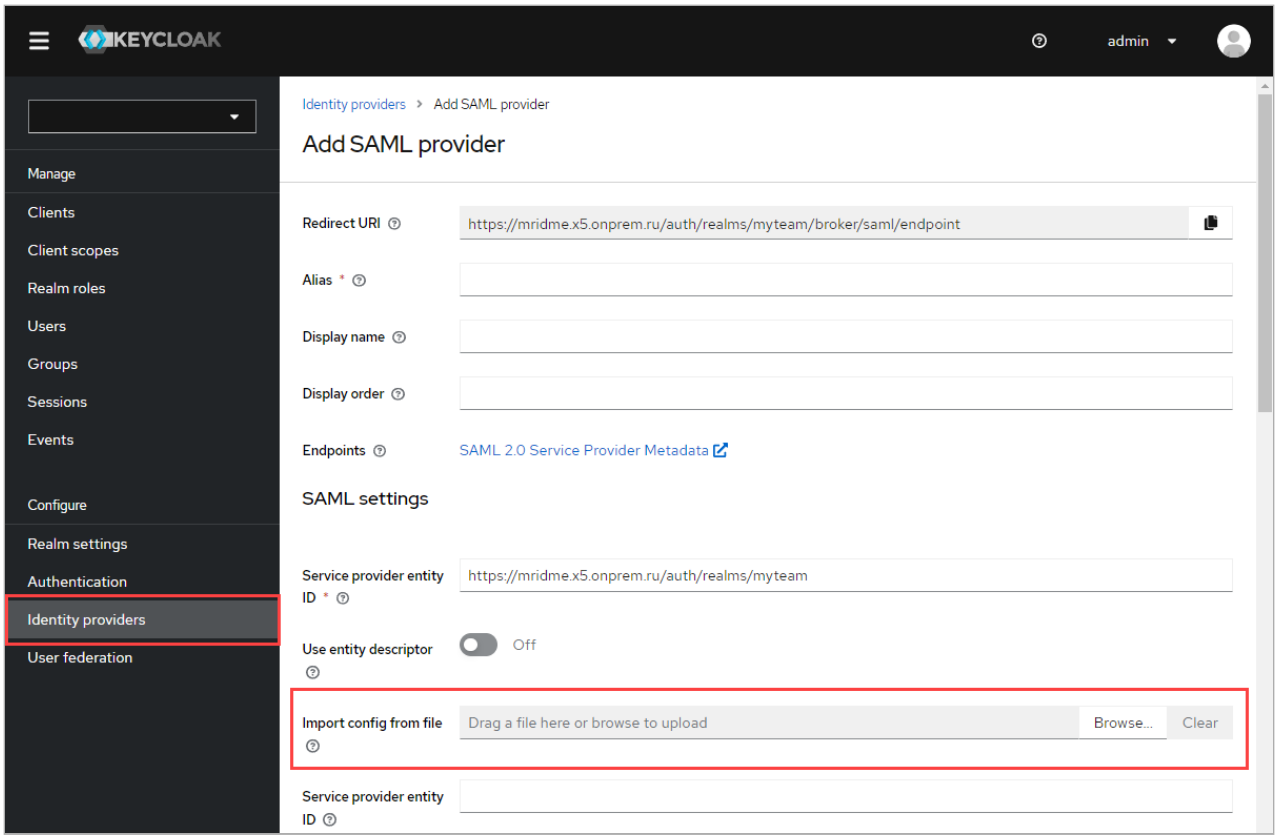


> **Примечание**
>
> Данные поля также можно заполнить, импортировав файл конфигурации. Запросить Import External Config можно у администратора IDP-сервера.

- Перетащить файл в поле **Import config from file**

или

- В поле **Import config from file** нажать **Browse** → выбрать файл <Import External Config>:



4. Нажать **Save**.

Шаг 3. Регистрация провайдеров аутентификации в сервисах VK Teams

Провайдеры регистрируются в сервисе Stdb. Оттуда информацию о них получают сервисы Front и Tokeeper.

SSO аутентификация поддерживает аутентификацию через несколько провайдеров.

Возможна поддержка нескольких провайдеров в двух форматах:

Вариант 1. Сервис Keycloak подключается к провайдеру аутентификации в режиме посредника, все взаимодействие с провайдером лежит на сервисе Keycloak.

Настроить выбор провайдера для различных вариантов подключения:

- Подключиться к сервису Stdb:

```
rlwrap nc 0.0.0.0 4041
```

- Далее добавить таблицу с данными провайдеров:

```
stdb_table_add idp_configurations issuer@string addr@string client_id@string scope@string
client_secret@string platforms_and_auth_extra_params@string need_register_user@string

stdb_row_add idp_configurations KK https://di.<DOMAIN_EXAMPLE.COM>/auth/realms/
```

```
myteam/.well-known/openid-configuration nomailcli openid use_secrets_luke '{ "web":
"kc_idp_hint=<Alias_1>", "desktop": "kc_idp_hint=<Alias_2>", "default":
"kc_idp_hint=<Alias_3>" }' false
```

, где `<Alias_1>`, `<Alias_2>`, `<Alias_3>` — значение поля **Alias** для провайдеров в Keycloak (см. Шаг 2. Добавление провайдера аутентификации).

