ФИЛОСОФИЯ.ИТ
РОСАТОМ

Общество с ограниченной ответственностью «Философия.ИТ»

Россия, 107023, г. Москва, ул. Измайловский Вал, д. 30
Тел.: +7 (495) 988-37-38, факс: +7 (495) 988-37-38,
e-mail: customer@fil-it.ru

ИНН 7713728490
КПП 771901001
ОГРН 1117746379145

ПОШАГОВАЯ ИНСТРУКЦИЯ
ПО РАЗВЕРТЫВАНИЮ ЭКЗЕМПЛЯРА ПО

ПЛАТФОРМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА «КОГНИТРОН»

г. Москва

Оглавление

1. Общие положения 3
1.1. Полное наименование программы для ЭВМ, обозначение 3
1.2. Назначение системы 3
1.3. Разработчик системы 3
1.4. Назначение документа 3
2. Требования к аппаратному обеспечению 4
3. Требования к программному обеспечению 5
4. Установка ПО 6
4.1. Установка зависимого ПО 6
4.2. Размещение дистрибутива и разворачивание файлов 7
4.3. Импорт образов контейнеров 7
4.4. Настройка окружения (.env) 8
4.5. Установка сертификатов TLS 8
4.6. Запуск и остановка системы 9
5. Настройка аутентификации 9
6. Доступ к системе 14
7. Контактная информация 14

1. Общие положения

1.1. Полное наименование программы для ЭВМ, обозначение

Полное наименование Программы для ЭВМ: Платформа искусственного интеллекта «Когнитрон» – далее по тексту Система.

1.2. Назначение системы

«Когнитрон» - комплексное программное решение, объединяющее инструменты обучения и развертывания моделей искусственного интеллекта и создания ИИ-помощников и ИИ-агентов. Платформа поддерживает работу с большими языковыми моделями (LLM) модальности любого типа: текст, изображение, аудио, видео. Решение выступает в роли среды, обеспечивающей коммуникацию и координацию ИИ-агентов между собой, и средства интеграции с внешними ИТ-системами.

1.3. Разработчик системы

Полное наименование: Общество с ограниченной ответственностью «Философия.ИТ». Сокращенное наименование: ООО «Философия.ИТ».

1.4. Назначение документа

Настоящий документ входит в комплект эксплуатационной документации по Платформе искусственного интеллекта «Когнитрон» и содержит пошаговую инструкцию для развертывания Системы.

2. Требования к аппаратному обеспечению

Для разворачивания комплекса необходимо подготовить сервера, который должен отвечать следующим характеристикам:

а) Сервер LLM

- CPU: 16 ядер;
- Оперативная память (MEM): 64 ГБ;
- Накопитель (SSD): 100 ГБ;
- GPU: NVIDIA A100 / H100, 80 ГБ.

б) Бэкэнд-сервер

- CPU: 16 ядер;
- Оперативная память (MEM): 32 ГБ;
- Накопитель (SSD): 300 ГБ.

3. Требования к программному обеспечению

Для разворачивания комплекса на серверах необходимо установить следующие компоненты:

а) Сервер LLM

- ОС: РЭДОС 8, Astra Linux 1.8 и более поздние выпуски;
- ПО: Docker Engine, Docker compose, NVIDIA Container Toolkit, NVIDIA GPU driver.

б) Бэкэнд-сервер

- ОС: РЭДОС 8, Astra Linux 1.8 и более поздние выпуски
- ПО: Docker Engine, Docker compose.

4. Установка ПО

Развертывание системы выполняется путем импортирования образов контейнеров и запуска контейнеров с помощью файла конфигурации `docker compose`.

Для выполнения установки необходим доступ к операционной системе с правами администратора (root). Необходимо ознакомится с документацией на необходимые компоненты по ссылкам:

- https://developer.nvidia.com/cuda-downloads/
- https://docs.docker.com/engine/install/ubuntu/
- https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html
- https://catalog.ngc.nvidia.com/orgs/nvidia/containers/pytorch

Подключитесь к серверу с помощью команды `ssh admin@сервер.домен.`

4.1. Установка зависимого ПО

Отключение автоматического обновления

Отключите автообновления командами:

```
#!/bin/bash
sudo systemctl stop unattended-upgrades
sudo systemctl disable unattended-upgrades
sudo systemctl mask unattended-upgrades
```

