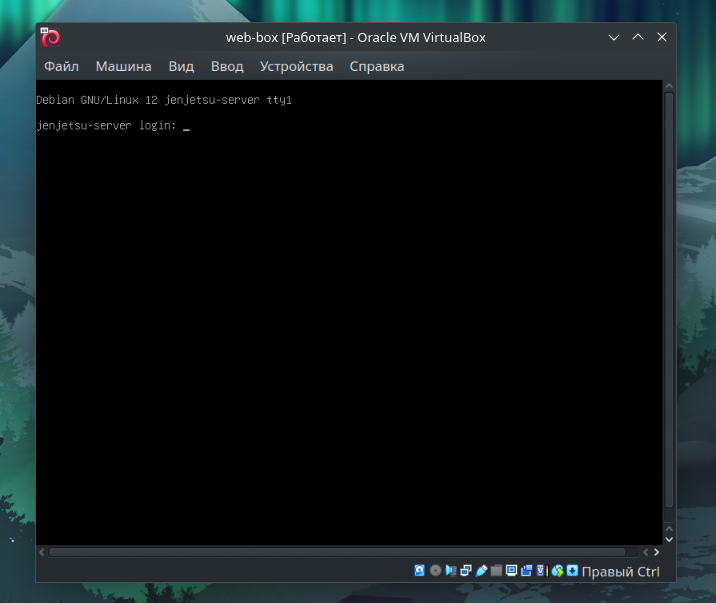
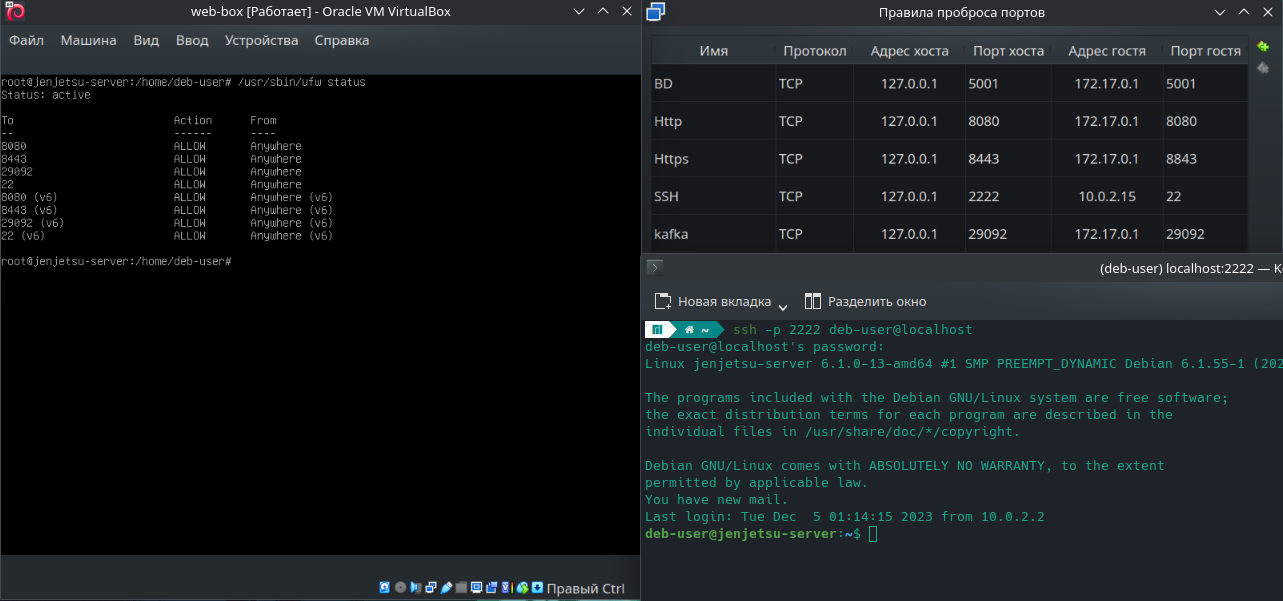
Шаг 1. Развёртывание консольной версии Debian дистрибутива на виртуальной машине и настраиваем для дальнейшей работы