Для разграничения платформ используется поле **platforms_and_auth_extra_params** таблицы сервиса Stdb. Уточнения значения поля **platforms_and_auth_extra_params**:

- **default** — переходим на базовую страницу авторизации сервиса Keycloak;
- **kc_idp_hint=<**Alias провайдера в настройках Keycloak**>**.

Доступные платформы:

- Web;
- Android;
- Desktop;
- IOS.

Внимание

Если одну и ту же платформу указать для нескольких провайдеров, сервис Stdb сообщит об этом в лог, SSO аутентификация работать не будет.

Примечание

Полезные команды в rlwrap:

- get //получить список:

```
stdb_table_get idp_configurations
```

- del // удалить:

```
stdb_row_del idp_configurations 1
```

- set // изменить:

```
stdb_row_set idp_configurations 1 KK http://di.<DOMAIN_EXAMPLE.COM>/auth/realms/myteam/.well-
known/openid-configuration nomailcli openid use_secrets_luke '{ "web": "kc_idp_hint=saml",
"desktop": "kc_idp_hint=saml", "default": "kc_idp_hint=ws1" }' false
```

Вариант 2. Отдельная регистрация каждого провайдера: в таблицу каждый провайдер добавляется новой строкой.

Шаг 4. Настройка внешней аутентификации

1. Добавить в **/usr/local/nginx-im/html/myteam/myteam-config.json** указанное содержимое:

```
"oauth-authorization": {
  "enabled": true,
  "config": {
  "auth-url": "https://u.<DOMAIN_EXAMPLE.COM>/api/v87/rapi/auth/oidc/authorize"
  }
},
```

2. Применить:

```
kubectl delete pod myteam-admin-* -n vkteams
```

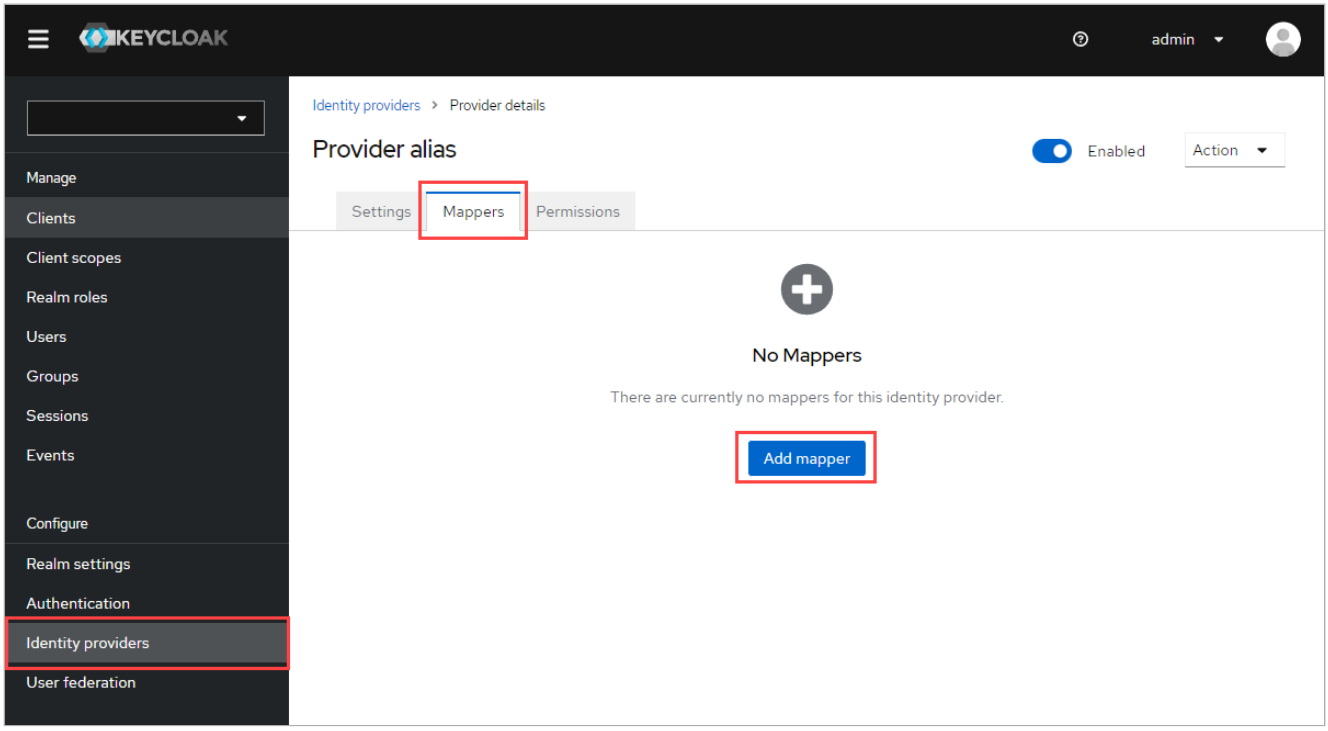
, где: ***** — уникальное имя пода. Имя пода необходимо получить с помощью вывода команды:

```
 kubectl get pods -A | grep admin
```