Установка драйвера NVIDIA

Установите драйвера NVIDIA следующими командами.

```
#!/bin/bash
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt update
sudo apt install -y nvidia-driver-570-server-open
sudo reboot
```

Проверьте: `nvidia-smi`

Установка Docker и Docker Compose

Для установки Docker и Docker Compose выполните следующие команды.

```
#!/bin/bash
sudo apt update
sudo apt install -y ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```
echo "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "${UBUNTU_CODENA
ME:-$VERSION_CODENAME}") stable" | sudo tee /etc/apt/sources.list.d/docker.list > /de
v/null
sudo apt update
sudo apt install -y docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker
-compose-plugin
sudo systemctl enable docker
sudo systemctl start docker
sudo usermod -aG docker $USER
```

Проверьте: `docker --version && docker compose version`

Установка NVIDIA Container Toolkit

```
#!/bin/bash
curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -
o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg
curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-t
oolkit.list | sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container
-toolkit-keyring.gpg] https://#g' | sudo tee /etc/apt/sources.list.d/nvidia-container
-toolkit.list
sudo apt update
sudo apt install -y nvidia-container-toolkit
sudo nvidia-ctk runtime configure --runtime=docker
sudo systemctl restart docker
```

Проверьте: `docker run --gpus all -it --rm nvcr.io/nvidia/cuda:12.3.1-base-ubuntu22.04 nvidia-smi`

4.2. Размещение дистрибутива и разворачивание файлов

Скопируйте дистрибутив на сервер: `scp cognitron-*.tgz admin@сервер:/opt/cognitron-dist`.

На сервере:

```
#!/bin/bash
mkdir -p /opt/cognitron
tar -xzvf /opt/cognitron-dist/cognitron-config.tgz -C /opt/cognitron
tar -xzvf /opt/cognitron-dist/cognitron-vllm.tgz -C /opt/cognitron/vllm/cache
tar -xzvf /opt/cognitron-dist/cognitron-infinity.tgz -C /opt/cognitron/infinity/cache
```

4.3. Импорт образов контейнеров

Образы контейнеров поставляются в виде файла архива `container_images.tgz`

Для импорта образов контейнеров выполните команду:

```
#!/bin/bash
docker load -i /opt/cognitron-dist/cognitron-images.tgz
```

4.4. Настройка окружения (.env)

Отредактируйте файл `/opt/cognitron/.env`

Перечень переменных окружения

Имя переменной	Значение	Примечание
COGNITRON_URL_HOST	cognitron.domain.local	Указать полное DNS имя сервера, по которому будет осуществятся доступ к системе
COGNITRON_DB_DATABASE	cognitron	Имя базы данных
COGNITRON_DB_USERNAME	cognitron	Имя пользователя доступа к базе данных
COGNITRON_DB_PASSWORD	**password**	Пароль доступа к базе данных
COGNITRON_OS_USERNAME	cognitron	Имя пользователя доступа к OpenSearch
COGNITRON_OS_PASSWORD	**password**	Пароль доступа к OpenSearch
COGNITRON_KEYCLOAK_PREFIX	/auth	Префикс доступа к консоли Keycloak
COGNITRON_KEYCLOAK_ADMIN	keycloak	Имя администратора Keycloak
COGNITRON_KEYCLOAK_ADMIN_PASSWORD	**password**	Пароль администратора Keycloak
COGNITRON_KEYCLOAK_DB_DATABASE	keycloak	Имя базы данных Keycloak
COGNITRON_KEYCLOAK_DB_USERNAME	keycloak	Имя пользователя доступа к базе данных Keycloak
COGNITRON_KEYCLOAK_DB_PASSWORD	**password**	Пароль доступа к базе данных Keycloak
COGNITRON_PGADMIN_PREFIX	/pgadmin	Префикс доступа к консоли PGadmin
COGNITRON_PGADMIN_DEFAULT_EMAIL	admin@pgadmin.com	Имя администратора PGadmin
COGNITRON_PGADMIN_DEFAULT_PASSWORD	**password**	Пароль администратора PGadmin
COGNITRON_REDIS_PASSWORD	**password**	Пароль Redis
COGNITRON_VLLM_API_KEY	**api_key_64_characters**	Ключ доступа сервера vLLM
COGNITRON_INFINITY_API_KEY	**api_key_64_characters**	Ключ доступа сервера Infibity
COGNITRON_OPENSEARCH_PREFIX	/osds	Префикс доступа к консоли OpenSearch
COGNITRON_OPENSEARCH_PASSWORD	**password**	Пароль администратора OpenSearch

