Московский государственный технический университет им. Н.Э. Баумана

|  |  |
| --- | --- |
| Утверждаю: |  |
| Галкин В.А. | "\_\_"\_\_\_\_\_\_\_\_\_\_\_202 г. |

Курсовая работа по дисциплине

«Сетевые технологии»

«Сервер заказов с хранением видео»

руководство администратора

(вид документа)

писчая бумага

(вид носителя)

3

(количество листов)

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛИ: |  |
| студенты группы ИУ5-61 |  |
| Агличеев М.С. | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| Мартынова П.В | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  |

Москва-2022

1. Системные требования

Для функционирования клиента необходим ПК с браузером Microsoft Edge версии 99 и выше или Google Chrome. Системные требования аналогичны таковым у перечисленных браузеров

Для функционирования серверной части необходимы:

1. S3-сервер с как минимум 1 ТБ SSD-хранилища (при хранении большого количества видео может понадобиться больше памяти, оперативная память и CPU не имеют большого значения)
2. Сервер для БД с минимум 50 ГБ SSD, 3 ГБ ОЗУ, 2 процессорами частотой не менее 1.5 ГГц каждый.
3. Сервер для микросервисов с минимум 50 ГБ SSD, 5 ГБ ОЗУ, 2 процессорами частотой не менее 1.5 ГГц каждый.
4. Сервер для балансировки с минимум 20 ГБ SSD, 3 ГБ ОЗУ, 2 процессорами частотой не менее 1.5 ГГц каждый.

Все сервера работают на Ubuntu 22.04

1. Требования к программному обеспечению:
2. На сервере для микросервисов должен быть предустановлен Docker, Docker-Compose, Git
3. На сервере для балансировки должны быть установлены Docker, Docker-Compose, Git, Nginx, Npm и Certbot
4. Порядок развёртывания серверов:
5. S3-сервер

- Развернуть S3-хранилище, сделать доступ на чтение свободным для любых IP

1. Сервер БД

- Развернуть СУБД PostgreSQL, пользователь – Mikhail, пароль – thisismyhair

- Запустить скрипт из файла восстановления (<https://github.com/ChocolaterToba/Course_work_6_sem_backend/blob/master/postgres/Postgres_DB_Backup.sql>)

1. Сервер микросервисов

- Склонировать с помощью Git Backend-репозиторий (<https://github.com/ChocolaterToba/Course_work_6_sem_backend>)

- Перейти в папку server склонированного репозитория и создать в ней файл passwords.env следующего содержания:

AMAZON\_DB\_USER = mikhail

AMAZON\_DB\_PASSWORD = thisismyhair

LOCAL\_DB\_USER = postgres

LOCAL\_DB\_PASSWORD = ohhibitchitsme

VK\_CLIENT\_SECRET = ph98X9nBAXABdr17gQUY

- Запустить команду docker-compose -f docker-compose-no-routing up –build, в результате запустятся микросервисы бэкенда

1. Сервер роутинга

- Склонировать с помощью Git Backend-репозиторий (<https://github.com/ChocolaterToba/Course_work_6_sem_backend>)

- В папке server в файле docker\_vars.env заменить значения переменных DOCKER\_USER\_PREFIX, DOCKER\_AUTH\_PREFIX и DOCKER\_PRODUCT\_PREFIX на ip сервера микросервисов

- Запустить команду docker-compose -f docker-compose-only-routing up –build, в результате запустится сервис роутинга

- Очистить папку /etc/nginx/sites-enabled, скопировать в неё файл nginx.conf из Backend-репозитория, заменив в нём параметр server\_name на ваш URL

- Запустить команду service nginx restart, чтобы изменения sites-enabled вступили в силу

- Склонировать Frontend-репозиторий

- Перейти в папку ugears/ugears внутри Frontend-репозитория

- Запустить команду npm install

- Запустить команду npm run build

- Скопировать содержимое появившейся папки build в папку /var/www/gears4us/html

- В результате, при переходе по вашему URL по порту 80 Nginx отдаст файлы Frontend, который будет обращаться к Backend-роутеру по порту 8080