Шаг 5. Настройка protocol mappers

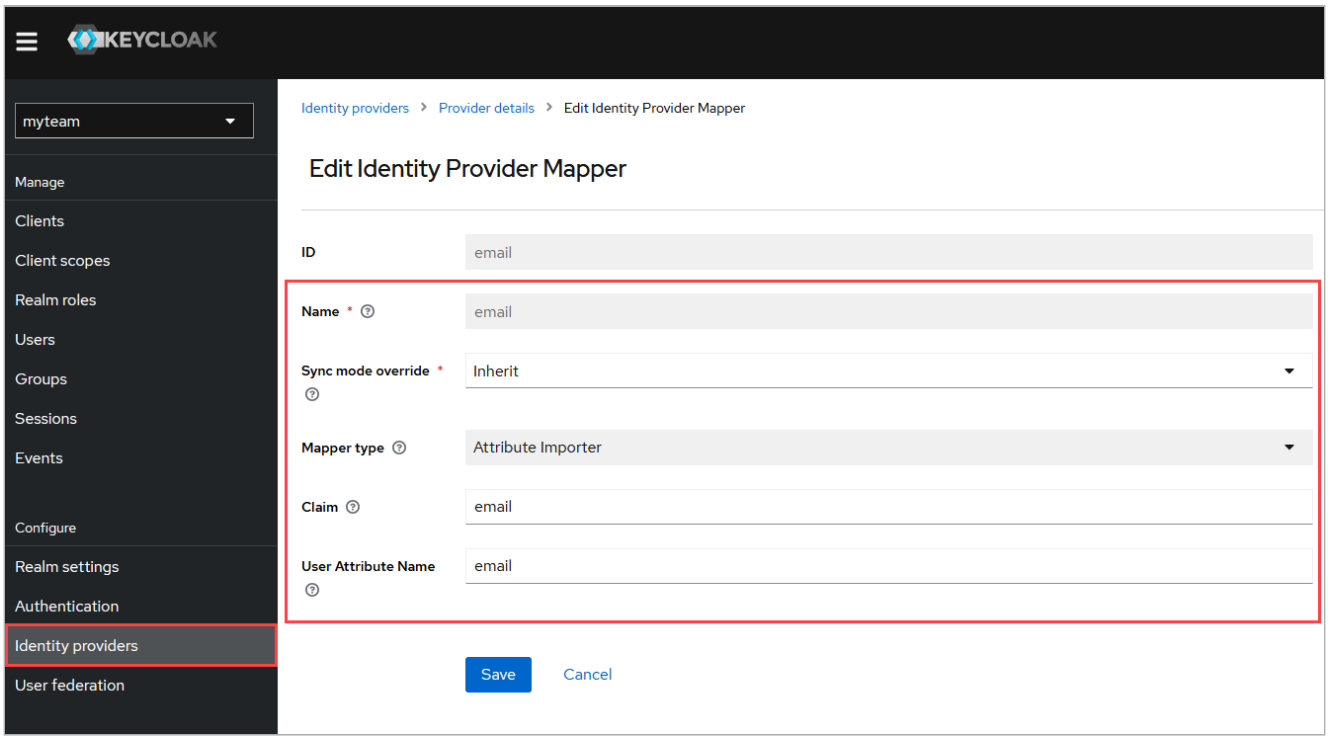
Необходимо настроить по три mapper'а для каждого провайдера — **email**, **lastName** и **firstName**.

1. Перейти **Identity Providers** → выбрать провайдера → вкладка **Mappers** → нажать **Add mapper**:



2. Указать следующие значения полей:
   - **Name** — задать имя mapper'а;
   - **Sync mode override** — **Inherit**;

- **Mapper type** — **Attribute Importer**;
- **Claim** — маска для поиска атрибута в токене (запросить у администратора IDP-сервера);
- **User Attribute Name** — прописать один из вариантов — **email**; **lastName** или **firstName**:



3. Нажать **Save**.
4. Повторить шаги 1-4 для создания остальных двух mappers.

Настройка SSO аутентификации по протоколу Kerberos в Microsoft Active Directory

Шаг 1. Создание файла .keytab

В зависимости от выбранного типа шифрования — RC4-HMAC-NT или AES128-SHA1, AES256-SHA1 — выполните шаги, представленные ниже:

При использовании шифрования RC4-HMAC-NT

1. На Windows Server сгенерировать файл **.keytab** для Kerberos аутентификации в Active Directory :

```
ktpass -princ HTTP/computer.contoso.com@CONTOSO.COM -mapuser keycloak -pass "z7A&piloNu" -
crypto RC4-HMAC-NT -ptype KRB5_NT_PRINCIPAL -out mcs.keytab
```

, где:

- `-princ` —FQDN сервера Keycloak в формате **HTTP/computer.contoso.com@CONTOSO.COM** для организации связи между сервисом Keycloak и Active Directory

> **Примечание**
>
> Данный параметр учитывает регистр.

- `-mapuser` — пользователь, для которого регистрируется SPN и генерируется файл **.keytab**;
- `-pass` — пароль пользователя;
- `-crypto` — тип шифрования. Чтобы сгенерировать файл **.keytab**, поддерживающий все способы шифрования, укажите для ключа `-crypto` значение `ALL`;
- `-ptype` — тип принципала;
- `-out` — имя создаваемого файла **.keytab**.

