

ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ»

СИСТЕМА РЕДАКТИРОВАНИЯ И СОВМЕСТНОЙ РАБОТЫ (СО)

СИСТЕМА ХРАНЕНИЯ ДАННЫХ (PGS)

3.3.1

**РУКОВОДСТВО ПО НАСТРОЙКЕ
ИНТЕГРАЦИИ С SSO**

Версия 1

На 24 листах

Дата публикации: 28.03.2025

**Москва
2025**

Все упомянутые в этом документе названия продуктов, логотипы, торговые марки и товарные знаки принадлежат их владельцам.

Товарные знаки «МойОфис», «MyOffice» и «Squadus» принадлежат ООО «НОВЫЕ ОБЛАЧНЫЕ ТЕХНОЛОГИИ».

Ни при каких обстоятельствах нельзя истолковывать любое содержимое настоящего документа как прямое или косвенное предоставление лицензии или права на использование товарных знаков, логотипов или знаков обслуживания, приведенных в нем.

Любое несанкционированное использование этих товарных знаков, логотипов или знаков обслуживания без письменного разрешения их правообладателя строго запрещено.

СОДЕРЖАНИЕ

1 Общие сведения	5
1.1 Назначение	5
1.2 О компонентах продукта	5
1.3 Архитектура решения	6
2 Настройка интеграции	8
2.1 Настройка внешнего SSO (на примере KeyCloak)	8
2.1.1 Создание OpenID Connect Client	8
2.1.2 Настройка access токена	10
2.1.3 Назначение роли администратора	11
2.1.4 Конфигурация времени жизни токенов / сессии	12
2.1.5 Конфигурация для сбора событий об изменении / удалении пользователей	14
2.2 Настройка компонента PGS	17
2.3 Настройка компонента СО	20
2.3.1 Подготовка конфигурационных файлов	20
2.3.2 Подготовка DNS-записи	21
2.3.3 Порядок установки сервиса	21
2.3.4 Проверка работы после установки	21
2.4 Настройка авторизации	22
2.4.1 Добавление пользователя	22
2.4.2 Получение токена авторизации	22
2.4.3 Использование токена в запросах	23
3 Ограничения	24
3.1 Отображение пользователей в продукте	24
3.2 Управление тенантами	24
3.3 Количество создаваемых тенантов	24
3.4 Изменение роли	24

ПЕРЕЧЕНЬ СОКРАЩЕНИЙ, ТЕРМИНОВ И ОПРЕДЕЛЕНИЙ

В настоящем документе применяют следующие сокращения с соответствующими расшифровками (табл. 1).

Таблица 1 — Сокращения и обозначения

Сокращение, термин	Расшифровка и определение
DNS	Domain Name System, система доменных имен
DU	Document Unit, синоним DCS
ETCD	Распределенная система хранения конфигурации
FQDN	Fully Qualified Domain Name, полностью определенное имя домена
Inventory	Файл, содержащий набор управляемых хостов для автоматизации установки и управления конфигурацией для сервиса Ansible
SSH	Secure Shell, «безопасная оболочка»
UI	User Interface, пользовательский интерфейс
URL	Uniform Resource Locator, единый указатель ресурса
ОС	Операционная система
ПК	Персональный компьютер
ПО	Программное обеспечение

1 ОБЩИЕ СВЕДЕНИЯ

1.1 Назначение

Настоящая инструкция описывает интеграцию Системы редактирования и совместной работы, Системы хранения данных и внешней SSO-системы.

Внешняя аутентификация (SSO) позволяет использовать корпоративные учетные данные для выполнения единого входа в систему. Использование внешней аутентификации обеспечивает расширенное управление безопасностью учетных записей и помогает интегрировать продукт с существующими продуктами.

1.2 О компонентах продукта

Система хранения данных — компонент, предназначенный для безопасного хранения корпоративных файлов и обеспечения возможностей авторизации, аутентификации и разграничения прав доступа пользователей.

Система редактирования и совместной работы — компонент, предназначенный для индивидуального и совместного редактирования текстовых и табличных документов, а также просмотра и демонстрации презентаций.

Представленные компоненты входят в состав следующих продуктов:

- «МойОфис Документы Онлайн»;
- «МойОфис Профессиональный 2»;
- «МойОфис Профессиональный 3»;
- «МойОфис Схема»;
- Squadus PRO.

Подробное описание возможностей продукта приведено в документе «Функциональные возможности».

1.3 Архитектура решения

На рисунке 1 показан процесс взаимодействия при обращении пользователя в систему с интеграцией внешней SSO.