4.5. Установка сертификатов TLS

Разместите в файлах на сервере
Закрытый ключ в файле `/opt/cognitron/traefik/cert.key`
Цепочка сертификатов в файле `/opt/cognitron/traefik/cert.cer`

4.6. Запуск и остановка системы

Запуск и остановка системы выполняется командой `docker compose` в директории `/opt/cognitron`

Для запуска системы выполните команду:

```
cd /opt/cognitron; docker compose up -d
```

В результате выполнения команды будут созданы контейнеры в целевой конфигурации и будет запущена система с автоматически запуском при загрузке операционной системы.

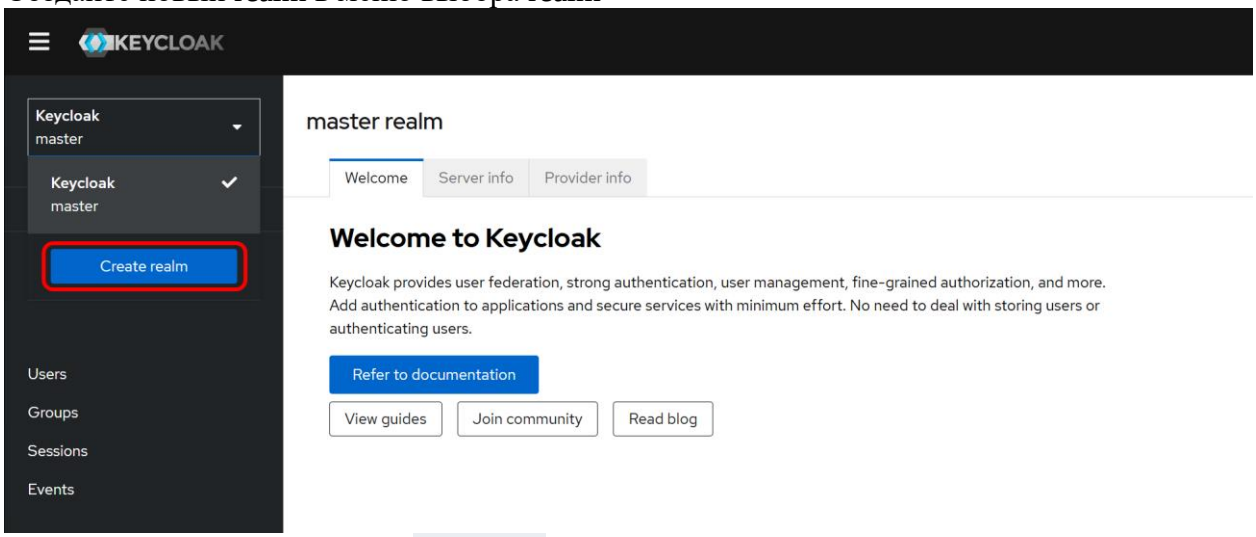
Для остановки системы и удаления контейнеров выполните команду:

```
cd /opt/cognitron; docker compose down
```

