

## Оглавление

<b>Руководство по полному развёртыванию сервисов «ЦПР» .....</b>	<b>2</b>
Предварительная настройка.....	2
Настройка Keuscloak .....	5
Настройка MiniO.....	14
Настройка сервисов .....	16

# Руководство по полному развёртыванию сервисов «ЦПР»

## Предварительная настройка

[Ссылка на репозиторий GitHub](#)

1. Загрузить последние обновления из [репозитория проекта](#) ветки **main**.
2. Перейти по пути: `.\CPR_services\`
3. В директории `CPR_services` находится файл ***config.yaml*** (см. листинг 1).

Листинг 1 – Файл *config.yaml*

```
services:
  keycloak-postgres:
    image: library/postgres:${KC_POSTGRES_IMAGE_TAG:-14}
    container_name: ${POSTGRES_CONTAINER_NAME:-postgres}
    restart: on-failure
    environment:
      POSTGRES_USER: postgres
      POSTGRES_PASSWORD: postgres
      POSTGRES_DB: postgres
    healthcheck:
      test: pg_isready -d postgres
      interval: 10s
      timeout: 5s
      retries: 3
      start_period: 5s
    ports:
      - ${KC_POSTGRES_PORT_MAPPING:-5433}:5432
    deploy:
      resources:
        limits:
          memory: 256M
    networks:
      - cpr_network

  keycloak:
    image: quay.io/keycloak/keycloak:20.0.2
    container_name: keycloak
```

```

command:
  - start --auto-build --db postgres --hostname-strict-https false --
hostname-strict false --proxy edge --http-enabled true --import-realm --spi-
user-profile-legacy-user-profile-read-only-attributes *_RES_ACCESS_MODE

environment:
  KC_DB_URL: jdbc:postgresql://keycloak-postgres:5432/postgres
  KC_DB_USERNAME: postgres
  KC_DB_PASSWORD: postgres
  KC_DB_SCHEMA: public
  KC_FEATURES: preview
  KEYCLOAK_ADMIN: admin
  KEYCLOAK_ADMIN_PASSWORD: admin

ports:
  - 8282:8080

depends_on:
  keycloak-postgres:
    condition: service_healthy

healthcheck:
  test: ["CMD", "curl", "-f", "http://0.0.0.0:8080/realms/master"]
  start_period: 10s
  interval: 30s
  retries: 3
  timeout: 5s

networks:
  - cpr_network

minio:
  image: minio/minio:latest
  container_name: minio
  environment:
    MINIO_ROOT_USER: ${MINIO_ACCESS_KEY:-UsEr_y8b-DSq-C2K-t32} $
    MINIO_ROOT_PASSWORD: ${MINIO_SECRET_KEY:-y8b-DSq-C2K-t32}
  command: server ~/minio --console-address :9090
  ports:
    - '9090:9090'

```

```

- '9000:9000'

volumes:
- minio-data:/minio

networks:
- cpr_network

volumes:
  minio-data:












networks:
  cpr_network:
    driver: bridge

```

Данный файл содержит образы Docker для keycloak, postgres и minio.

Для образа keycloak необходимо указать адрес базы данных KC\_DB\_URL (можно оставить по умолчанию).

4. Находясь в одной директории с файлом **config.yaml** выполнить команду:  
***docker compose -f config.yaml -p config-services up -d***
5. Дождаться, пока все контейнеры поднимутся.

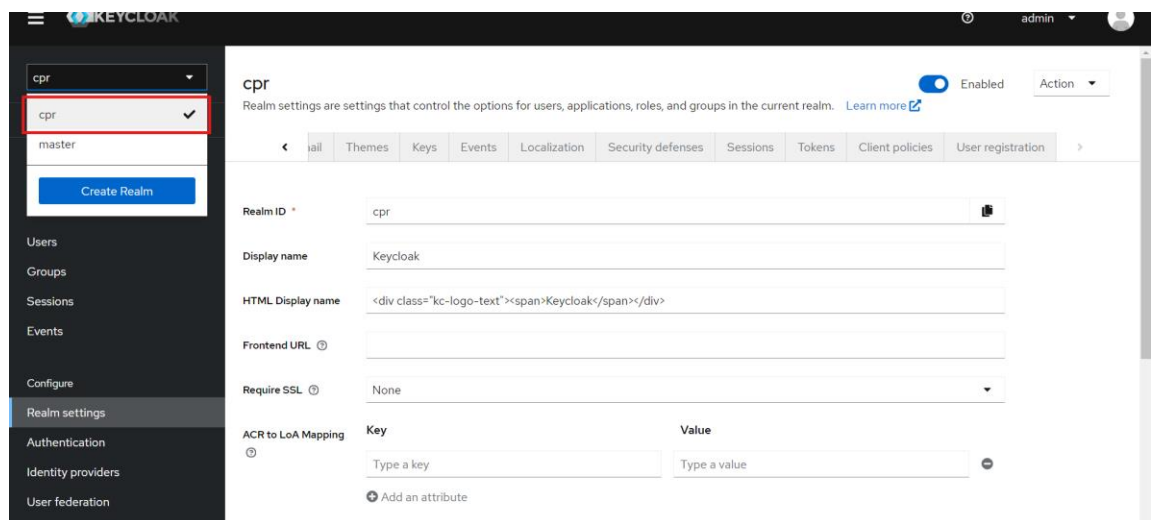
<input type="checkbox"/>		<b>config-se</b>	Running (3/3)	3.43%	9 minutes ago	<input type="checkbox"/>	:		
<input type="checkbox"/>		<b>minio</b> 1a0c21: <a href="#">minio/minio:late</a>	Running	<a href="#">9000:9000</a>  <a href="#">Show all ports (2)</a>	0.08%	9 minutes ago	<input type="checkbox"/>	:	
<input type="checkbox"/>		<b>keyclo</b> b0b27f: <a href="#">quay.io/keycloak</a>	Running	<a href="#">8282:8080</a> 	0.36%	9 minutes ago	<input type="checkbox"/>	:	
<input type="checkbox"/>		<b>postgr</b> 58fbda7: <a href="#">library/postgres:</a>	Running	<a href="#">5433:5432</a> 	2.99%	9 minutes ago	<input type="checkbox"/>	:	