2. Дополнительно включить шифрование RC4-HMAC-NT в контейнере с Keycloak (оно автоматически отключается, так как считается слабым):

- создать файл через любой текстовый редактор (название указать любое, в примере использовано название файла **allow-weak**) со следующим содержимым:

```
[libdefaults]
 allow_weak_crypto = true
 permitted_enctypes = aes256-cts-hmac-sha1-96 aes256-cts-hmac-sha384-192 camellia256-
cts-cmac aes128-cts-hmac-sha1-96 aes128-cts-hmac-sha256-128 camellia128-cts-cmac
arcfour-hmac
```

- создать **configMap** на основе файла, созданного на предыдущем шаге:

```
kubectl create configmap krb5-week-conf --from-file=allow-weak --namespace=keycloak
```

- чтобы отредактировать deployments выполните команду:

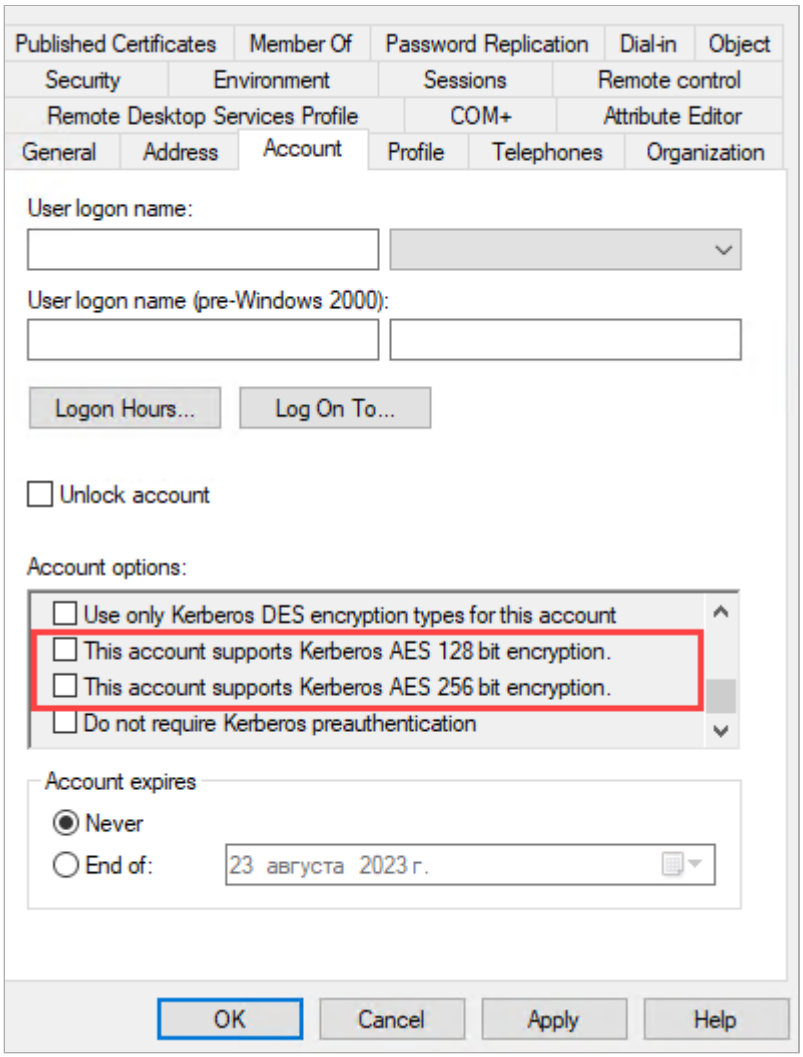
```
kubectl -n keycloak edit deployments
```

- добавить строки в секции `volumeMounts:` и `volumes:`

```
spec:
  template:
    spec:
      containers:
        volumeMounts:
        - mountPath: /etc/krb5.conf.d/
          name: krb5-week-conf
        volumes:
        - configMap:
          defaultMode: 420
          name: krb5-week-conf
        name: krb5-week-conf
```

При использование шифрования AES128-SHA1, AES256-SHA1

1. При использовании шифрования AES128-SHA1, AES256-SHA1 необходимы настройки для пользователей в Active Directory. В свойствах учетных записей пользователей необходимо установить поддержку типов шифрования AES128-SHA1, AES256-SHA1 — либо через групповые политики, либо вручную.



2. На Windows Server сгенерировать файл **.keytab** для Kerberos аутентификации в Active Directory :

```
ktpass -princ HTTP/computer.contoso.com@CONTOSO.COM -mapuser keycloak -pass "z7A&piloNu" -
crypto AES128-SHA1 -crypto AES256-SHA1 -ptype KRB5_NT_PRINCIPAL -out mcs2.keytab
```

, где:

- `-princ` —FQDN сервера Keycloak в формате **HTTP/computer.contoso.com@CONTOSO.COM** для организации связи между сервисом Keycloak и Active Directory

> **Примечание**
>
> Данный параметр учитывает регистр.

- `-mapuser` — пользователь, для которого регистрируется SPN и генерируется файл **.keytab**;

- `-pass` — пароль пользователя;
- `-crypto` — тип шифрования. Чтобы сгенерировать файл **.keytab**, поддерживающий все способы шифрования, укажите для ключа `-crypto` значение `ALL`;
- `-ptype` — тип принципала;
- `-out` — имя создаваемого файла **.keytab**.