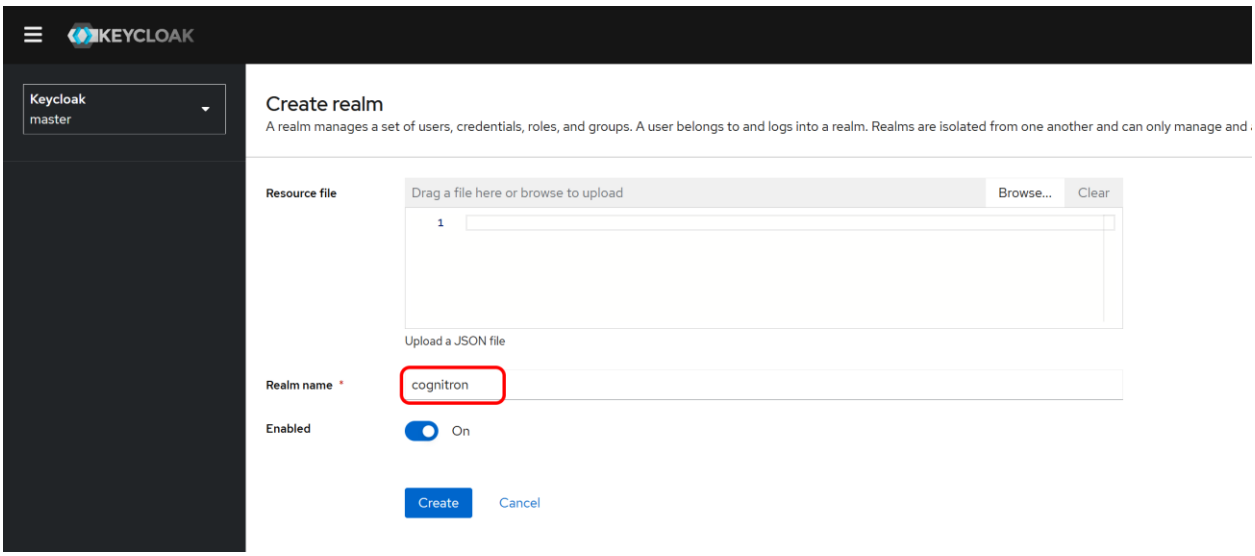
В результате система будет остановлена, контейнеры будут удалены, автоматического запуска системы при загрузке операционной системы не будет выполнятся.

5. Настройка аутентификации

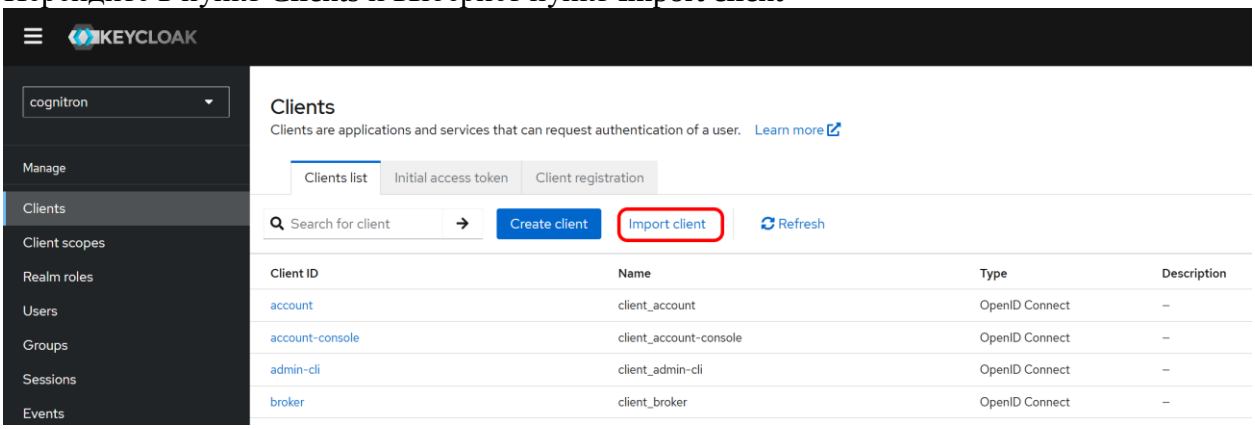
Откройте консоль управления Keycloak по ссылке `https://cognitron.domain.local/auth` (используйте значения переменных `COGNITRON_URL_HOST` и `COGNITRON_KEYCLOAK_PREFIX`)
В качестве имени пользователя и пароля используйте значения из переменных COGNITRON_KEYCLOAK_ADMIN и `COGNITRON_KEYCLOAK_ADMIN_PASSWORD`.
Создайте новый realm в меню выбора realm