# Настройка Keycloak

1. Перейти по пути `http://localhost:8282/` и выбрать Administration Console

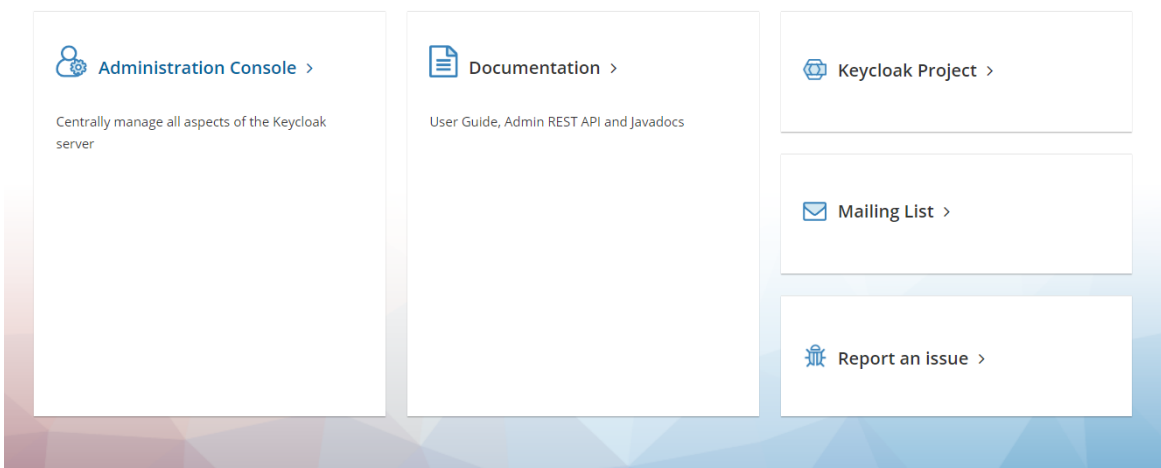
*Настройки Realm должны быть импортированы при запуске контейнера, но нужно убедиться, что всё соответствует настройкам, приведённым ниже.*

**При успешном импорте настроек необходимо в верхнем левом углу переключиться на realm CPR.**

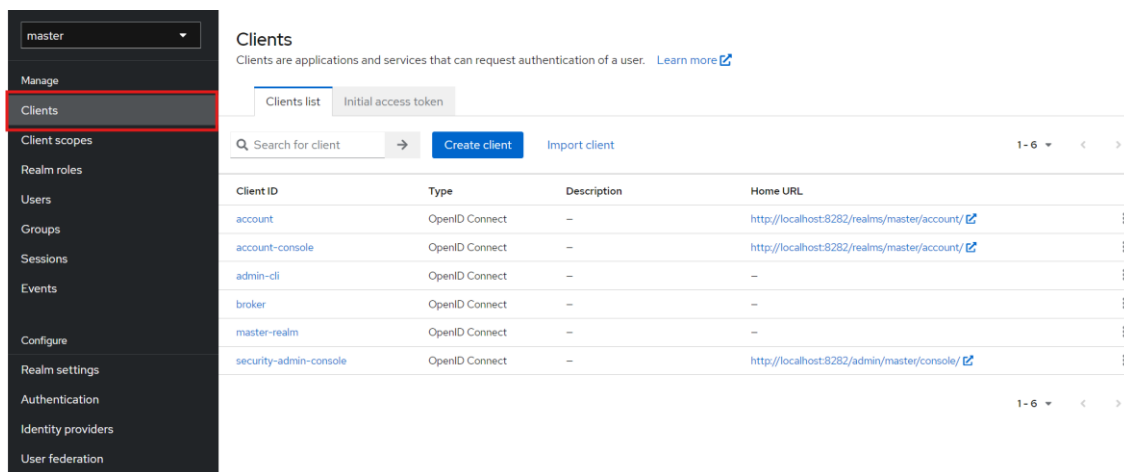


*Ниже приведена полная настройка Keycloak, если все настройки были успешно импортированы, необходимо только изменить клиентские ключи (см. п.13 и п.18). Также может потребоваться добавить пользователей (см. п. 16)*