Для этого:

1. Установлен docker при помощи команд
   1. apt install apt-transport-https ca-certificates curl software-properties-common - установка пакетного менеджера для apt
   2. curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg - добавление gpg ключа репозитория docker
   3. echo "deb [arch=amd64 signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/debian $(lsb\\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null - добавление репозитория docker в apt
   4. apt update - обновление пакетов
   5. apt-cache policy docker-ce && apt install docker-ce - переключение в репозиторий docker и установка
   6. systemctl start docker && systemctl status docker - запуск сервиса и проверка состояния
   7. usermod -aG docker deb-user - добавление обычного пользователя в группу docker
2. Настроен ssh-сервер при помощи команд:
   1. apt install openssh-server - установка ssh-сервера
   2. systemctl start ssh - запуск службы ssh
   3. systemctl status ssh - проверка работоспособности сервера + узнаём порт работы (22)
3. Установка ufw firewall при помощи команд:
   1. apt install ufw
   2. /usr/sbin/ufw default deny incoming - запрет на любые входящие соедининия
   3. /usr/sbin/ufw default deny outcoming - запрет на любые выходящие соедининия
   4. /usr/sbin/ufw allow 22 - разрешение на порты ssh
   5. /usr/sbin/ufw allow 8080 & /usr/sbin/ufw allow 8443 - разрешение на порты keycloak
4. Прокидываем порты 22 и 8080 через виртуалку

Итог: (Забыл про ufw, только недавно настроил)



Шаг 2. Развёртывание docker образов на виртуалке

Описание dockerfile для keycloak:

FROM openjdk:17-alpine

RUN apk update && apk add bash

RUN apk install curl

ENV KC\_HEALTH\_ENABLED=true

ENV KC\_INFO\_ENABLED=true

COPY ./keycloak-22.0.5 /keycloak

EXPOSE 8080

EXPOSE 8443

ENTRYPOINT ["/keycloak/bin/kc.sh", "-v", "start"]

Описание docker-compose.yml файла:

version: "3.8"

services:

web-database:

image: postgres:15.4

restart: always

container\_name: web-database

environment:

- POSTGRES\_DB=jenjetsu\_database

- POSTGRES\_USER=Stey

- POSTGRES\_PASSWORD=<net tam parolya>

volumes:

- ./pgdata:/var/lib/postgresql/data

- ./2. Init Database:/docker-entrypoint-initdb.d

ports:

- "5001:5432"

healthcheck:

test: pg\_isready -d postgres

interval: 10s

timeout: 5s

start\_period: 20s

keycloak:

restart: always

build:

context: ./

dockerfile: keycloak.dockerfile

container\_name: keycloak

environment:

- KEYCLOAK\_ADMIN=admin

- KEYCLOAK\_ADMIN\_PASSWORD=<zaporoleno>

- KC\_DB\_URL\_HOST=web-database

- KC\_DB\_URL\_PORT=5432

- KC\_DB=postgres

- KC\_DB\_URL\_DATABASE=jenjetsu\_database

- KC\_DB\_SCHEMA=keycloak

- KC\_DB\_USERNAME=keycloak\_user

- KC\_DB\_PASSWORD=ataW9nSrgfEo7mt

- KC\_HOSTNAME\_ADMIN=localhost

- KC\_HOSTNAME\_STRICT=false

- KC\_HOSTNAME\_STRICT\_HTTPS=false

- KC\_HTTP\_ENABLED=true

- KC\_HTTP\_HOST=0.0.0.0

- KC\_HOSTNAME=localhost

ports:

- "8080:8080"