Установите имя нового realm `cognitron`



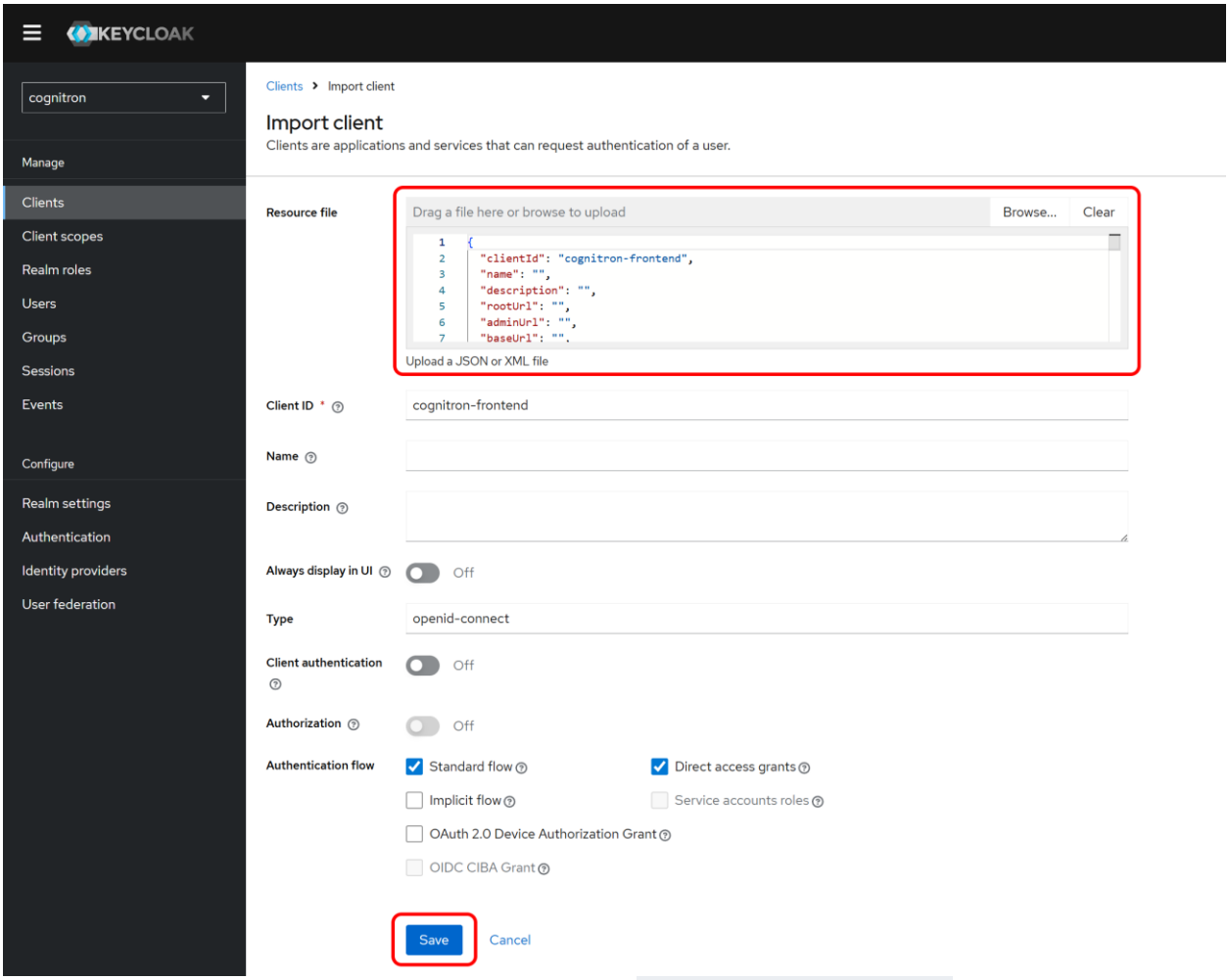
Перейдите в пункт Clients и выбериет пункт Import client