3. Создать секрет из файла **.keytab** и прокинуть его внутрь контейнера с Keycloak:

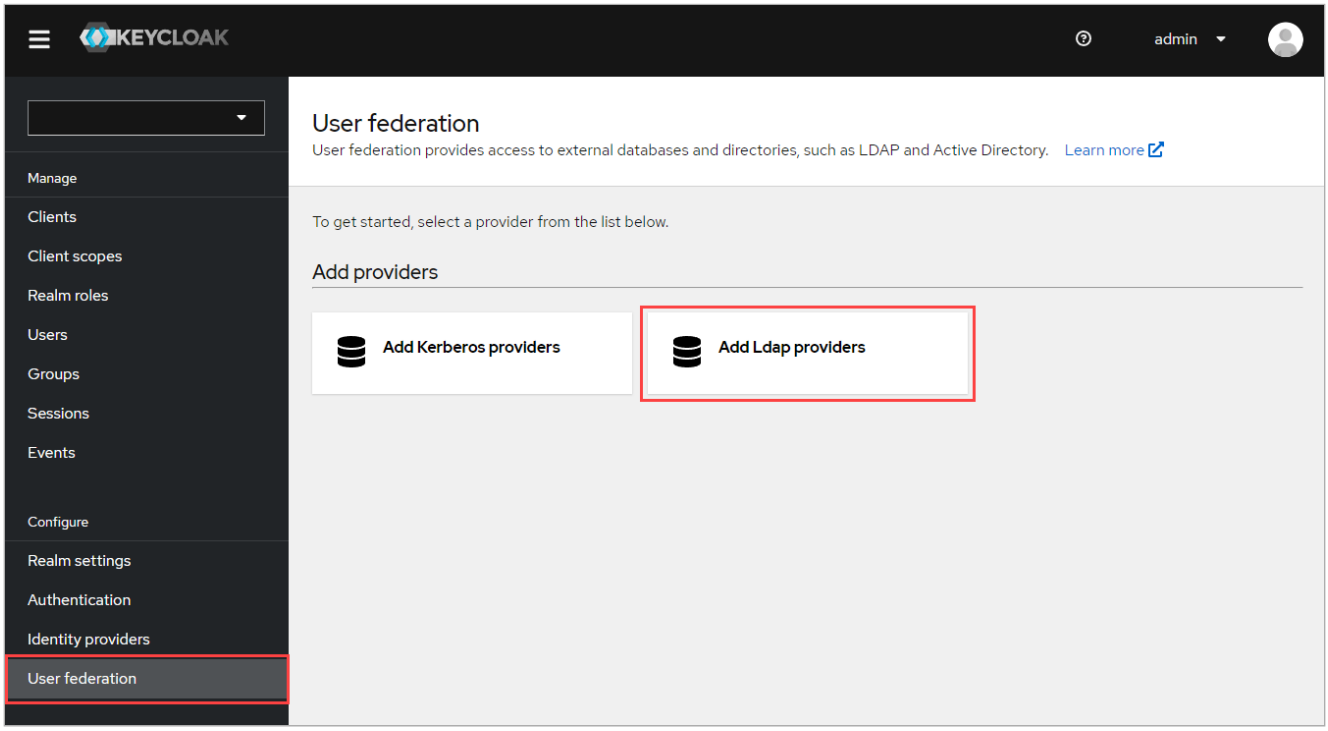
```
kubectl -n keycloak create secret generic keycloak-keytab --from-file=mcs_new.keytab --dry-run=client -o yaml | kubectl apply -f -
```

Шаг 2. Настройка realm

Перейти в веб-интерфейс сервиса Keycloak и настроить realm. Подробное описание представлено в разделе.

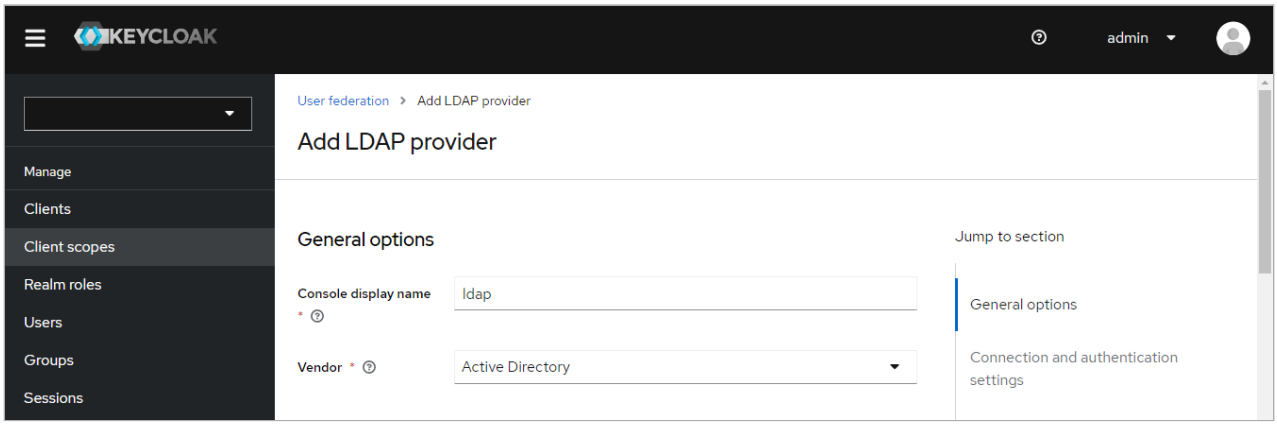
Шаг 3. Подключение пользователей из Keycloak через User Federation

1. Перейти в раздел **Configure** → **User federation** → нажать на кнопку **Add Ldap providers**:



2. Установить следующие значения полей:

- **Vendor** — **Active Directory**:



- **Connection URL** – IP-адрес контроллера домена Active Directory.