- "8443:8443"

healthcheck:

test: curl -X GET http://localhost:8080/health

interval: 10s

timeout: 5s

start\_period: 20s

depends\_on:

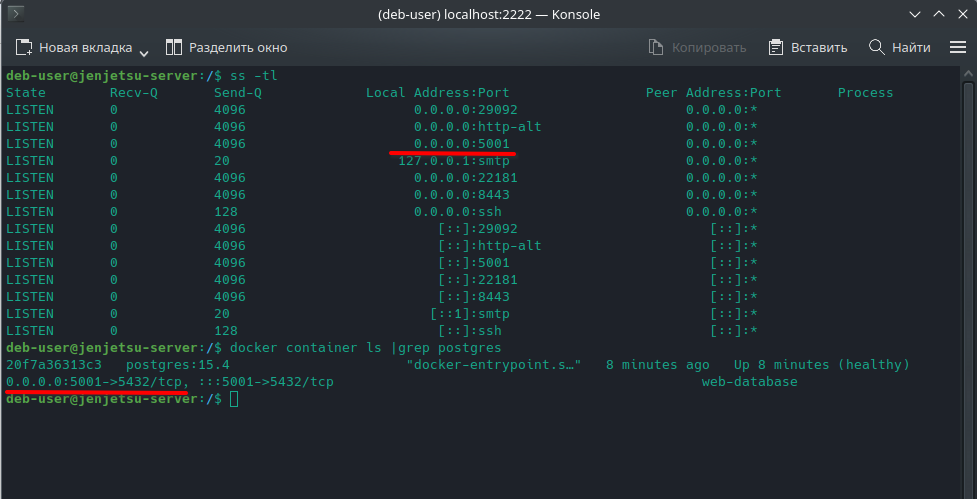
- web-database

Порты у СУБД:

5432 - порт контейнера внутри сети docker, к которому подключаются другие сервисы (keycloak и т.д.)

5001 - порт для хостовой машины для подключения к БД во время написания rest сервиса (прокинут через виртуальную машину)

Вывод по портам:



В БД jenjetsu\_database была создана схема для keycloak + создан отдельный

пользователь с правами работы только в этой схеме

Подключение к БД: docker container exec -it web-database psql -U Stey -d jenjetsu\_database

Создание схемы: CREATE SCHEMA keycloak

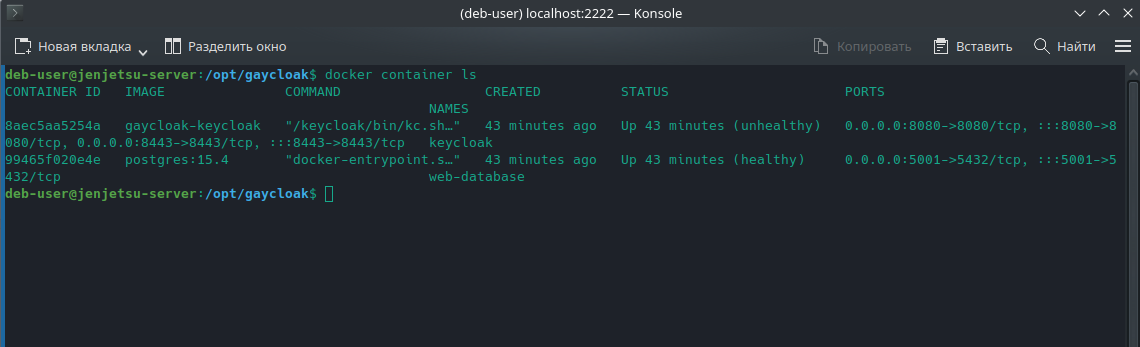
Создание пользователя: CREATE USER keycloak\_user WITH CREDENTIALS ‘ataW9nSrgfEo7mt’