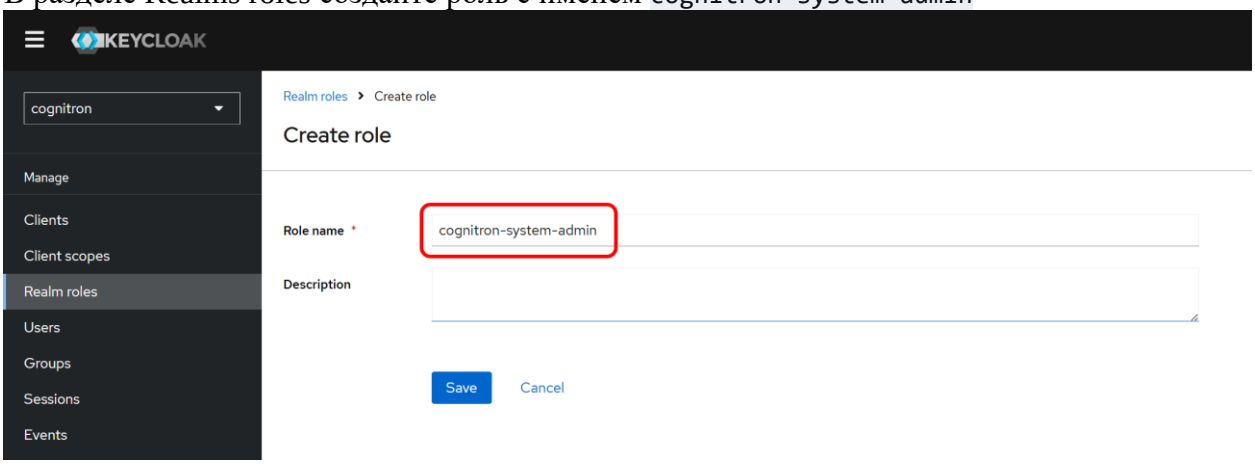
Загрузите JSON файл следующего содержимого.

```
{
  "clientId": "cognitron-frontend",
  "name": "",
  "description": "",
  "rootUrl": "",
  "adminUrl": "",
  "baseUrl": "",
  "surrogateAuthRequired": false,
  "enabled": true,
  "alwaysDisplayInConsole": false,
  "clientAuthenticatorType": "client-secret",
  "redirectUris": [
    "/*"
  ],
  "webOrigins": [
    "*"
  ],
  "notBefore": 0,
  "bearerOnly": false,
  "consentRequired": false,
  "standardFlowEnabled": true,
  "implicitFlowEnabled": false,
  "directAccessGrantsEnabled": true,
  "serviceAccountsEnabled": false,
  "publicClient": true,
  "frontchannelLogout": true,
  "protocol": "openid-connect",
  "attributes": {
    "client.secret.creation.time": "1729624683",
    "client.introspection.response.allow.jwt.claim.enabled": "false",
```

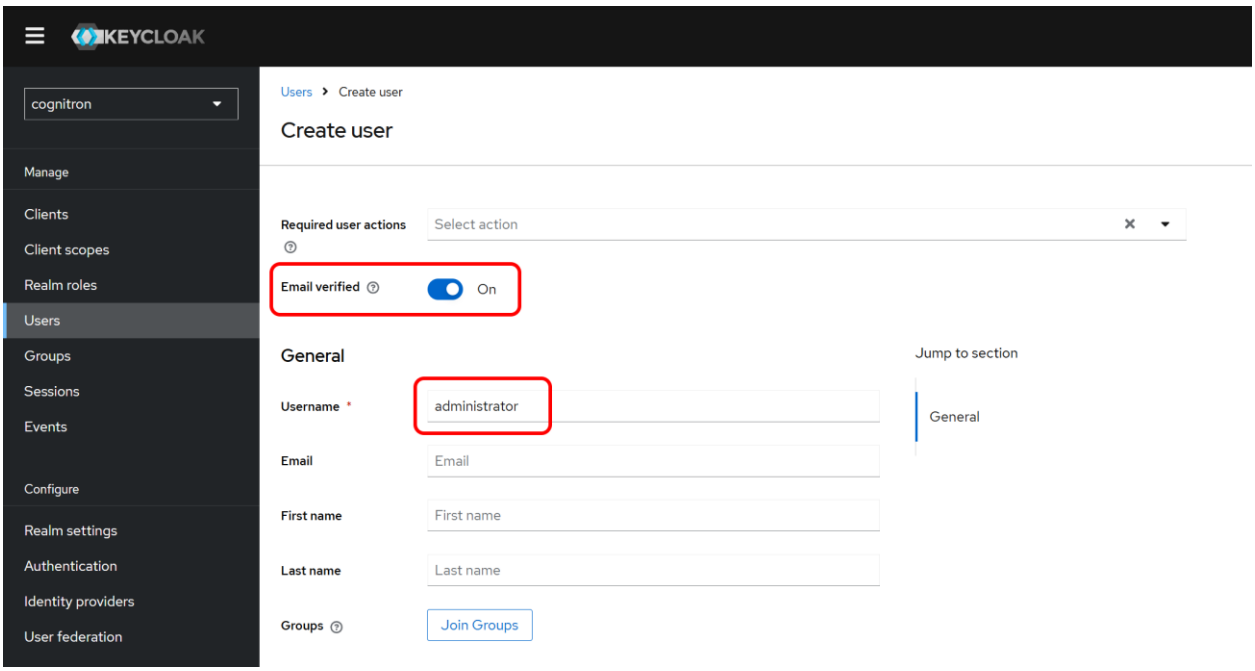
```
    "post.logout.redirect.uris": "+",
    "oauth2.device.authorization.grant.enabled": "false",
    "use.jwks.url": "false",
    "backchannel.logout.revoke.offline.tokens": "false",
    "use.refresh.tokens": "true",
    "exclude.session.state.from.auth.response": "false",
    "realm_client": "false",
    "oidc.ciba.grant.enabled": "false",
    "client.use.lightweight.access.token.enabled": "false",
    "exclude.issuer.from.auth.response": "true",
    "backchannel.logout.session.required": "true",
    "client_credentials.use_refresh_token": "false",
    "acr.loa.map": "{}",
    "require.pushed.authorization.requests": "false",
    "tls.client.certificate.bound.access.tokens": "false",
    "display.on.consent.screen": "false",
    "token.response.type.bearer.lower-case": "false"
  },
  "authenticationFlowBindingOverrides": {},
  "fullScopeAllowed": true,
  "nodeReRegistrationTimeout": -1,
  "defaultClientScopes": [
    "web-origins",
    "acr",
    "profile",
    "roles",
    "basic",
    "email"
  ],
  "optionalClientScopes": [
    "address",
    "phone",
    "organization",
    "offline_access",
    "microprofile-jwt"
  ],
  "access": {
    "view": true,
    "configure": true,
    "manage": true
  }
}
```