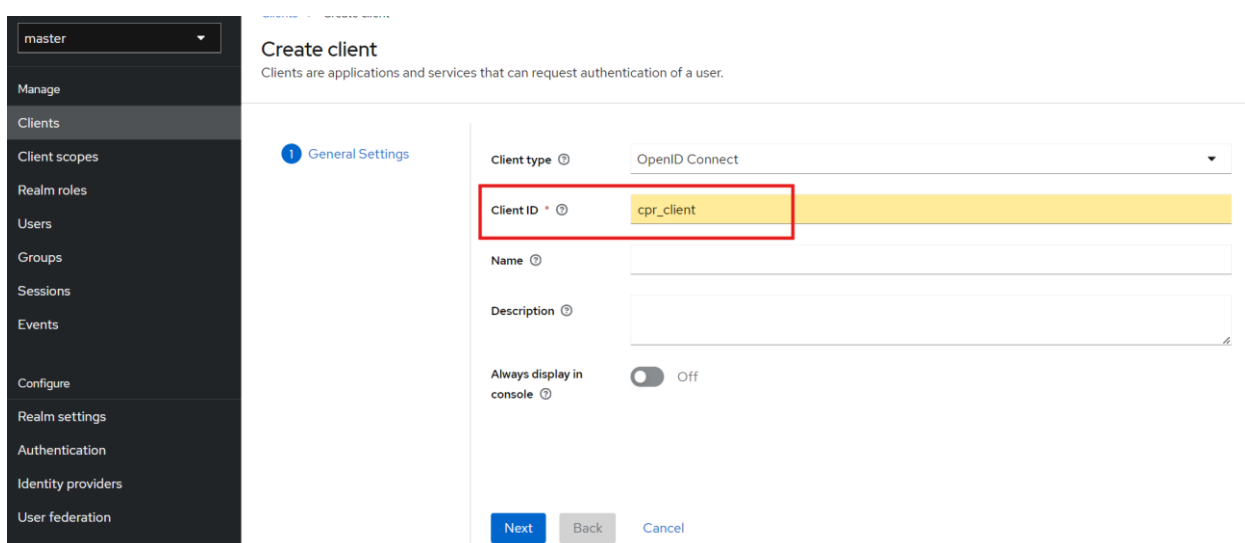
Welcome to **Keycloak**



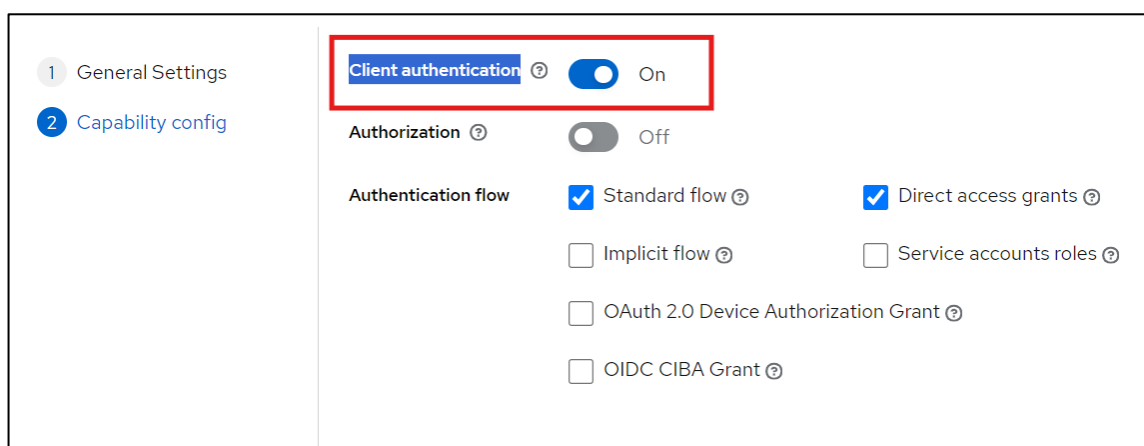
2. В открывшемся окне аутентификации ввести: **admin** для логина и пароля (если не было изменено в файле `config.yaml`) и нажать кнопку «**Sign in**».
3. На открывшейся странице в меню слева выбрать вкладку «**Clients**».



4. Нажать кнопку «Create client»
5. В поле «Client id» ввести: «cpr-client»



6. Нажать кнопку «Next». Перевести переключатель «Client authentication» в положение «on».



7. Нажать кнопку «Save».
8. В открывшемся окне выбрать вкладку «Credentials».

Client Created successfully

Clients > Client details

**cpr-client** OpenID Connect Enabled

Clients are applications and services that can request authentication of a user.

Settings Keys **Credentials** Roles Client scopes Sessions Permissions Advanced

Client Authenticator Client Id and Secret

Save

Client secret .....

Regenerate

9. Скопировать ключ из поля «Client secret».

В директории с проектом CPR\_services открыть файл **docker-compose.yml** и в списке сервисов найти **gateway** (API шлюз приложения).

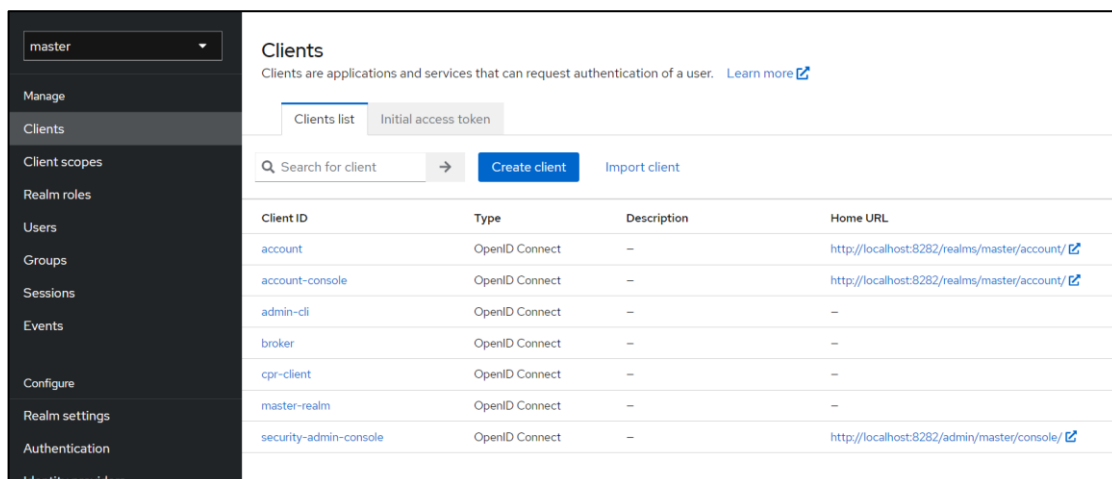
Отредактировать переменные среды:

- KEYCLOAK\_CLIENT\_SECRET=5e8qfIVapUsPqvmk42I7gfwohZTDZmrO
- KEYCLOAK\_CLIENT\_ID=cpr-client

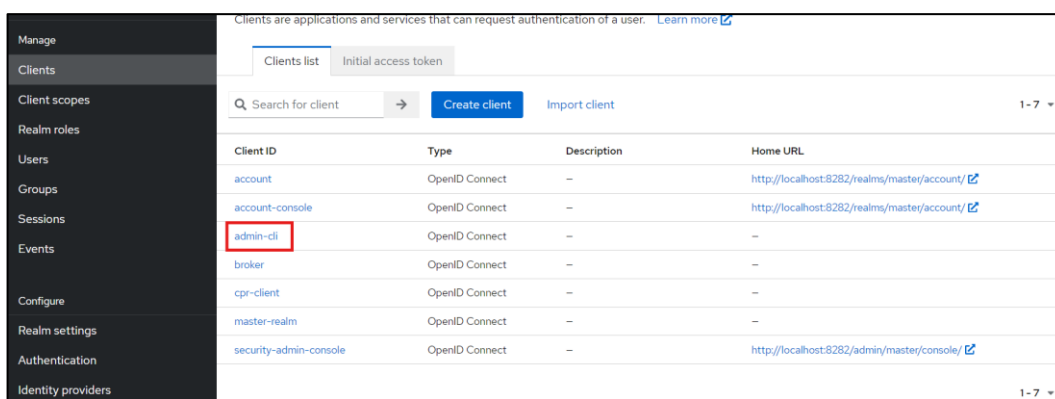
Заменить скопированным ключом значение переменной **KEYCLOAK\_CLIENT\_SECRET**.

```
services:
  gateway:
    build:
      context: .
      dockerfile: api-gateway/Dockerfile
    image: uzorovkirill/api-gateway:latest
    environment:
      - HOST_IP=localhost
      - DOCUMENT_SERVICE_PORT=83
      - TEXT_CORRECTION_SERVICE_PORT=84
      - KEYCLOAK_SERVER=http://127.0.0.1:8282
      - KEYCLOAK_CLIENT_SECRET=5e8qfIVapUsPqvmk42I7gfwohZTDZmrO
      - KEYCLOAK_CLIENT_ID=cpr-client
      - KEYCLOAK_ADMIN_ID=admin-cli
      - KEYCLOAK_ADMIN_SECRET=7ILwD8oKlOcRf0j6Bg4CJqtm4o6ksLED
      - KEYCLOAK_ADMIN_USERNAME=admin
      - KEYCLOAK_ADMIN_PASSWORD=admin
```

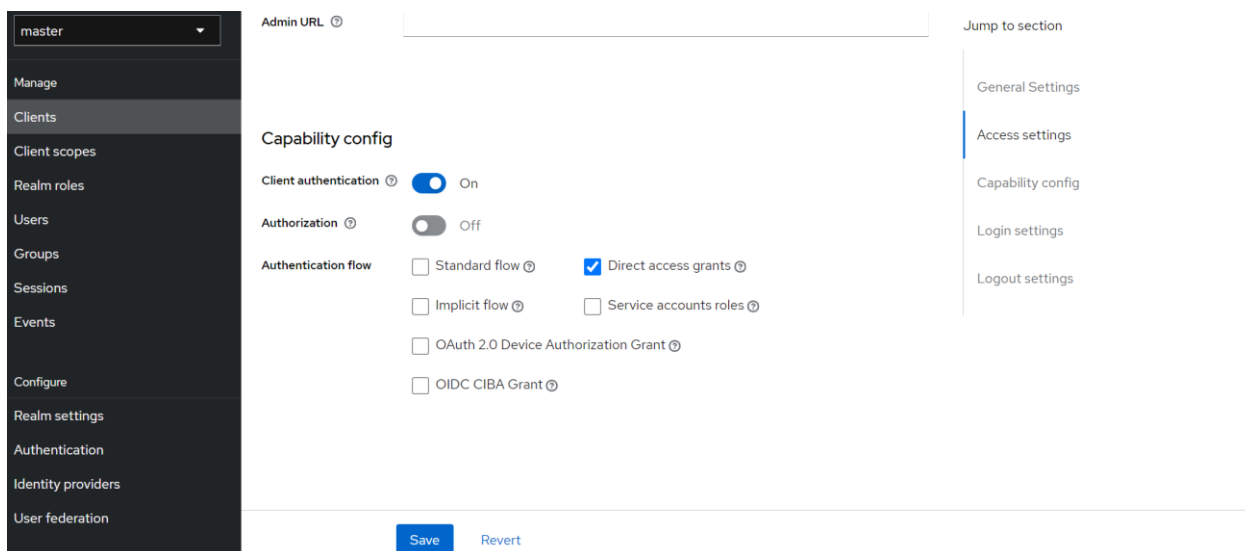
10. В панели администратора Keycloak выбрать в меню вкладку «**Clients**»



## 11. Выбрать клиента **admin-cli**



## 12. Промотать страницу вниз до «**Capability config**» и перевести переключатель «**Client authentication**» в положение «**ON**».



Нажать кнопку «**Save**».

## 13. Вернуться вверх страницы и выбрать появившуюся вкладку **Credentials**.



admin-cli OpenID Connect

Enabled ⓘ Action ▾

Clients are applications and services that can request authentication of a user.

Settings Keys **Credentials** Roles Client scopes Sessions Permissions Advanced

Client Authenticator ⓘ Client Id and Secret ▾

Save

Client secret ..... ⓘ ⓘ Regenerate

14. Аналогично шагам выше, скопировать значение ключа из поля «Client secret», в директории CPR\_services открыть файл **docker-compose.yaml** и в списке сервисов найти **gateway**.