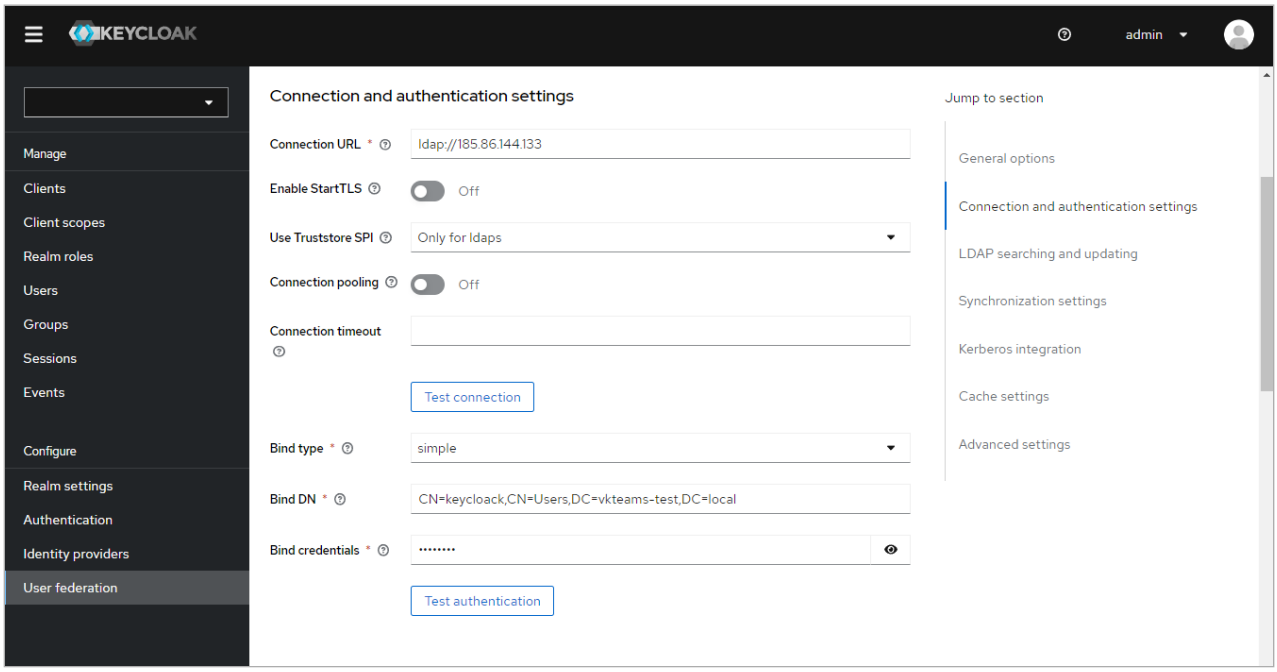
  При использовании защищенного соединения, указать протокол **ldaps** и сделать активными переключатели:

  - **Enable StartTLS** – **On**;
  - **Connection pooling** – **On**.

  Для проверки настроенного соединения нажать на кнопку **Test connection**. При успешном соединении получаем сообщение «Successfully connected to LDAP».

- Указать Bind DN и пароль пользователя, от имени которого планируется подключаться к Active Directory.