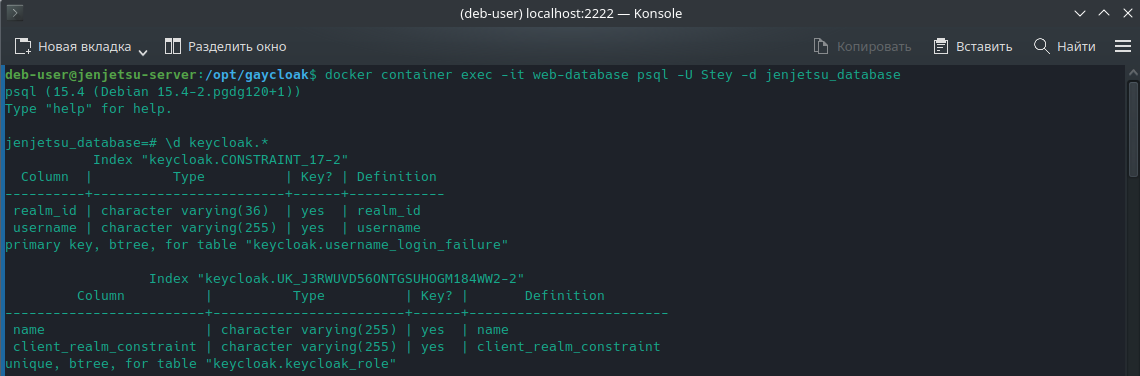
Выдача прав:

GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA keycloak TO keycloak\_user

GRANT ALL PRIVILEGES ON ALL SEQUENCES IN SCHEMA keycloak TO keycloak\_user

Итого:





Шаг 3. Настраиваем keycloak

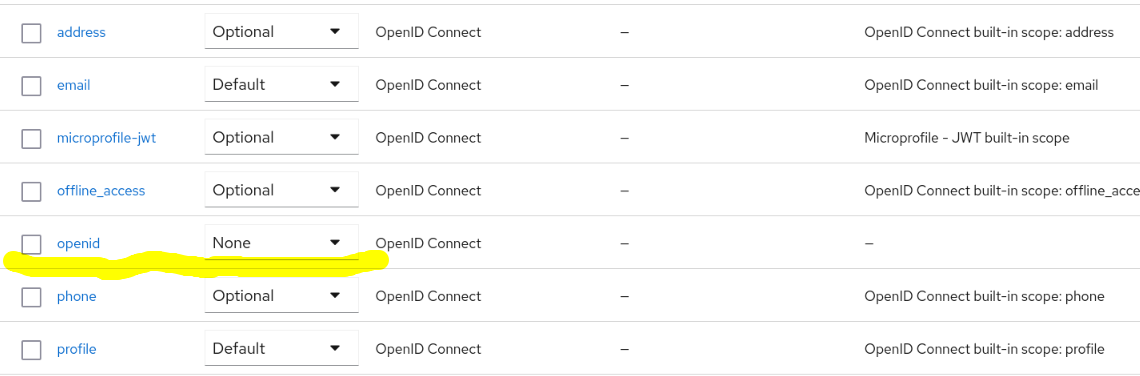
Настройки реалма:

Для вкладки Login: User registration, Forgot password, Remember me, Login with email, Verify email