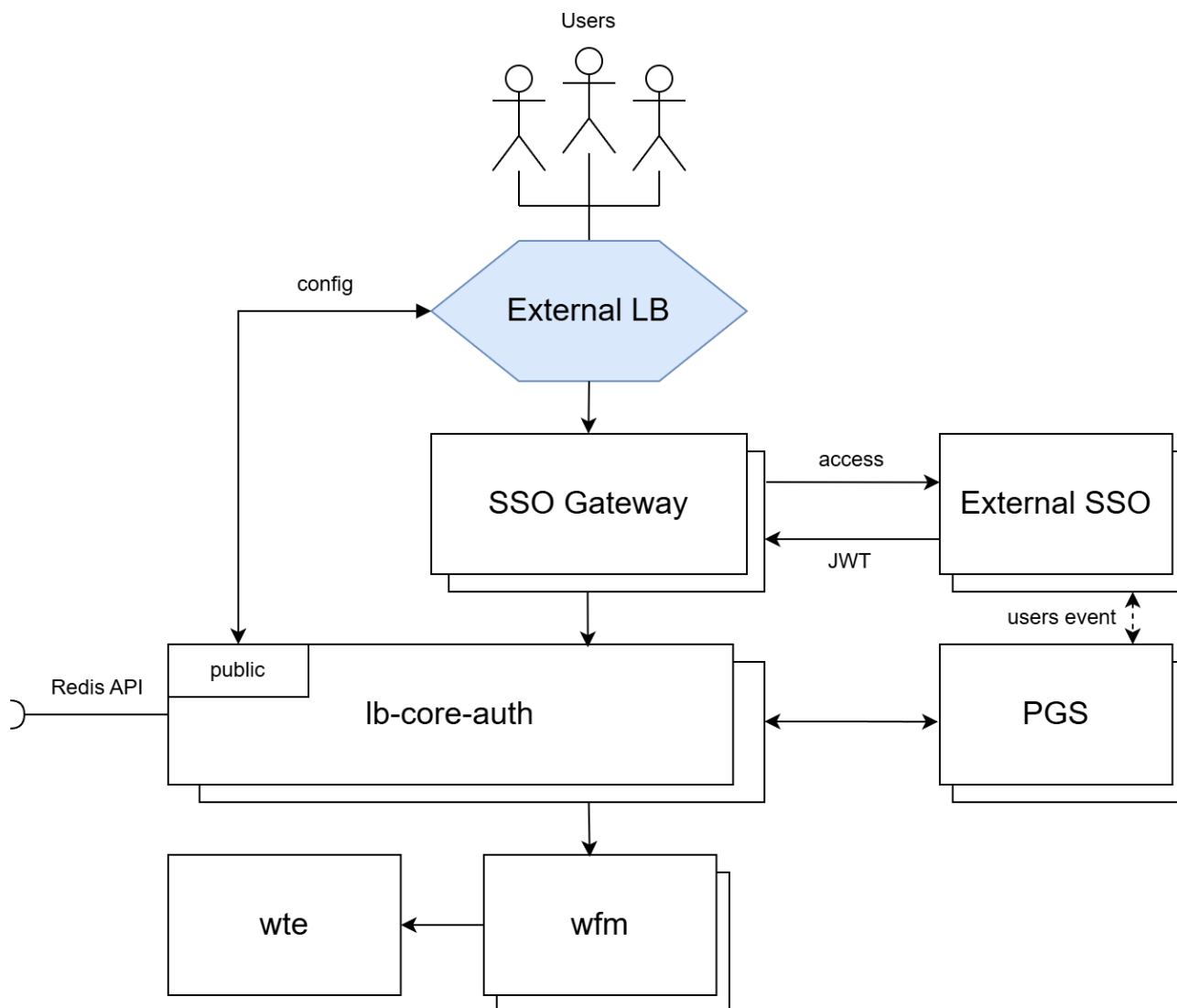


Рисунок 1 — Архитектурная схема взаимодействия с внешним SSO

Таблица 2 — Перечень компонентов

	Наименование сервиса	Описание
Система редактирования и совместной работы	lb-core-auth	Расширенный веб-сервер NGINX с поддержкой Lua. Отвечает за авторизацию, доступность API, балансировку
	SSO Gateway	SSO-Шлюз, выполняет следующие функции: 1. Аутентификация и авторизация запросов перед их проксированием к продуктовым сервисам (валидация и интроспекция токена). 2. Извлечение информации о пользователе из токена и проксирование этой информации в заголовках к продуктовым сервисам. 3. Контроль за WebSocket соединениями (в случае отсутствия сессии у пользователя во внешнем SSO произойдет разрыв соединения со стороны gateway)
	WFM	Web file manager (SPA веб-приложение). Веб-клиент файлового менеджера (Виджет вложений). Отвечает за операции с файлами (кроме редактирования) и просмотр файлов по публичным ссылкам
	WTE	Web text editors (Веб-клиент офисного приложения). Отвечает за редактирование и чтение файлов
Система хранения данных	PGS	Серверный компонент, обеспечивающий хранение объектов пользователя, хранение общих корпоративных объектов, выполнение файловых операций для пользователя
Внешние системы	External LB	Внешний балансировщик
	External SSO	Внешняя SSO-система

2 НАСТРОЙКА ИНТЕГРАЦИИ

2.1 Настройка внешнего SSO (на примере KeyCloak)

2.1.1 Создание OpenID Connect Client

1. В боковой панели нажимаем **Clients** → **Create client**.

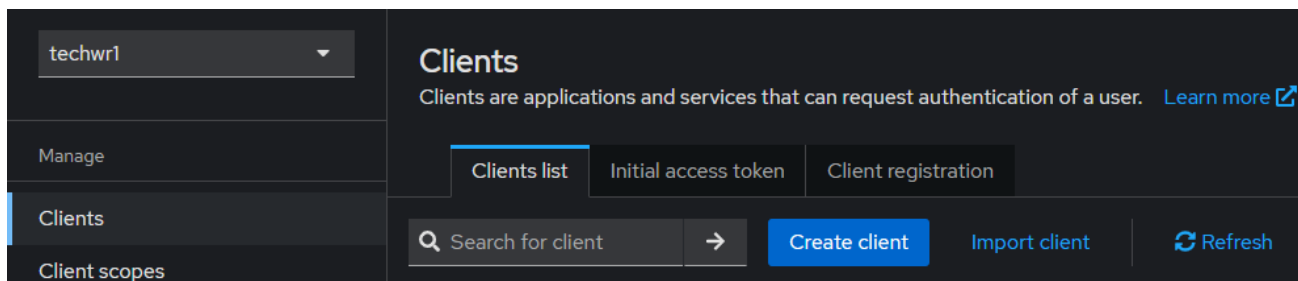


Рисунок 2 — Создание клиента

2. Настраиваем раздел **General settings**

- Выпадающее меню поля Client type должно быть выставлено в **OpenID Connect**
- Поле **Client ID** заполняется новым ID.
- Поля **Name** и **Description** заполняются при необходимости

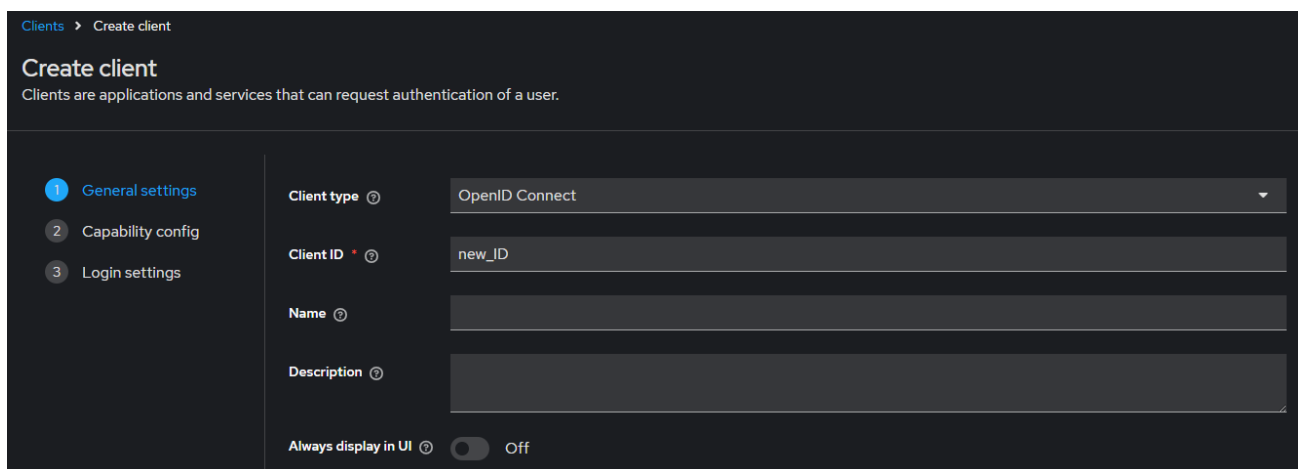


Рисунок 3 — Создание клиента. Раздел **General settings**

3. Настраиваем раздел **Capability Config**

- Устанавливаем пункты Client authentication и Authorization в положение  **On**
- Authentication flow оставляем по умолчанию включенными Standard flow и Direct access grants