Отредактировать переменную среды, заменив текущее значение, скопированным ключом:

- KEYCLOAK\_ADMIN\_SECRET=7ILwD8oKlOcRf0j6Bg4CJqtm4o6ksLEd

15. В панели администратора Keycloak выбрать в меню вкладку «**Realm roles**» и нажать кнопку «**Create role**»

master ▾

Manage

Clients

Client scopes

**Realm roles**

Users

Groups

Sessions

Events

realm roles > Create role

Create role

Role name \* supervisor

Description

Save Cancel

Ввести: «**supervisor**» и нажать кнопку «**Save**».

Аналогично добавить роль «**employee**».

Manage

Clients

Client scopes

**Realm roles**

Users

Groups

Sessions

Events

Configure

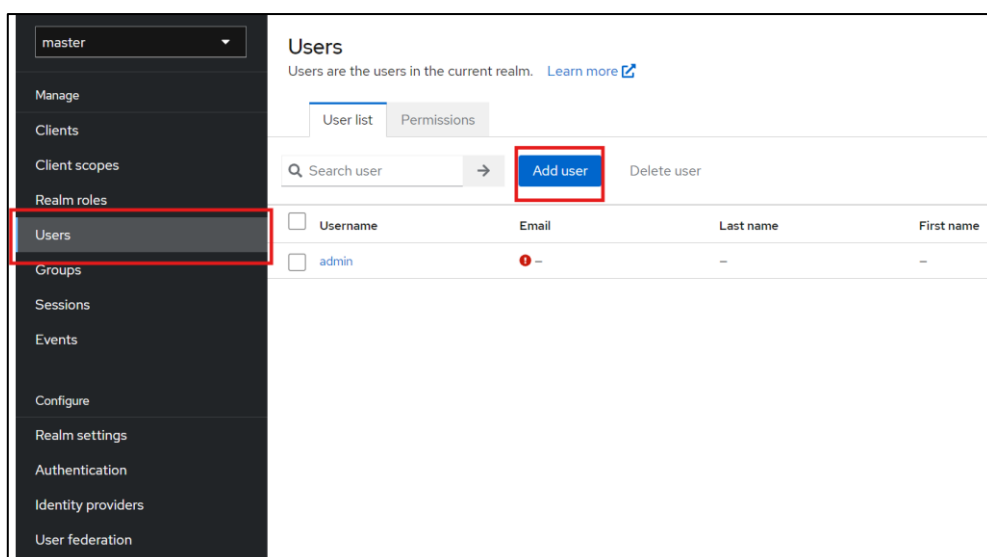
Realm settings

Authentication

Search role by name → Create role 1-7 < >

Role name	Composite	Description
admin	True	`\${role_admin}`
create-realm	False	`\${role_create-realm}`
default-roles-cpr ⓘ	True	`\${role_default-roles}`
employee	False	—
offline_access	False	`\${role_offline-access}`
supervisor	False	—
uma_authorization	False	`\${role_uma_authorization}`

16. В панели администратора Keycloak выбрать в меню вкладку «Users» и нажать кнопку «Add user»



17. Заполнить все поля как показано на рисунке ниже:

The screenshot shows the 'Create user' form in Keycloak. The left sidebar is the same as in the previous image, with 'Users' highlighted. The main area is titled 'Create user' and has a toggle switch labeled 'Enabled' which is turned on. The form contains the following fields and values: 'Username' with 'ivanova.t', 'Email' with 'ivanova@mail.ru', 'Email verified' with a toggle switch set to 'On', 'First name' with 'Таисия', and 'Last name' with 'Иванова'. There is a 'Required user actions' dropdown menu set to 'Select action' and a 'Groups' section with a 'Join Groups' button. At the bottom, there are 'Create' and 'Cancel' buttons.


18. Нажать кнопку «Create».

19. В открывшемся окне перейти на вкладку «Credentials».

Users > User details

ivanova.t Enabled Action

Details Attributes **Credentials** Role mapping Groups Consents Identity provider links Sessions



No credentials

This user does not have any credentials. You can set password for this user.


[Set password](#)


[Credential Reset](#)


20. Нажать на кнопку «**Set password**»

21. В открывшемся окне ввести «123» и убрать переключатель «**Temporary**». Нажать кнопку «**Save**».

Set password for ivanova.t

Password \* 123 

Password confirmation \* 123 

Temporary  ☐ Off


[Save](#) [Cancel](#)

22. Перейти на вкладку «**Role mapping**» и нажать кнопку «**Assign role**»

Users > User details

ivanova.t Enabled

Details Attributes Credentials **Role mapping** Groups Consents Identity provider links Sessions

 ☒ Hide inherited roles [Assign role](#) [Unassign](#)

<input type="checkbox"/>	Name	Inherited	Description
<input type="checkbox"/>	default-roles-cpr	False	\${role_default-roles}

23. Выбрать ранее созданные роли из списка и нажать кнопку «**Assign**»

Assign roles to ivanova.t account

Filter by realm roles Search by role name

Name	Description
<input type="checkbox"/> admin	`\${role_admin}`
<input type="checkbox"/> create-realm	`\${role_create-realm}`
<input checked="" type="checkbox"/> employee	
<input type="checkbox"/> offline_access	`\${role_offline-access}`
<input checked="" type="checkbox"/> supervisor	
<input type="checkbox"/> uma_authorization	`\${role_uma_authorization}`

Assign Cancel

Users > User details

ivanova.t Enabled Action

Details Attributes Credentials Role mapping Groups Consents Identity provider links Sessions

Search by name Hide inherited roles Assign role Unassign

Name	Inherited	Description
<input type="checkbox"/> default-roles-cpr	False	`\${role_default-roles}`
<input type="checkbox"/> employee	False	–
<input type="checkbox"/> supervisor	False	–

Пользователь с назначенными ролями

<input type="checkbox"/> ivanova.t	ivanova@mail.ru	Иванова	Таисия	–
<input type="checkbox"/> service-account-admin-cli	–	–	–	–
<input type="checkbox"/> uzorov.k	uzorov_02@mail.ru	Узоров	Кирилл	–

Опционально добавить второго пользователя

24. В боковом меню слева выбрать «**Client scopes**». В открывшемся списке найти строку «**roles**» и нажать на неё.