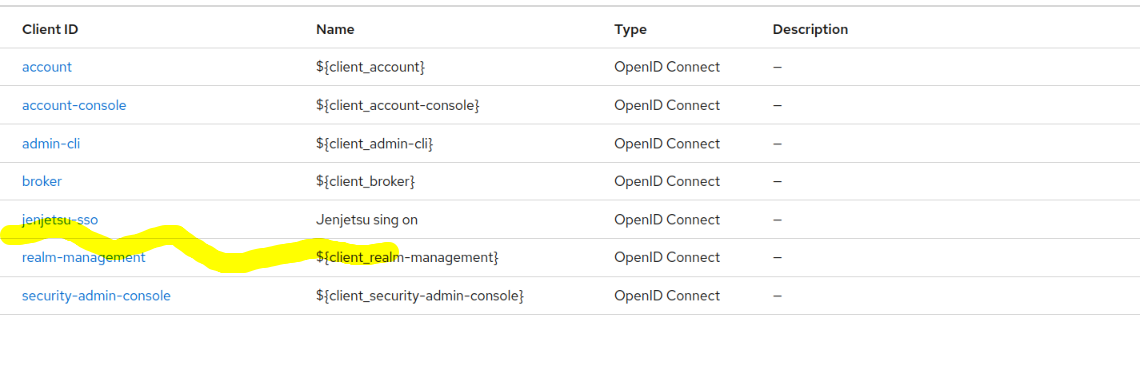
Для вкладки Email: Настроен from - моя gmail почта, Connection & authentication - host=smtp.gmail.com, port=587, encryption=enable-startTLS

Access token был выставлен на 150 мин для простоты тестирования, должен быть 15 мин.

Создан scope openid для клиента:



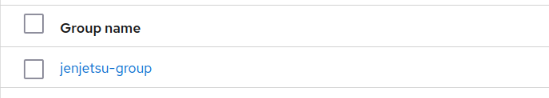
Сам клиент:



Создаём пользователей (кроме lakmilak1 - проверка регистрации по почте)



Группа, роль (ROLE\_MODERATOR будет использована в 4 лабе)





Шаг 5. Настройка дампа БД

Для этого:

1. apt install cron
2. Создание скрипта для дампа бд
3. Настройка crontab для дампа каждый час

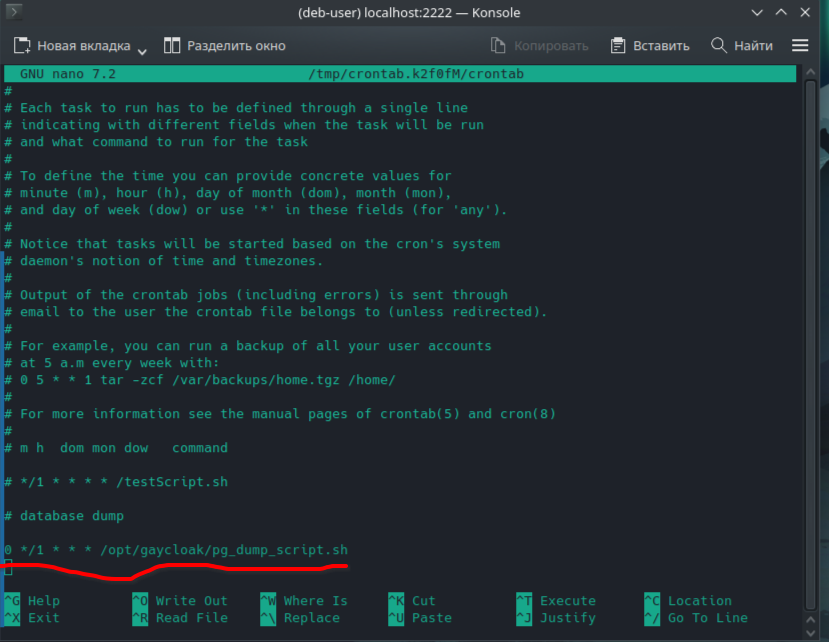
Скрипт дампа:

#usr/bin/env bash

currentTime=`date +"%Y-%m-%d-%H-%M-%S"`

/usr/bin/docker container exec -it web-database pg\_dump -U Stey -d jenjetsu\_database > /opt/gaycloak/dumps/dump\_${currentTime}.dump