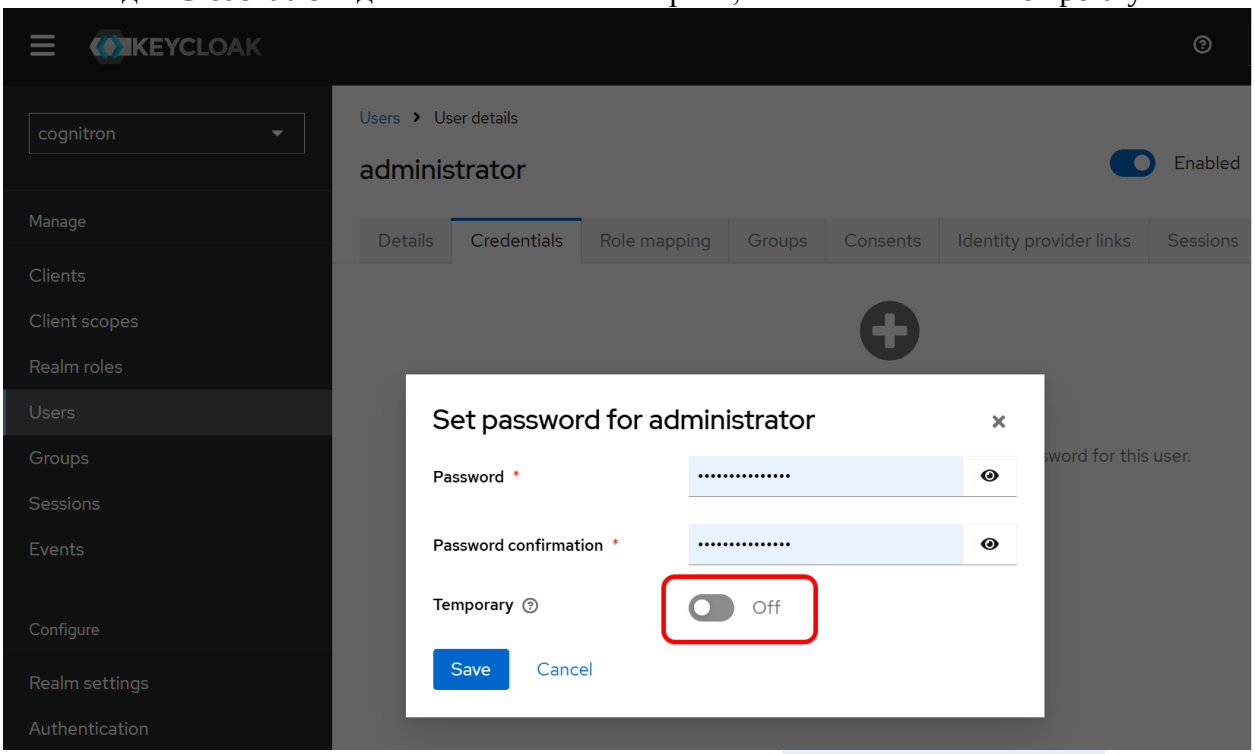
В разделе Realms roles создайте роль с именем `cognitron-system-admin`