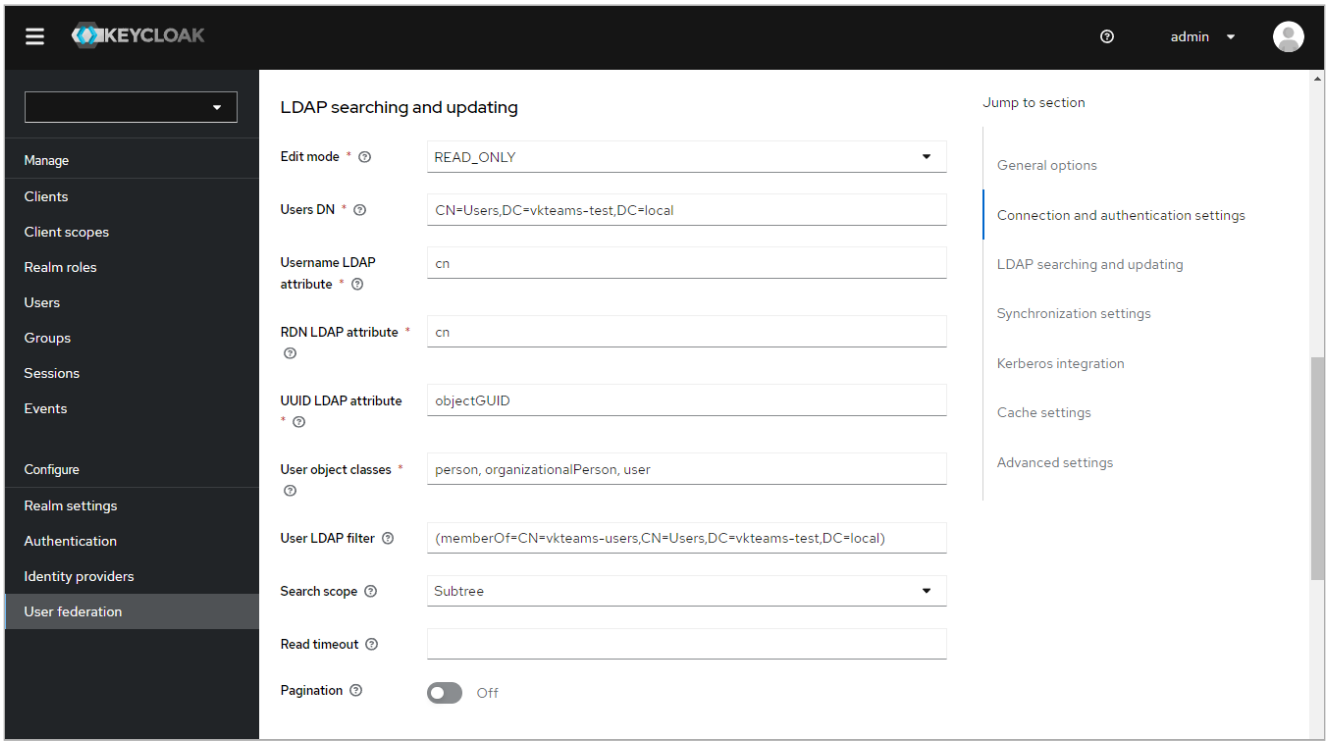
  Для проверки подключения к Active Directory нажать на кнопку **Test authentication**. При успешном соединении получаем сообщение «Successfully connected to LDAP».



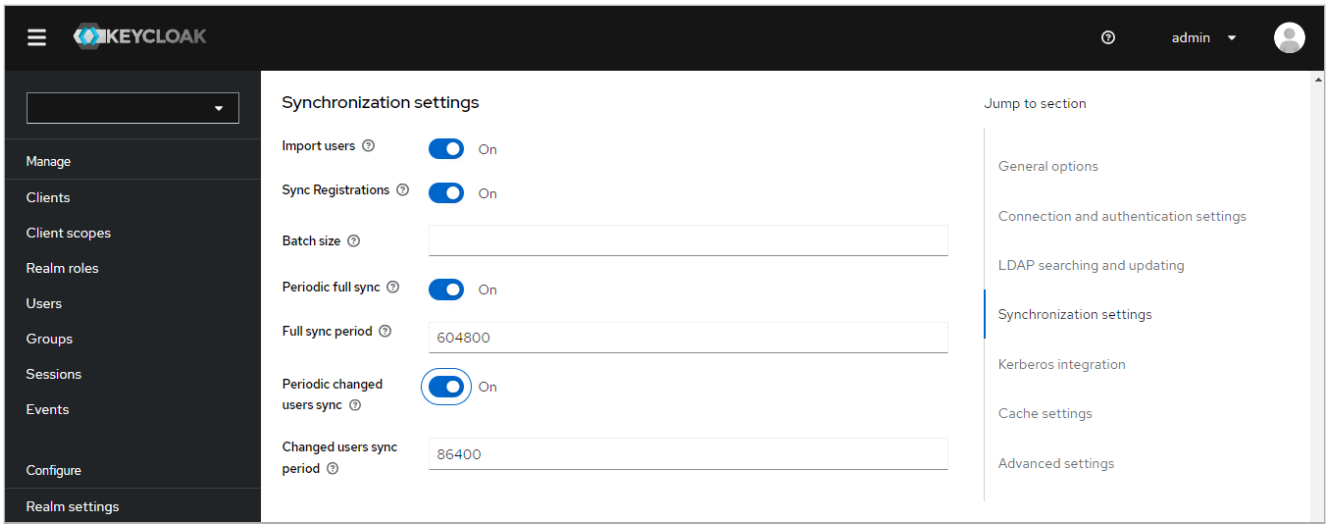
3. В блоке с настройками поиска и обновления LDAP указать следующие значения полей:

- **Edit mode** — **READ_ONLY**;
- **UsersDN** — атрибут **distinguishedName** из Active Directory;
- **Username LDAP attribute** — **cn**;

- **RDN LDAP attribute** — **cn**;
- **UUID LDAP attribute** — **objectGUID**;
- **User object classes** — **person, organizationalPerson, user**;
- **User LDAP filter** — опциональный фильтр, который указывает, из какой группы в Active Directory брать пользователей;
- **Search scope** — **Subtree** для сквозного поиска пользователей согласно фильтру в поле **User LDAP filter**.