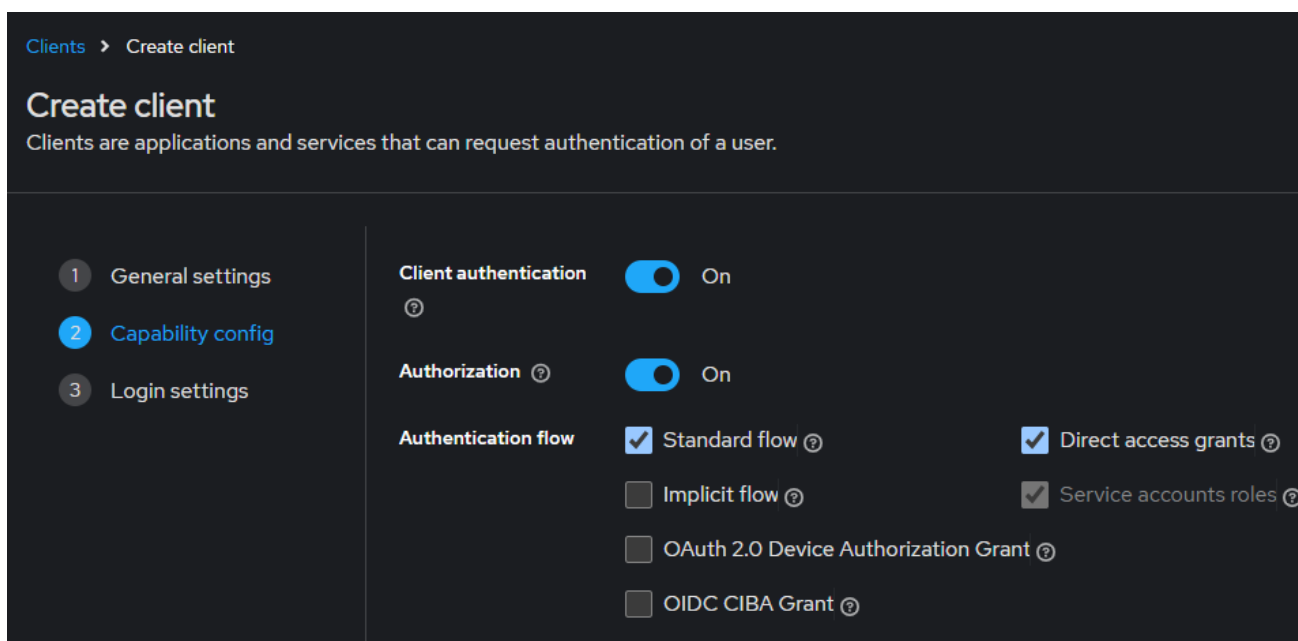


Рисунок 4 — Создание клиента. Раздел **Capability Config**

4. Настраиваем **Login settings**

В поле **Valid redirect URIs** указываем шаблон URI на gateway SSO (пример: https://gw-sso.example.net:8281/*)

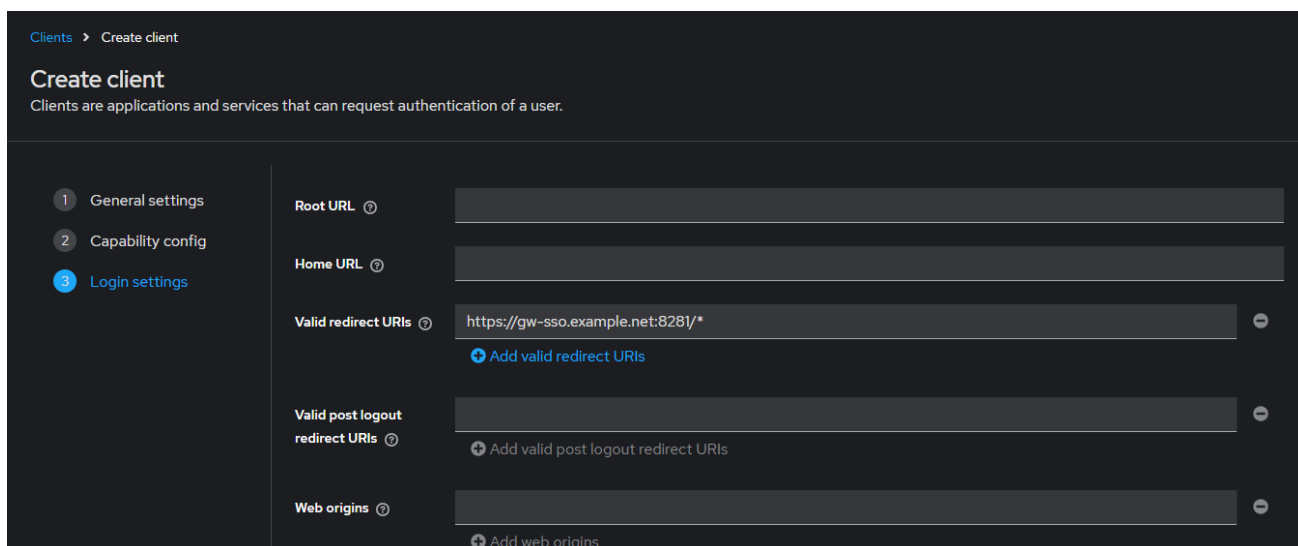


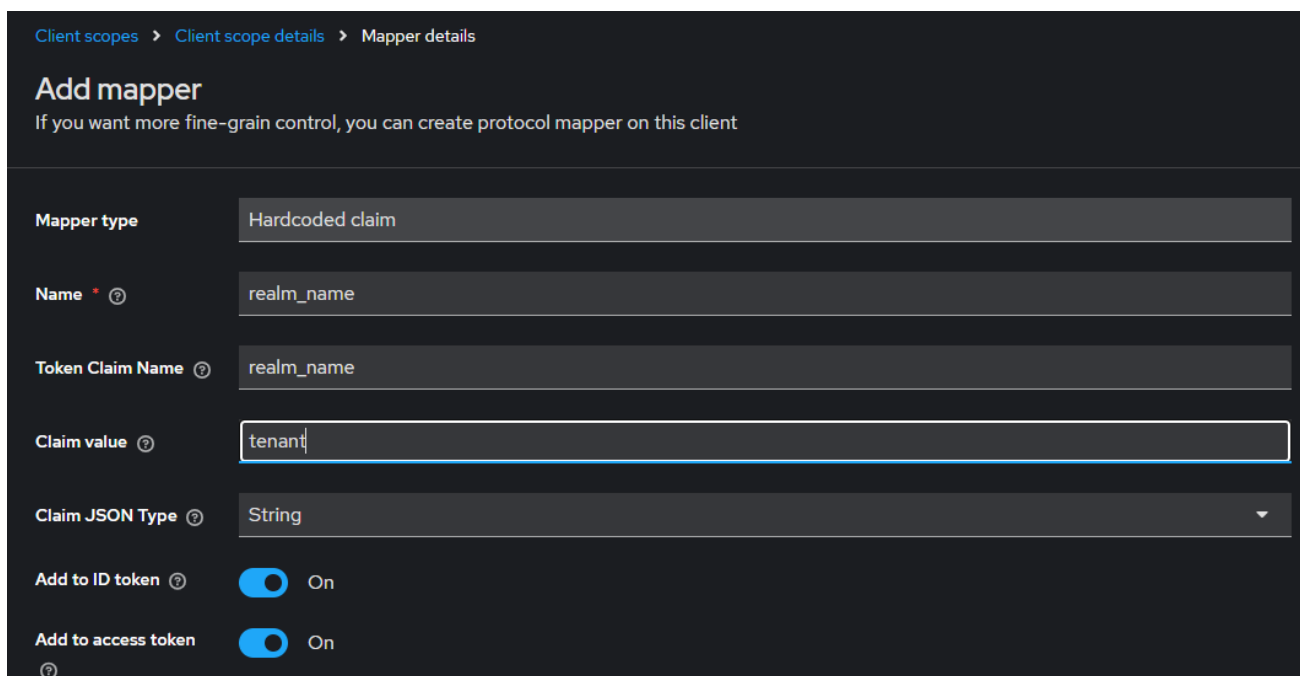
Рисунок 5 — Создание клиента. Раздел **Login settings**

После создания OpenID Connect Client необходимо использовать значения Client ID (из п.2) и Client Secret (во вкладке Credentials в панели редактирования клиента) для конфигурирования сервиса authn (секция keycloak.client).