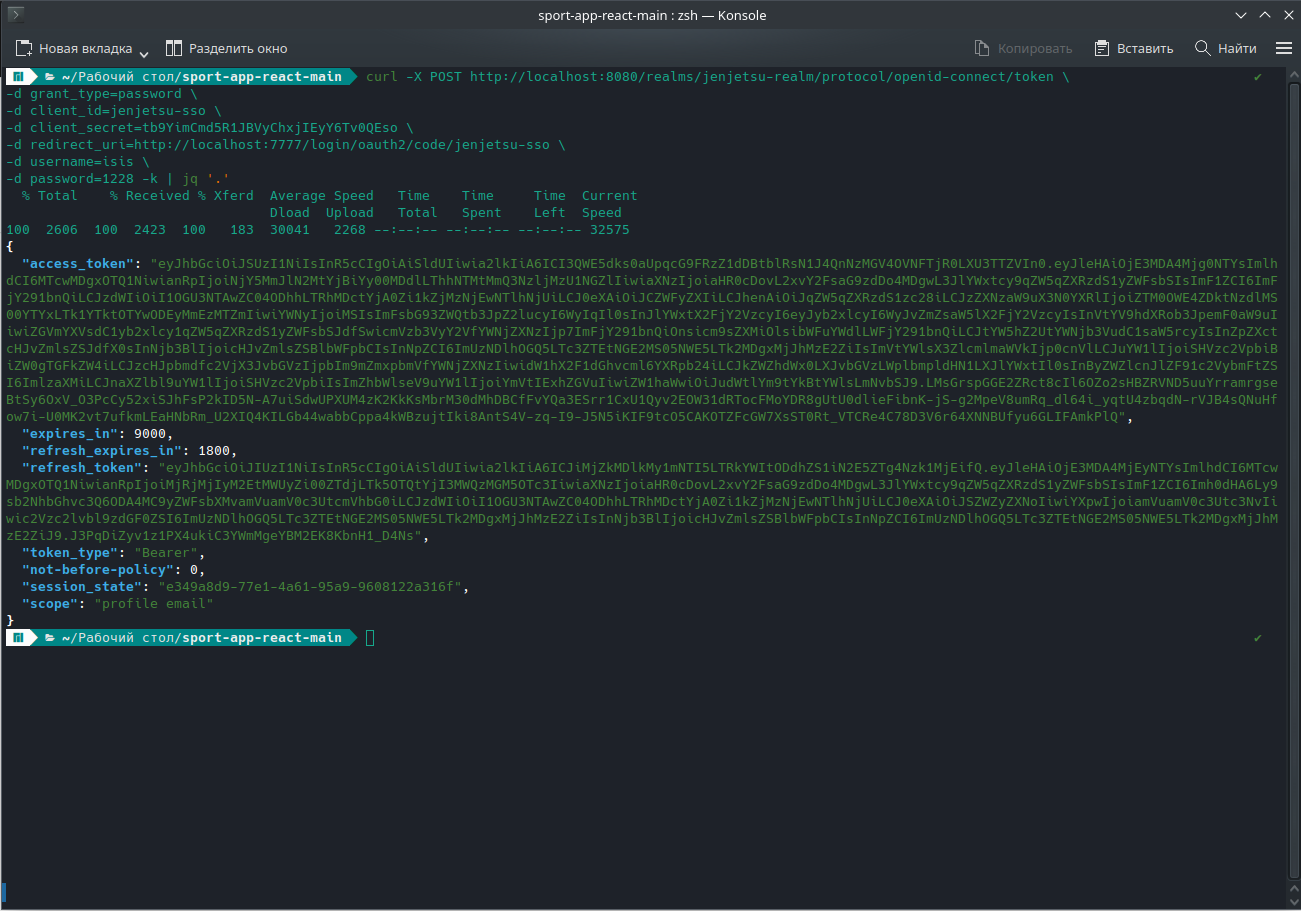
Crontab



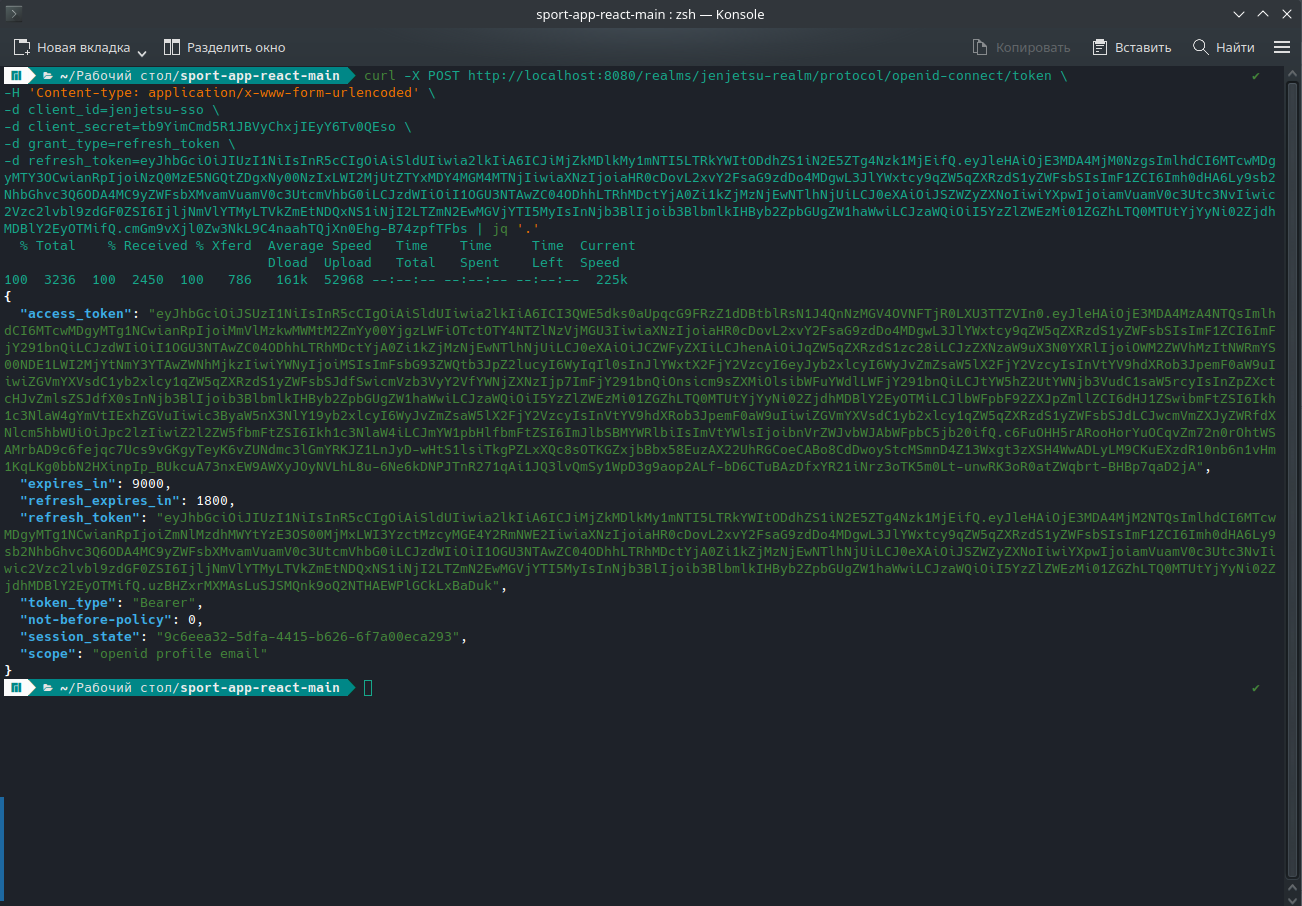
Шаг 6. Checking endpoints

Всё выполнялось через curl, вывод данных в красивый формат при помощи программы jq