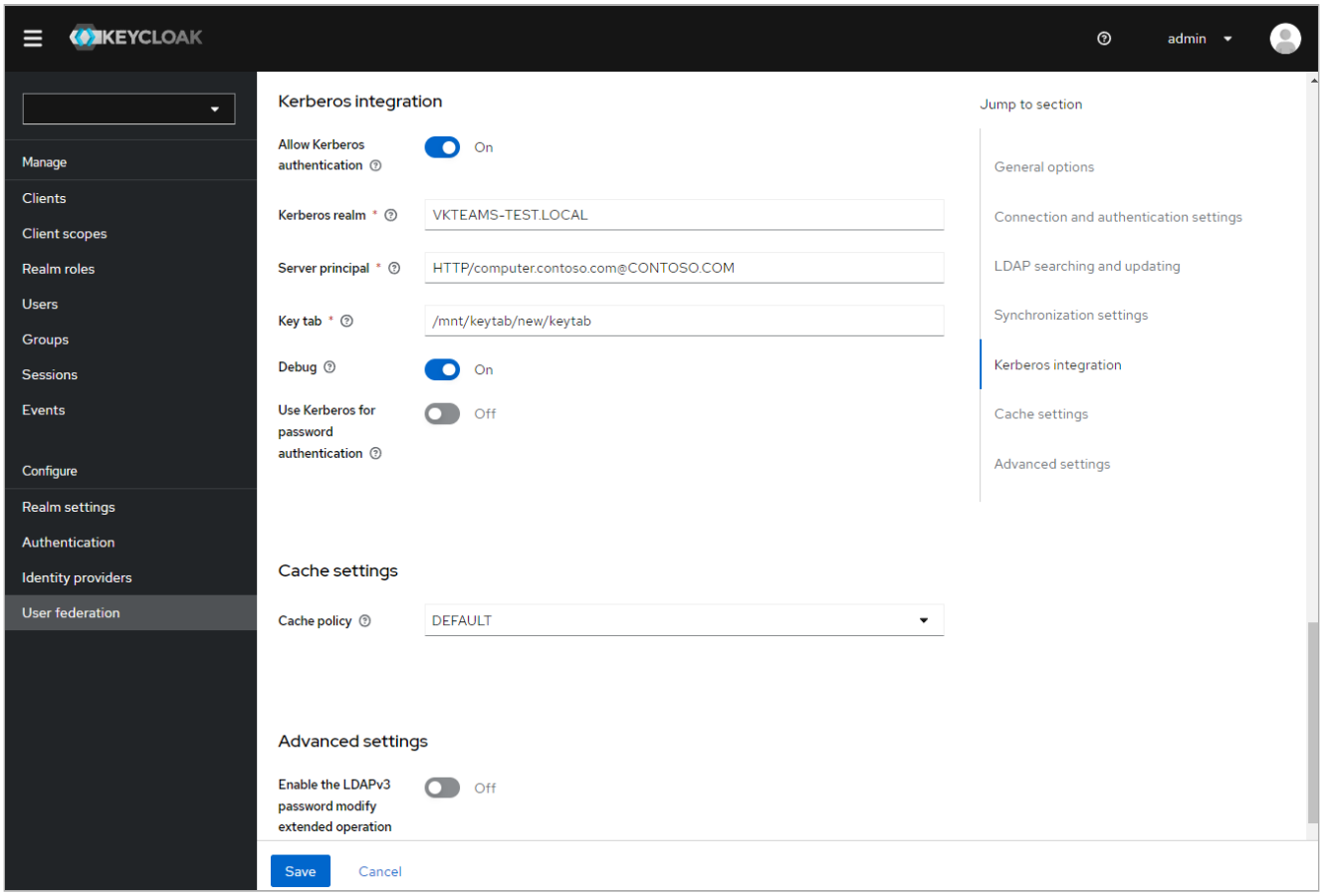


4. В блоке с настройками синхронизации указать следующие значения полей:
   - **Import users** — **On**;
   - **Sync Registrations** — **On**;
   - **Periodic full sync** — **On;**
   - **Full sync period** — указать период синхронизации в секундах;
   - **Periodic changed users sync — On**;
   - **Changed users sync period** — указать период синхронизации в секундах.



5. В блоке с настройками Kerberos указать следующие значения полей:

   - **Allow Kerberos authentication** — **On**;
   - **Kerberos realm** — заглавными буквами указать наименование домена, настроенного на шаге 2;
   - **Server principal** — указать SPN, указанный при создании файла **.keytab** (см. шаг 1);
   - **Key tab** — указать путь до файла .**keytab**;
   - **Debug** – **On** (опционально).



6. Нажать на кнопку **Save**.

Шаг 4. Регистрация Keycloak в сервисе Stdb

Описание представлено в разделе.

Шаг 5. Настройка внешней аутентификации

Описание в разделе.

Распространенные проблемы

Проблема: надпись **Server error** в web-интерфейсе VK Teams, окно логина не открылось.

Решение: необходимо отключить блокировку всплывающих окон в браузере.

Проблема: вместо окна логина отображается **Required parameter not found**.

Решение: проверить, что в сервисе Stdb верно прописано поле **addr** в **idp_configurations**.

Проблема: после логина появляется ошибка **Unexpected error when authenticating with identity provider**,

в логах сервиса Keycloak: **Failed to make identity provider oauth callback: javax.net.ssl.SSLHandshakeException: PKIX path building failed: sun.security.provider.certpath.SunCertPathBuilderException: unable to find valid certification path to requested target**.

Решение: проверить, что у сервиса Keycloak есть все сертификаты.

Дата обновления документа: 11.10.2023 г.