2.1.2 Настройка access токена

После создания OpenID Connect Client необходимо добавить атрибут `realm_name` в JWT access токен. Для этого, в панели редактирования клиента:

1. Нажимаем **Client Scopes** → **<client_id>-dedicated** (например admin-dedicated).
2. Во вкладке Mappers нажимаем **Add mapper** → **By configuration**.
3. Из списка выбираем **Hardcoded claim**.
4. Заполняем поля **Name** и **Token Claim Name** значением `realm_name`.
5. Заполняем поле **Claim value** именем текущего тенанта.
6. **Claim JSON Type** оставляем **String** по умолчанию.
7. Убедиться, что переключатель  **On** напротив Add to access token, остальные checkboxes можно оставить включенными по умолчанию.



Client scopes > Client scope details > Mapper details

Add mapper

If you want more fine-grain control, you can create protocol mapper on this client

Mapper type	Hardcoded claim
Name *	realm_name
Token Claim Name	realm_name
Claim value	tenant
Claim JSON Type	String
Add to ID token	☒ On
Add to access token	☒ On

Рисунок 6 — Меню настройки `realm_name`

2.1.3 Назначение роли администратора

Для назначения пользователю роли администратора необходимо:

1. Выбрать пункт **Realm roles** в боковом меню и нажать **Create role**.
2. В поле **Role name** указать имя pgs-admin и нажать **Save**.

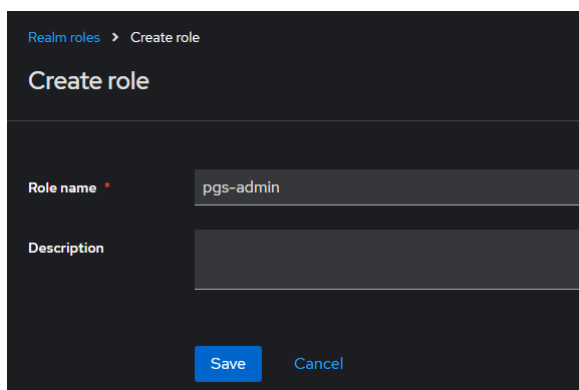


Рисунок 7 — Создание роли

3. Выбрать пункт **Users** в боковом меню и в открывшемся списке выбрать пользователя для назначения роли.
4. Открыть карточку пользователя и выбрать вкладку **Role mapping**.

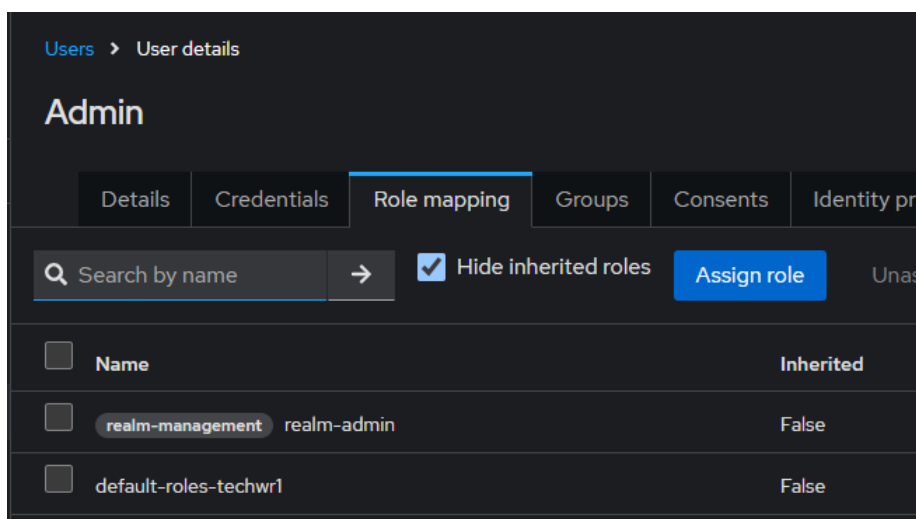


Рисунок 8 — Вкладка **Role mapping** перед добавлением роли

5. Нажать кнопку **Assign role**, в открывшемся окне переключить фильтр со значения **Filter by Client** на значение **Filter by realm roles**.

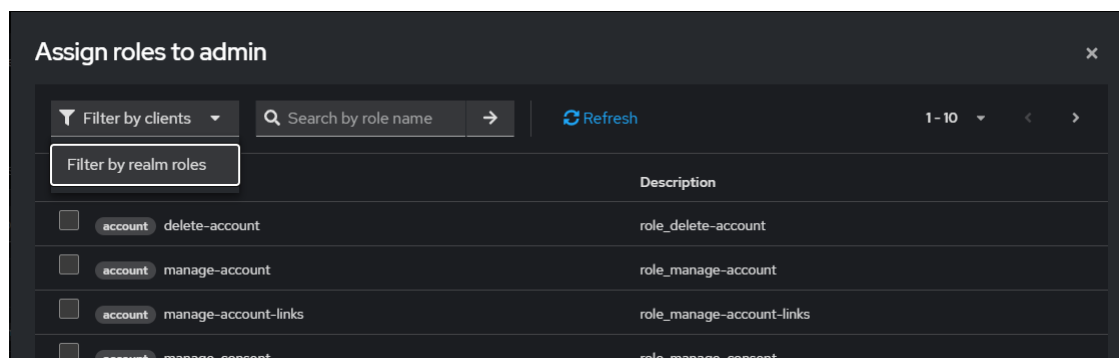


Рисунок 9 — Список ролей

6. Выбрать роль pgs-admin и нажать **Assign**.

7. После добавления роли pgs-admin пользователю вкладка **Role mapping** будет содержать добавленную роль.

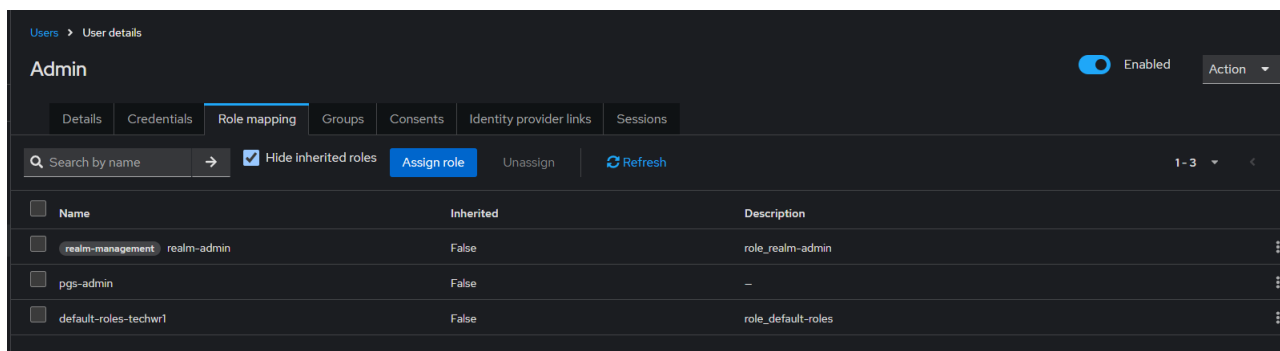


Рисунок 10 — Вкладка **Role mapping** после добавления роли

2.1.4 Конфигурация времени жизни токенов / сессии

1. В боковой панели нажимаем **Realm settings** → **Sessions**

2. В разделе **SSO Session Settings** необходимо настроить:

- SSO Session Idle (является временем жизни refresh токена) рекомендуется выставить не менее 3 дней
- SSO Session Max рекомендуется выставить не менее 10 дней

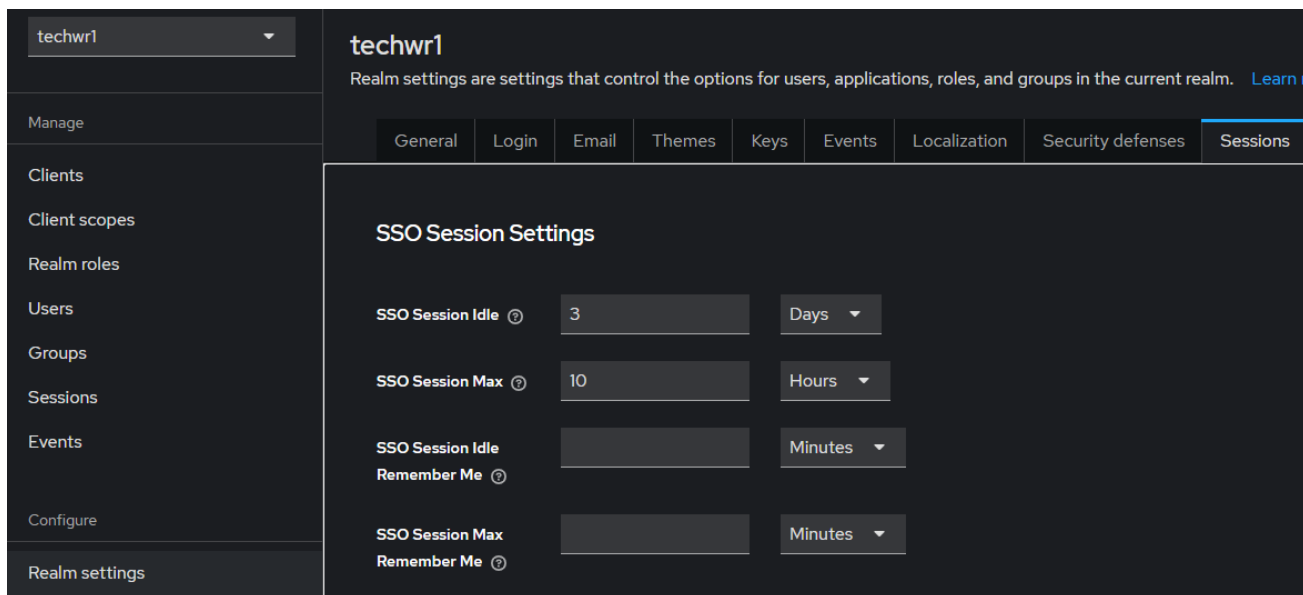


Рисунок 11 — Вкладка **Session** раздела **Realm settings**

3. В боковой панели нажимаем **Realm settings** → **Tokens**
4. В разделе **Access tokens** рекомендуется установить значение для поля **Access Token Lifespan** не менее 30 минут.

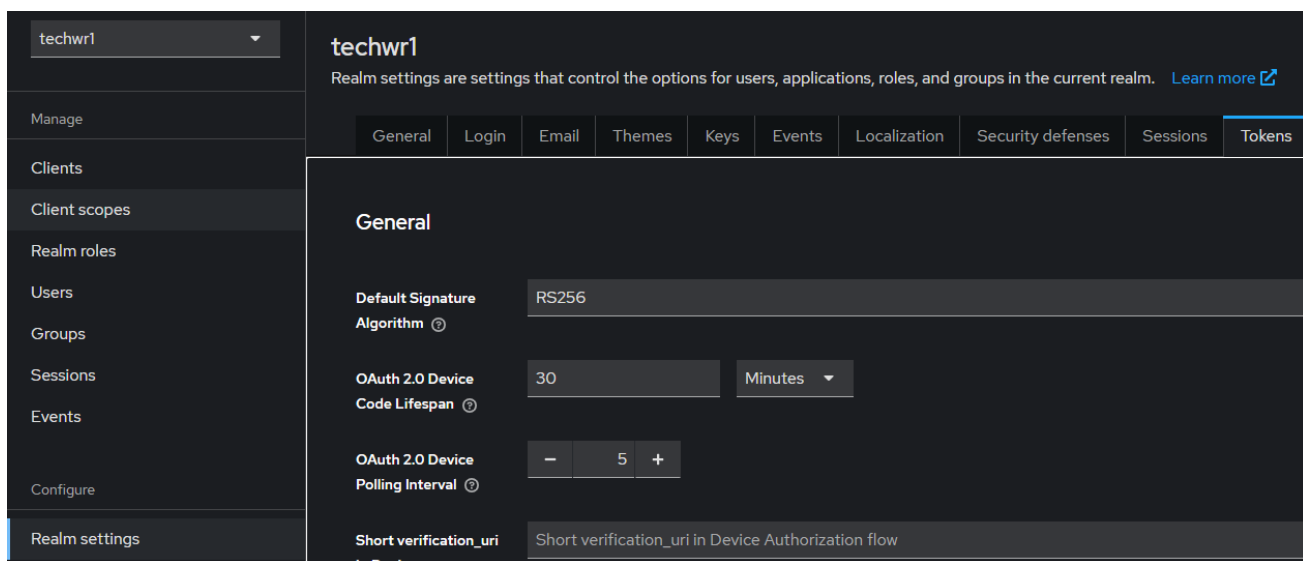


Рисунок 12 — Вкладка **Tokens** раздела **Realm settings**

2.1.5 Конфигурация для сбора событий об изменении / удалении пользователей

Для включения в KeyCloak регистрации и сохранения событий необходимо выполнить следующие действия:

1. В боковой панели нажать **Realm Settings** → **Events**
2. Нажимаем на нижнюю вкладку **Admin events settings**
 - Устанавливаем переключатель  On для **Save events** и **Include representation**;
 - В поле **Expiration** рекомендуется установить значение в несколько дней (2-4 дня);
 - Нажимаем Save.