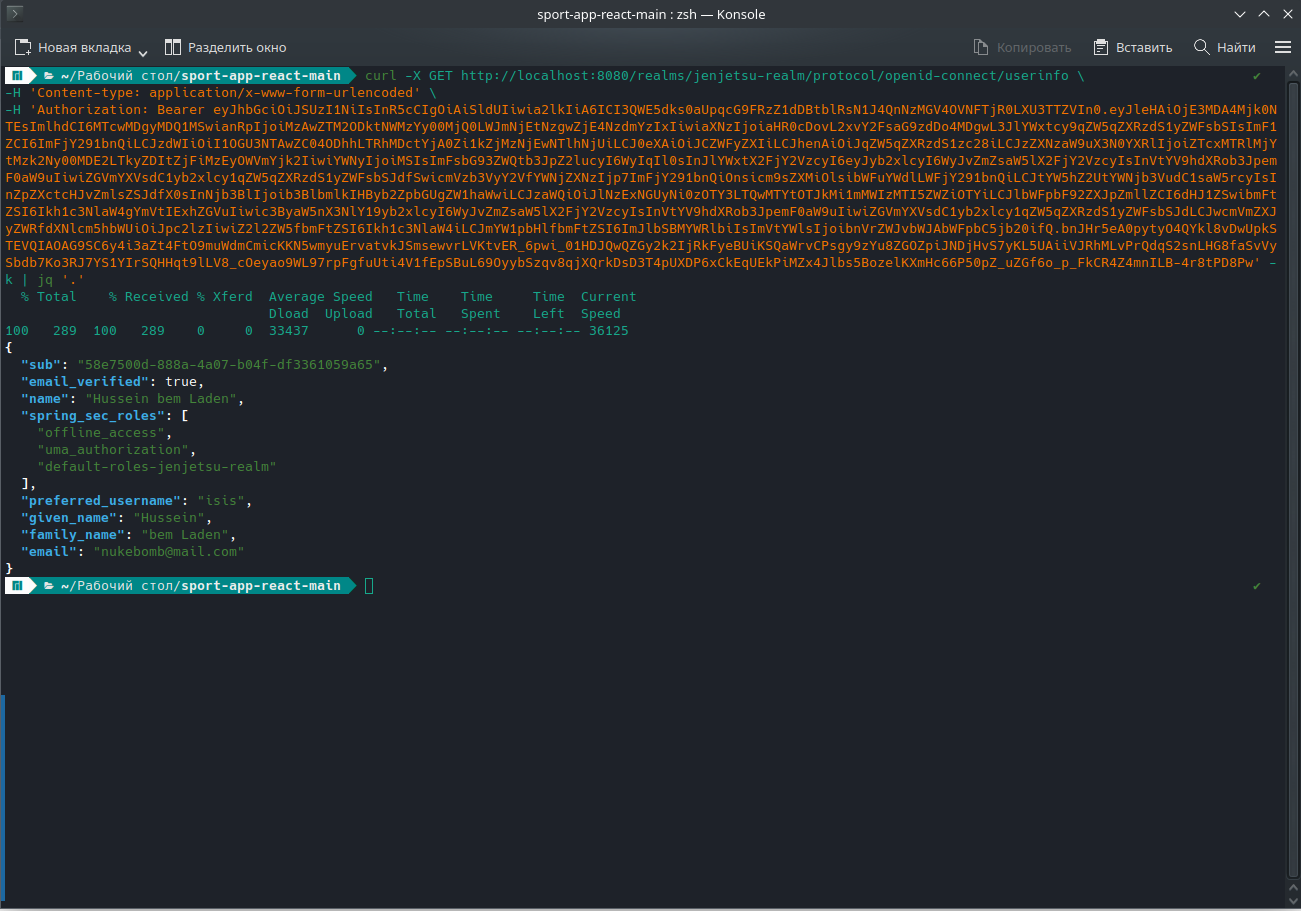
Получение access token через password authority



Получение access token через refresh token



Получение userinfo



Получение конфигурации .well-known