Manage	<input type="checkbox"/> address	Optional	OpenID Connect	–	OpenID Connect built-in scope: address
Clients	<input type="checkbox"/> email	Default	OpenID Connect	–	OpenID Connect built-in scope: email
Client scopes	<input type="checkbox"/> microprofile-jwt	Optional	OpenID Connect	–	Microprofile - JWT built-in scope
Realm roles	<input type="checkbox"/> offline_access	Optional	OpenID Connect	–	OpenID Connect built-in scope: offline_access
Users	<input type="checkbox"/> phone	Optional	OpenID Connect	–	OpenID Connect built-in scope: phone
Groups	<input type="checkbox"/> profile	Default	OpenID Connect	–	OpenID Connect built-in scope: profile
Sessions	<input type="checkbox"/> role_list	Default	SAML	–	SAML role list
Events	<input type="checkbox"/> roles	Default	OpenID Connect	–	OpenID Connect scope for add user roles to the access token
Configure	<input type="checkbox"/> web-origins	Default	OpenID Connect	–	OpenID Connect scope for add allowed web origins to the access token
Realm settings					
Authentication					
Identity providers					
User federation					

25. В открывшемся окне перевести переключатель «**Include in token scope**» в положение «**on**»

Manage

Clients

Client scopes

Realm roles

Users

Groups

Sessions

Events

Configure

Realm settings

Authentication

Identity providers

User federation

SettingsMappersScope

Name \* ⓘ

roles

Description ⓘ

OpenID Connect scope for add user roles to the access token

Type ⓘ

Default

Display on consent screen ⓘ

☒ On

Consent screen text ⓘ

`${rolesScopeConsentText}`

Include in token scope ⓘ

☒ On

Display Order ⓘ

Нажать кнопку «**Save**»

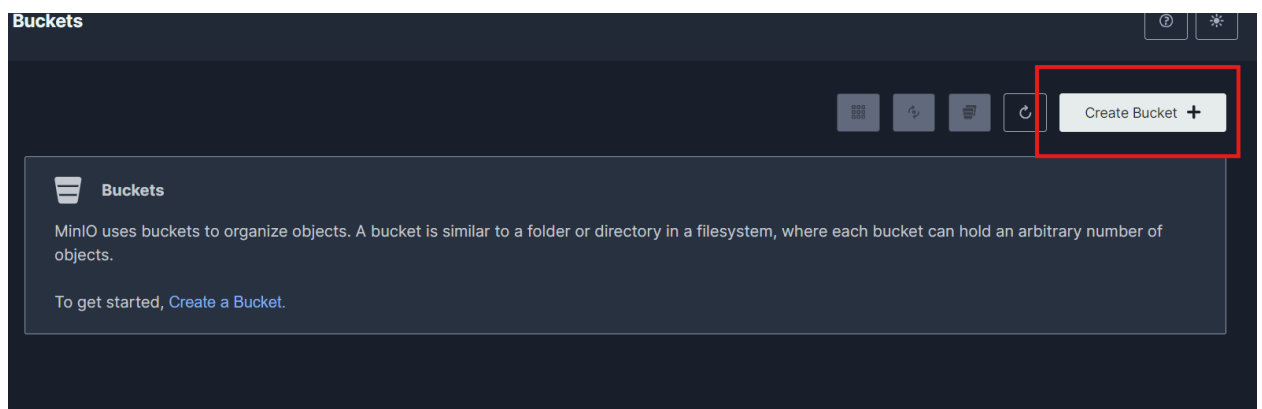
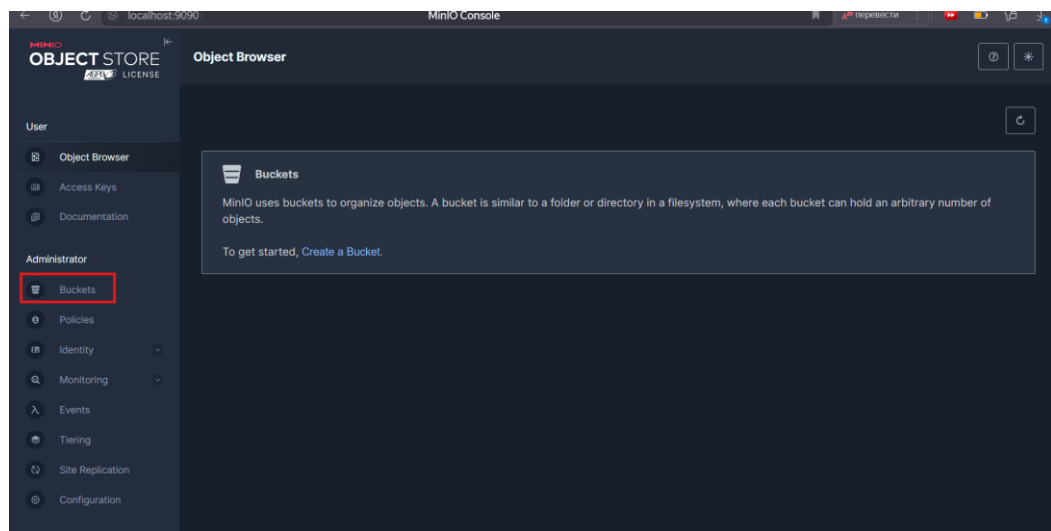
## Настройка MiniO