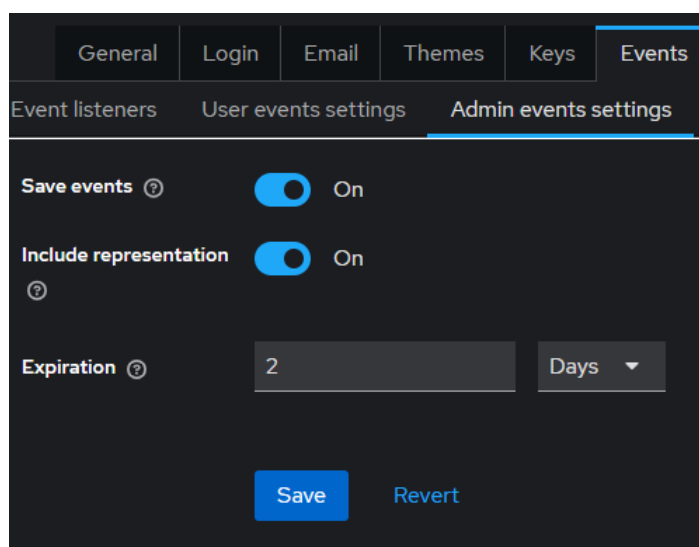


Рисунок 13 — Вкладка Admin events settings

3. Опционально нажимаем на нижнюю вкладку **User events settings** (сбор обычных пользовательских событий пока не используется).
 - Выставляем checkbox в true для **Save events**;
 - **Expiration** рекомендуется выставить в несколько дней (2-4 дня);
 - Нажимаем Save.

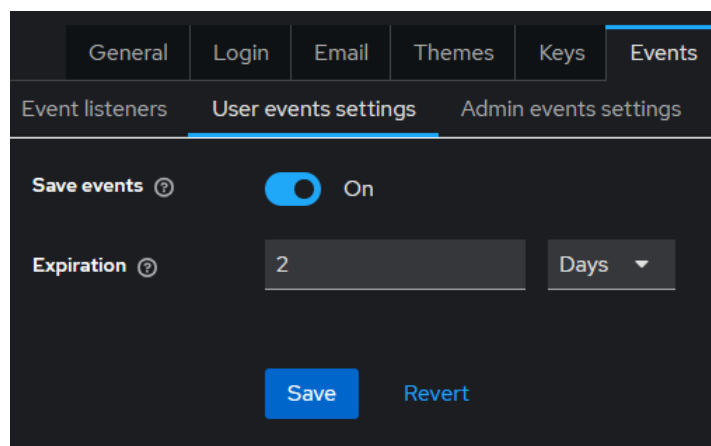


Рисунок 14 — Вкладка User events settings

4. Создаем и настраиваем пользователя для получения списка событий:

- В боковой панели нажимаем Users → Add user
- В General секции вводим Username, например, event_viewer
- Остальные поля заполняются опционально, нажимаем Create

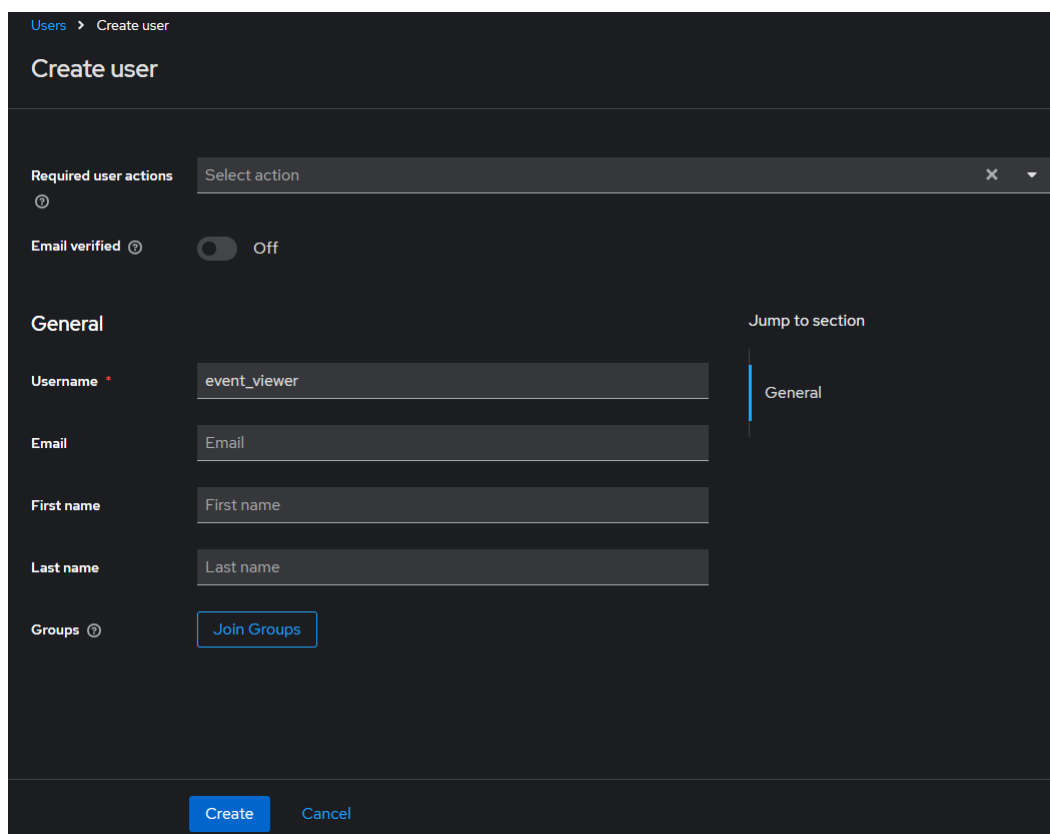


Рисунок 15 — Окно создания пользователя

5. После создания пользователя остаемся в панели его редактирования:

- Во вкладке Credentials устанавливаем пароль, нажав на Set password (checkbox Temporary выставляем в Off)

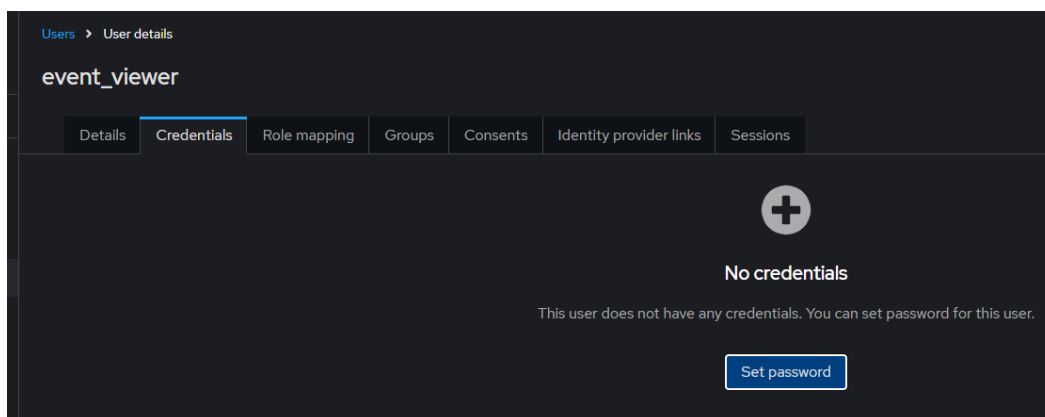


Рисунок 16 — Вкладка Credentials

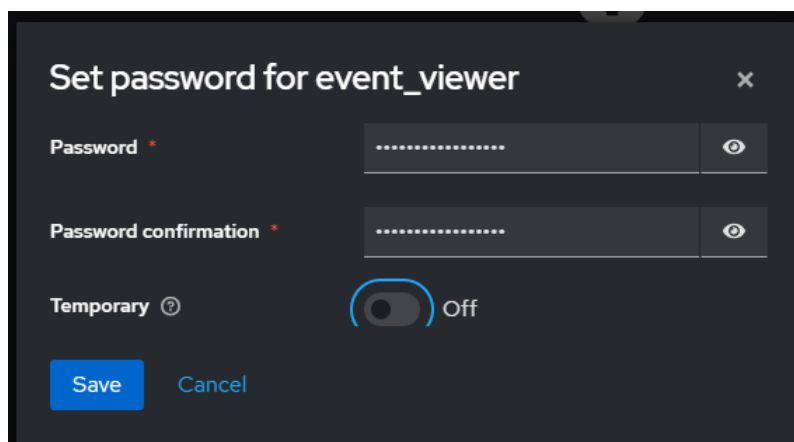


Рисунок 17 — Окно установки пароля

- Во вкладке Role mapping назначаем роль пользователю для просмотра событий (Assign Role → в списке находим и выбираем роль [realm-management]view-events → Assign)