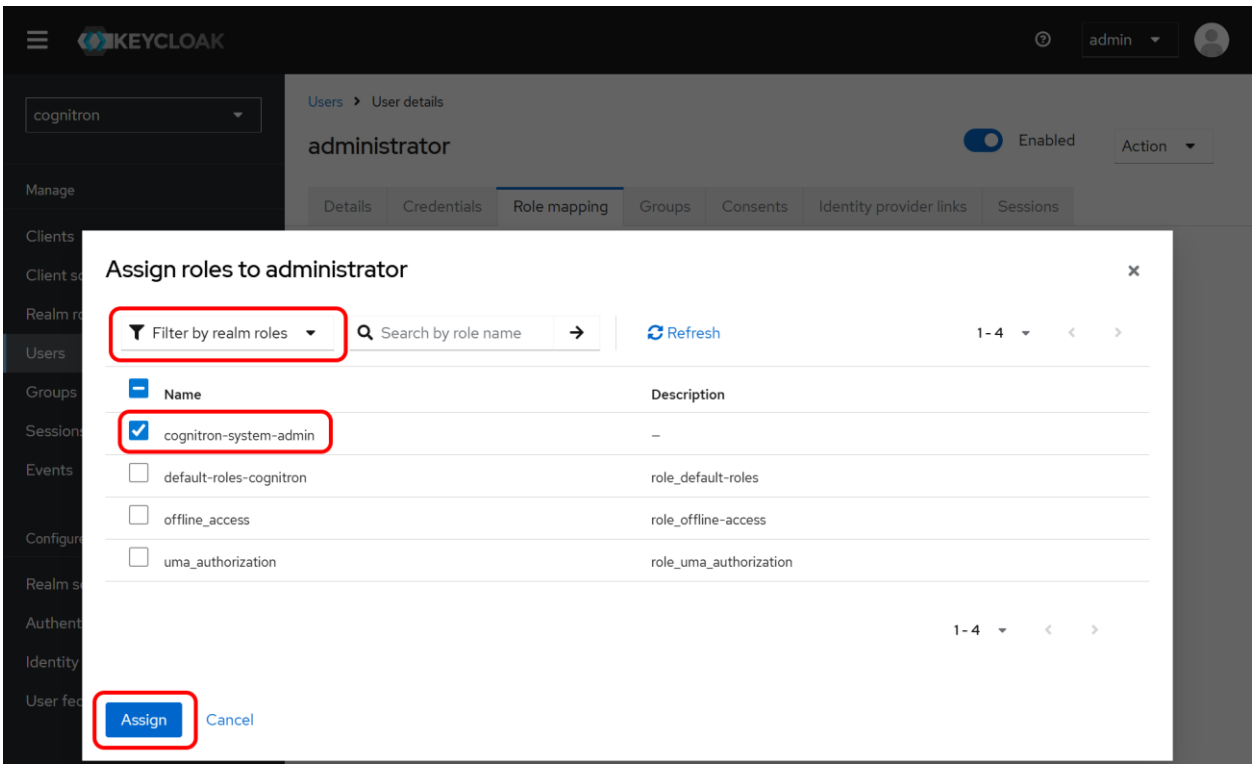
В разделе Users создайте пользователя с произвольным именем, установите значение Email verified



На закладке Credentials задайте пользователю пароль, снимите значение Temporary



На закладке Role mapping присвойте пользователю роль `cognitron-system-admin`, для чего выбрать фильтр ролей realms и выбрать роль cognitron-system-admin.



6. Доступ к системе

В браузере перейдите по адресу `https://cognitron.domain.local/auth` (используйте значения переменой `COGNITRON_URL_HOST`)
Для входа используйте учетные данные пользователя с ролью cognitron-system-admin, созданные на этапе настройки аутентификации.

7. Контактная информация

В случае возникновения вопросов по настоящей инструкции просим обратиться к следующим специалистам:

Белоусов Олег Викторович,

Директор по искусственному интеллекту

+7 (916) 904 30 88,

obelousov@fil-it.ru

Русанов Николай Константинович,

Системный архитектор

+7(916) 603 76 15,

nrusanov@fil-it.ru