1. Перейти по пути: <http://localhost:9090/login>
2. Ввести логин и пароль:

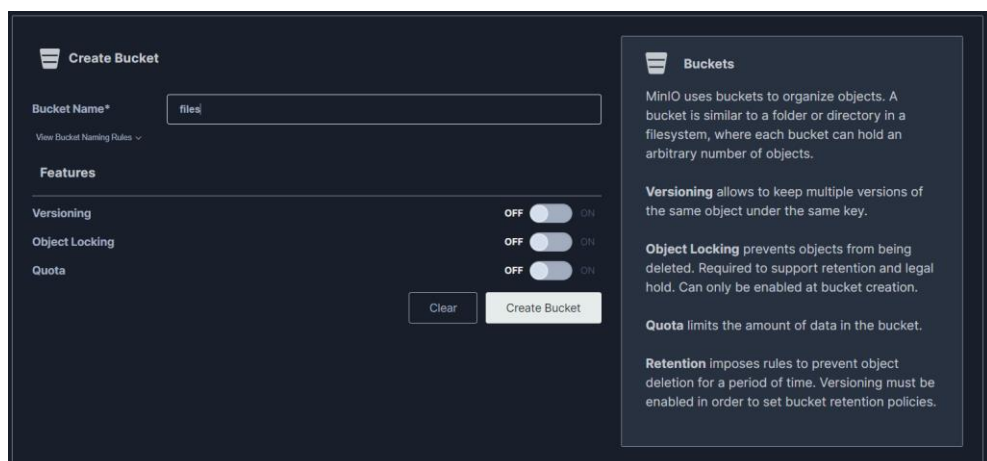
Логин: **UsEr\_y8b-DSq-C2K-t32**

Пароль: **y8b-DSq-C2K-t32**

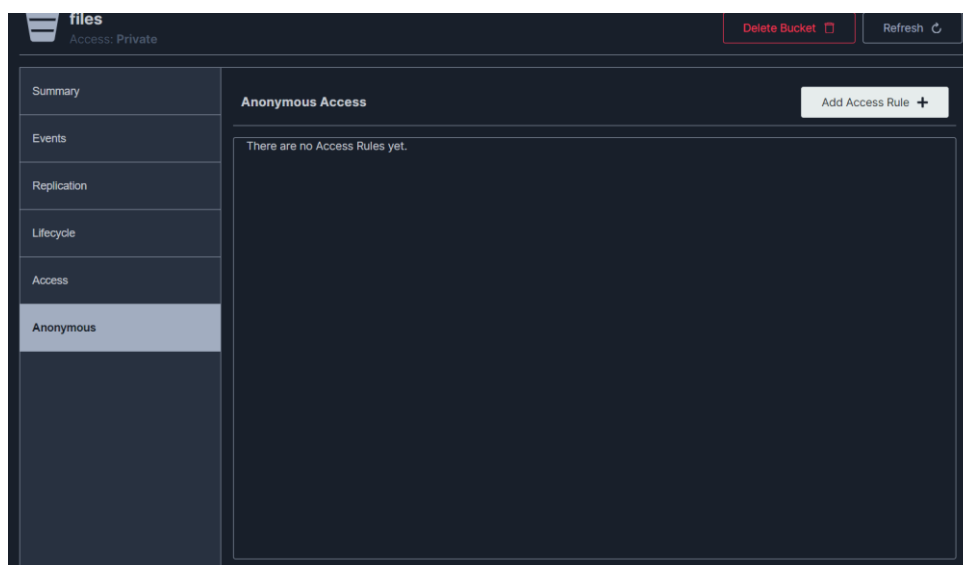
3. В боковом меню слева выбрать «**Buckets**» и нажать кнопку «**Create Bucket**»



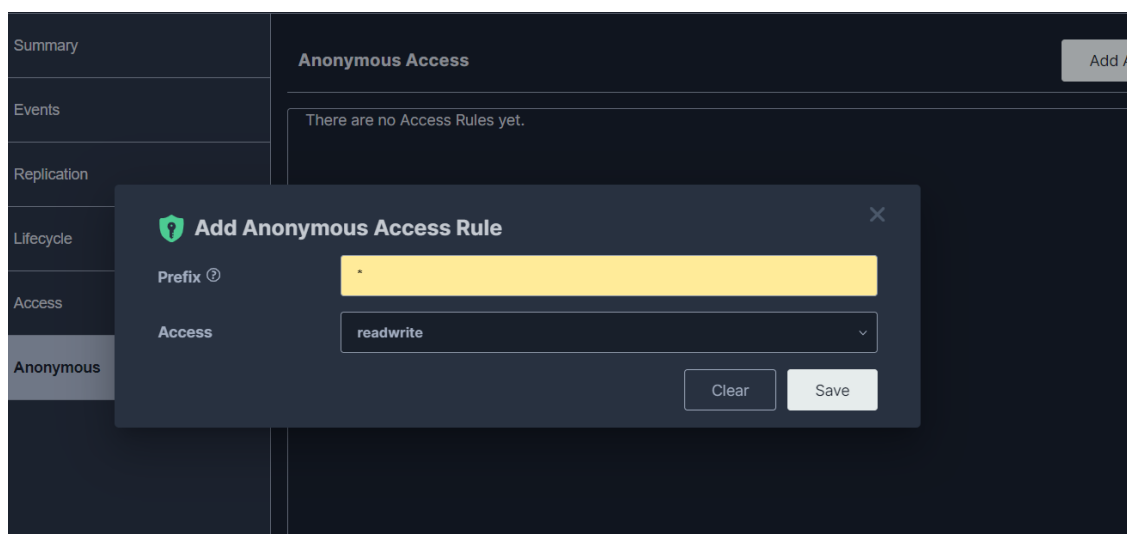
4. Ввести имя Bucket: **files** и нажать кнопку «**Create bucket**»



5. Нажать на созданный Bucket и выбрать в боковом меню окна вкладку «Anonymous».