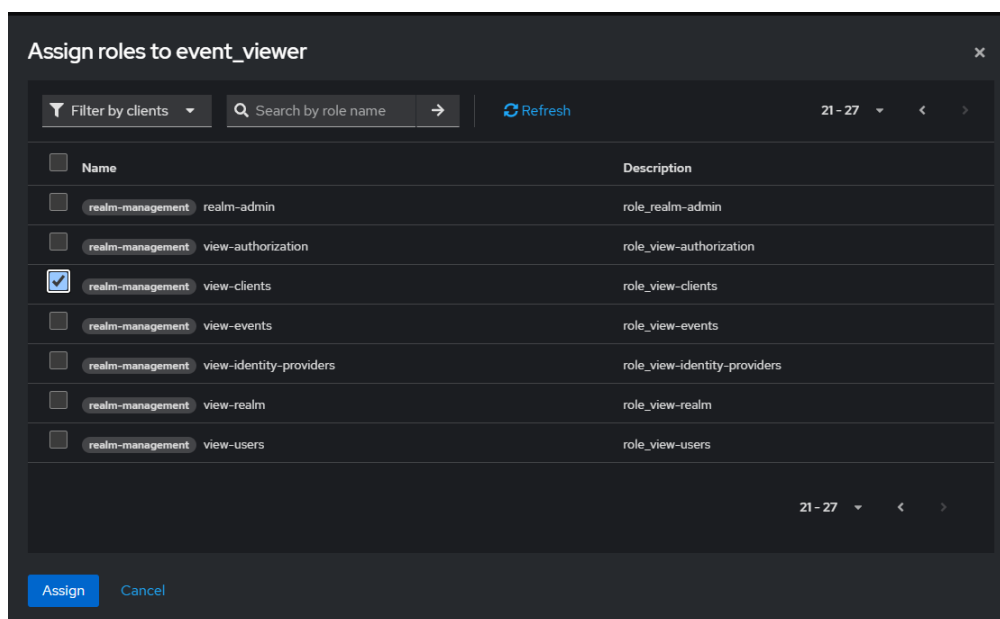


Рисунок 18 — Список ролей

2.2 Настройка компонента PGS

Для подключения внешнего SSO перед началом установки внутренних сервисов необходимо открыть с помощью текстового редактора файл в `hosts.yml` и добавить переменные, представленные в таблице 3.

Таблица 3 — Переменные для настройки внешнего SSO

Имя переменной	Значение по умолчанию	Тип переменной	Описание
<code>redis_sentinel</code>	<code>true</code>	<code>bool</code>	Всегда по-умолчанию <code>true</code>
<code>redis_host</code>	<code>""</code>	<code>string / array</code>	Значение необходимо взять из файла CO inventory из группы хостов <code>co_imc</code>. Для standalone установки переменная принимает тип <code>string</code> <code>redis_host: "co-infra-1.installation.example.net"</code> Для кластерной установки переменная принимает тип <code>array</code>. Формат записи массива хостов: <code>["host1", "host2", "host3"]</code> Пример значения: <code>redis_host: ["co-infra-1.installation.example.net;co-infra-2.installation.example.net;co-infra-3.installation.example.net"]</code> Перечисляемые хосты в значении должны быть через знак <code>;</code> без пробелов
<code>redis_password</code>	<code>""</code>	<code>string</code>	Пароль для подключения к сервису Redis на стороне CO. Значение переменной должно совпадать с аналогичной переменной в CO
<code>redis_sentinel_port</code>	<code>26379</code>	<code>string</code>	Порт для подключения к сервису Redis на стороне CO. Значение переменной должно совпадать с аналогичной переменной в CO
<code>common_redis_sentinel_master_name</code>	<code>6379</code>	<code>string</code>	Имя мастер-узла в кластере Redis. Значение по умолчанию <code>6379</code> , должно совпадать с аналогичной переменной в CO

Пример заполнения переменных при установке типа standalone:

```
co:
  coapiurl: "https://co-api-host:8443/"
  redis_sentinel: "true"
  redis_host: "co-infra-1.installation.example.net" # Group host "co_imc" from CO
inventory
  redis_password: "" # Redis password from CO group_vars
  redis_sentinel_port: "26379"
  common_redis_sentinel_master_name: "6379"
```

Пример заполнения переменных при кластерной установке:

```
co:
  coapiurl: "https://co-api-host:8443/" # CO auth-nodes load balancer
  co_lb: false
  vip_auth: "co-vip-ip"
  lb_keepalived_pass: "your_strong_password"
  redis_sentinel: "true"
  redis_host: "co-infra-1.installation.example.net;co-infra-
2.installation.example.net;co-infra-3.installation.example.net" # Group host
"co_imc" from CO inventory
  redis_password: "" # Redis password from CO group_vars
  redis_sentinel_port: "26379"
  common_redis_sentinel_master_name: "6379"
```

Если значение переменной `redis_sentinel` указано `false`, то сервис разворачивается, но не запускается. При проверке после установки сервис будет представлен со значением 0/0.

`pgs_users 0/0`

Для включения сервиса необходимо в файле `/opt/Pythagoras/pgs-stack.yml` указать переменные в секции `users`.

Пример заполнения переменных:

```
users:
  deploy:
    replicas: 0 # Устанавливаем значение 1
    restart_policy:
      condition: any
    placement:
      max_replicas_per_node: 1
      constraints:
        - node.labels.pythagoras == true
  environment:
    - EUCLID_BASE_URL=http://euclid:8852
    - REDIS_SENTINEL_ENABLED=True
    - REDIS_SENTINEL_MASTER_NAME=6379 # Описание см. в таблице 3
    - REDIS_PASSWORD= # Описание см. в таблице 3
    - REDIS_HOST= # Описание см. в таблице 3
    - REDIS_PORT=26379 # Описание см. в таблице 3
  image: pgs-private-registry:5001/pythagoras/users-service:0.1.6-71
  volumes:
    - type: bind
      source: /var/run/docker.sock
      target: /var/run/docker.sock
  networks:
    - pgs-network
```

```
logging:
  driver: syslog
  options:
    syslog-address: udp://172.17.0.1:514
    tag: users
```

После заполнения конфигурационного файла необходимо выполнить команду:

```
docker service scale pgs_users=1
```

2.3 Настройка компонента СО

2.3.1 Подготовка конфигурационных файлов

Перед началом установки внутренних сервисов для подключения внешнего SSO необходимо с помощью текстового редактора открыть файл в `group_vars/main.yml` и добавить переменные, представленные в таблице.