6. Нажать на кнопку «Add Access Rule» и в поле «Prefix» указать символ «\*», а в выпадающем списке в поле «Access» выбрать «readwrite», после чего нажать кнопку «Save».



## Настройка сервисов

1. Открыть файл **docker-compose.yaml** и проверить, что все данные, указанные в передаваемых в контейнеры переменных, корректны. Ниже представлено описание всех переменных для каждого сервиса:

Название сервиса	Название переменной	Описание
<b>document-processing-service</b>	POSTGRES_DB	Название созданной базы данных (Значение по умолчанию, не нужно изменять)
	POSTGRES_USER	Имя пользователя базы данных (Значение по умолчанию, не нужно изменять)
	POSTGRES_PASSWORD	Пароль пользователя базы данных (Значение по умолчанию, не нужно изменять)
	POSTGRES_HOST	Хост, на котором прослушивает развёрнутый контейнер с БД (Значение по умолчанию, не нужно изменять)
	POSTGRES_PORT	Порт, на котором прослушивает развёрнутый контейнер с БД (Значение по умолчанию, не нужно изменять)
	MINIO_HOST	Хост, на котором прослушивает развёрнутый контейнер с MiniO (Значение по умолчанию, не нужно изменять)
	MINIO_ACCESS_KEY	Логин MiniO (Значение по умолчанию, не нужно изменять)
	MINIO_SECRET_KEY	Пароль MiniO (Значение по умолчанию, не нужно изменять)
	HOST_IP	Хост, где развёрнут сам сервис (Значение по умолчанию, не нужно изменять)
<b>gateway</b>	DOCUMENT_SERVICE_URL	URL сервиса <b>document-processing-service</b> (Значение по умолчанию, не нужно изменять)
	TEXT_CORRECTION_SERVICE_URL	URL сервиса <b>ai-service</b> (Значение по умолчанию, не нужно изменять)
	KEYCLOAK_SERVER	URL адрес Keycloak (Значение по умолчанию, не нужно изменять)
	KEYCLOAK_CLIENT_ID	ID клиента Keycloak (Значение по умолчанию, не нужно изменять)
	KEYCLOAK_CLIENT_SECRET	Секрет клиента Keycloak (Значение необходимо изменить, см. п.13 Настройка Keycloak)
	KEYCLOAK_ADMIN_ID	ID админа Keycloak (Значение по умолчанию, не нужно изменять)



	KEYCLOAK_ADMIN_SECRET	Секрет админа Keycloak (Значение необходимо изменить, см. п.18 Настройка Keycloak)
	KEYCLOAK_ADMIN_USERNAME	Имя админа Keycloak (Значение по умолчанию, не нужно изменять)
	KEYCLOAK_ADMIN_PASSWORD	Пароль админа Keycloak (Значение по умолчанию, не нужно изменять)
web-interface	VUE_APP_GATEWAY_URL	URL шлюза приложения (Сервис <b>gateway</b> – значение по умолчанию, не нужно изменять)

- После проверки корректности переменных, находясь в директории `./CPR_services/`, ввести команду: ***docker compose up --build -d***
- Сборка сервисов занимает продолжительное время (~ 15 минут) и требует не менее 25ГБ свободного места на диске, где происходит развёртывание.

Если во время развёртывания сервис **document-processing-service** выдаёт ошибку: **./entrypoint.sh: not found** – нужно перейти в директорию `document-processing-service/app/`, открыть в этой директории файл `./entrypoint.sh`, скопировать его содержимое, затем удалить файл `entrypoint.sh`, снова создать его в этой же директории и вставить в него скопированное содержимое. (Ошибка связана с повреждением файла скрипта при скачивании с github).

Сервис **document-processing-service** должен успешно применить миграции к базе данных:

```
document-processing-service-1 | INFO [alembic.runtime.migration] Context impl PostgresqlImpl.
document-processing-service-1 | INFO [alembic.runtime.migration] Will assume transactional DDL.
document-processing-service-1 | INFO [alembic.runtime.migration] Will assume transactional DDL.
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade -> d7f00add2de0, Initial migration
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade -> d7f00add2de0, Initial migration
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade d7f00add2de0 -> 50aae5492d5b, Document changed 2 migration
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade d7f00add2de0 -> 50aae5492d5b, Document changed 2 migration
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade 50aae5492d5b -> 398a57485399, Change registration_date to Date
document-processing-service-1 | INFO [alembic.runtime.migration] Running upgrade 50aae5492d5b -> 398a57485399, Change registration_date to Date
document-processing-service-1 | INFO: Started server process [11]
document-processing-service-1 | INFO: Waiting for application startup.
document-processing-service-1 | INFO: Application startup complete.
document-processing-service-1 | INFO: Uvicorn running on http://0.0.0.0:83 (Press CTRL+C to quit)
```

- По завершении сборки интерфейс системы будет доступен по пути: <http://localhost:8000/>. Войти в систему можно под одной из ранее созданных учётных записей пользователей:

Цифровой помощник руководителя

Логин

ivanova.t

Пароль

\*\*\*

ВОЙТИ

Word localhost:8000 Сервис работы с поручениями "ЦПР"

Поручения успешно загружены

Цифровой помощник руководителя

ДОБАВИТЬ ПОРУЧЕНИЕ

Фильтр по статусу

Фильтр по приоритету

Поиск по заголовку

Поручения, назначенные на меня (0)

Поручения, созданные мной (1)