Таблица 4 — Переменные для подключения внешнего SSO

Наименование переменной	Значение переменной по умолчанию	Тип переменной	Описание
<code>common_sso_enabled</code>	<code>false</code>	<code>boolean</code>	Включение поддержки внешнего SSO (принимает значения <code>true</code> и <code>false</code>)
<code>authn_keycloak_listen_endpoint</code>	<code>""</code>	<code>string</code>	Полное доменное имя внешнего Keycloak (и <code>":порт"</code> - если внешний keycloak доступен по порту, отличному от 443)
<code>authn_keycloak_realm</code>	<code>""</code>	<code>string</code>	Имя realm во внешнем Keycloak
<code>authn_client_id</code>	<code>""</code>	<code>string</code>	<code>Client_id</code> OpenID Connect клиента во внешнем Keycloak
<code>authn_client_secret</code>	<code>""</code>	<code>string</code>	<code>Client_secret</code> OpenID Connect клиента во внешнем Keycloak
<code>authn_admin_username</code>	<code>""</code>	<code>string</code>	Имя пользователя с доступом администратора к внешнему Keycloak
<code>authn_admin_password</code>	<code>""</code>	<code>string</code>	Пароль пользователя с доступом администратора к внешнему Keycloak
<code>gateway_identity_provider_issuer</code>	<code>""</code>	<code>string</code>	URL внешнего Keycloak "https://{ { authn_keycloak_listen_endpoint } }/realms/{ { authn_keycloak_realm } }"
<code>gateway_identity_provider_jwk_url</code>	<code>""</code>	<code>string</code>	URL на информацию о JWK во внешнем Keycloak "https://{ { authn_keycloak_listen_endpoint } }/realms/{ { authn_keycloak_realm } }/protocol/open id-connect/certs"

2.3.2 Подготовка DNS-записи

1. Создать внешнюю DNS-запись, которая будет указывать на один из хостов в группе `co_gateway: gw.<domain_name>` или `gw-<domain_env>.<domain_name>` в зависимости от выбранного способа формирования параметра `co_domain_module`.
2. Открыть порт 7443 для внешнего доступа к хосту, указанному в п. 2 раздела «Подготовка конфигурационных файлов».

2.3.3 Порядок установки сервиса

Для запуска установки сервисов для интеграции SSO, необходимо запустить следующую команду:

```
ansible-playbook playbooks/co.yml --diff --tags sso,openresty
```

Для полной переустановки системы, необходимо запустить следующую команду:

```
ansible-playbook playbooks/main.yml --diff
```

2.3.4 Проверка работы после установки

После завершения установки необходимо выполнить проверку корректности записи ключей, для этого необходимо:

1. Открыть с ETCD.
2. Открыть ветку `keys/nct/co/config/wfe` и проверить наличие следующих ключей:

```
authentication.sso.gateway.base.url: https://(gw.<domain_name>|gw-  
<domain_env>.<domain_name>):7443  
authentication.sso.gateway.debug: False  
authentication.sso.gateway.enabled: True  
authentication.sso.gateway.introspect.time.before.expired: 250
```

3. Открыть ветку `keys/nct/co/config/wfe/routing` и проверить наличие следующего ключа:

```
coapi.base.url: https://(gw.<domain_name>|gw-<domain_env>.<domain_name>):7443
```

2.4 Настройка авторизации

2.4.1 Добавление пользователя

Для добавления пользователя следует использовать etcd-браузер.

1. Открыть etcd-браузер используя следующую ссылку:

```
https://co-etcd.<DEFAULT_DOMAIN>:8001
```

где: `<DEFAULT_DOMAIN>` — домен используемый в продукте.

2. Открыть ветку `/keys/nct/co/config/wfe/oauth2_clients`

3. Для создания пользователя с именем `pgs` необходимо добавить запись в формате:

```
pgs: { "client_secret": "<client_secret>", "redirect_uri": "pgs" }
```

Переменные, используемые при создании пользователя описаны в таблице 5.

Таблица 5 — Переменные для создания пользователя

Имя переменной	Значение по умолчанию	Тип переменной	Описание
client_secret	-	string	Создаваемый пользователем ключ. Рекомендуется использовать стороннюю утилиту <code>pwgen</code>
redirect_uri	pgs	string	Возвращаемое значение для проверки авторизации

Для генерации паролей рекомендуется использовать утилиту `pwgen`. Безопасный пароль необходимо генерировать с помощью команды:

```
pwgen <длина пароля> 1
```

где: `<длина пароля>` — должна быть не меньше 20 символов.

2.4.2 Получение токена авторизации

Для получения токена авторизации необходимо выполнить следующую команду:

```
curl -L 'https://auth-sso.<DEFAULT_DOMAIN>/oauth2/srv/token' \  
-u pgs:<client_secret> \  
-d 'grant_type=client_credentials&redirect_uri=pgs&sub=pgs@default.local'
```

где:

— `<DEFAULT_DOMAIN>` — домен используемый в продукте;

— `<client_secret>` — ключ созданный пользователем и добавленный в etcd.

Пример ответа:

```
{
  "expires_in":3600,
  "refresh_token":"0d9bb08f9cb2b62842203dd6d5add3da",
  "access_token":"698715ae82fc19e561a4fe2a82f5c1ec"
}
```

где:

- expires_in — время жизни токена в секундах;
- refresh_token — токен обновления (не используется в текущей конфигурации);
- access_token — токен авторизации.

2.4.3 Использование токена в запросах

Пример запроса с использованием полученного токена авторизации

```
"access_token":"698715ae82fc19e561a4fe2a82f5c1ec"
curl -XGET 'https://admin-ss0.devoffice.ru/adminapi/tenants/default' \
-H 'x-sid: 698715ae82fc19e561a4fe2a82f5c1ec' -vvvs
```

3 ОГРАНИЧЕНИЯ

Ограничения текущей версии продукта при использовании интеграции.

3.1 Отображение пользователей в продукте

Синхронизация пользователей с внешней системой SSO происходит при авторизации (запрос `get_token_v2`) или при обновлении токена. Это означает, что новые пользователи появляются в системе только после первичной авторизации в Облаке. До этого момента они не отображаются в административном интерфейсе, диалогах шаринга и списке контактов в документах.

3.2 Управление тенантами

1. Необходимо создать внутренний тенант на уровне PGS, который будет синхронизирован с realm внешней SSO (Keycloak).
2. Название тенанта и имя realm должны совпадать.

3.3 Количество создаваемых тенантов

Поддерживается работа только с одним realm (тенантом) на стороне компонента SSO_Gateway.

При использовании интеграции с внешней SSO-системой, не рекомендуется создавать более 1 тенанта для корректной работы системы.

3.4 Изменение роли

Для мгновенного изменения прав доступа необходимо:

- Открыть веб-интерфес внешней SSO (Keycloak);
- Нажать в боковом меню пункт **Users** и выбрать из списка нужного пользователя;
- В учетной записи пользователя перейти в закладку **Sessions** и нажать кнопку **Logout all sessions**.

	Admin		
Manage	DetailsCredentialsRole mappingGroupsConsentsIdentity provider linksSessions		
Clients	Search session → Logout all sessions Refresh		
Client scopes			
Realm roles			
Users	Started Last access 3/13/2025, 11:38:21 AM3/13/2025, 11:38:22 AM		

Рисунок 19 — Порядок завершения активных сессий пользователя