|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №5**

по дисциплине «Разработка клиент-серверных приложений»

**Студент группы** ИКБО-20-19 Анваржонов Ж. Т

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Зарипов Е. В.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

**Создание децентрализированной распределённой системы**

**Задание:**

Разработать децентрализованную распределенную систему для хранения файлов с использованием Spring/Spring Boot.

Требуемый функционал системы: загрузка и выгрузка файлов различных

форматов.

Для демонстрации системы необходимо развернуть как минимум 4 экземпляра приложения при помощи Docker и продемонстрировать работоспособность при отключении каждого из 4х экземпляров, а также при

отказе любых двух экземпляров.

В случае невозможности реализации системы с использованием

предложенного стека технологий необходимо обосновать это и предложить

решение на собственном стеке.

**Выполнение задания**

Покажем методы для сохранения и извлечения всех файлов:

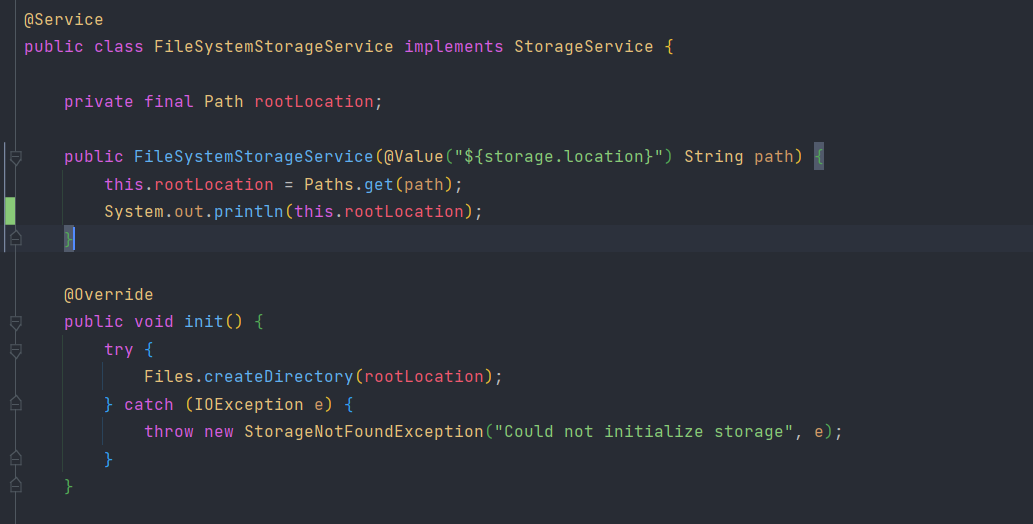


Рисунок 1 – сервис для сохранения и извлечения файлов



Рисунок 2 – метод для сохранения файлов по пути destinationFile



Рисунок 3 - методы показа всех файлов

В качестве демонстрации децентрализованной системы будем использовать docker-compose с nginx и 4 репликами нашего сервиса. Nginx будет управлять маршрутизацией между репликами приложений и если упадет один или несколько экземпляров, запросы будут идти к работающему. Как видно по рисункам 4-5 nginx будет слушать на порту 9999 на нашей машине и 80 – порт в контейнере

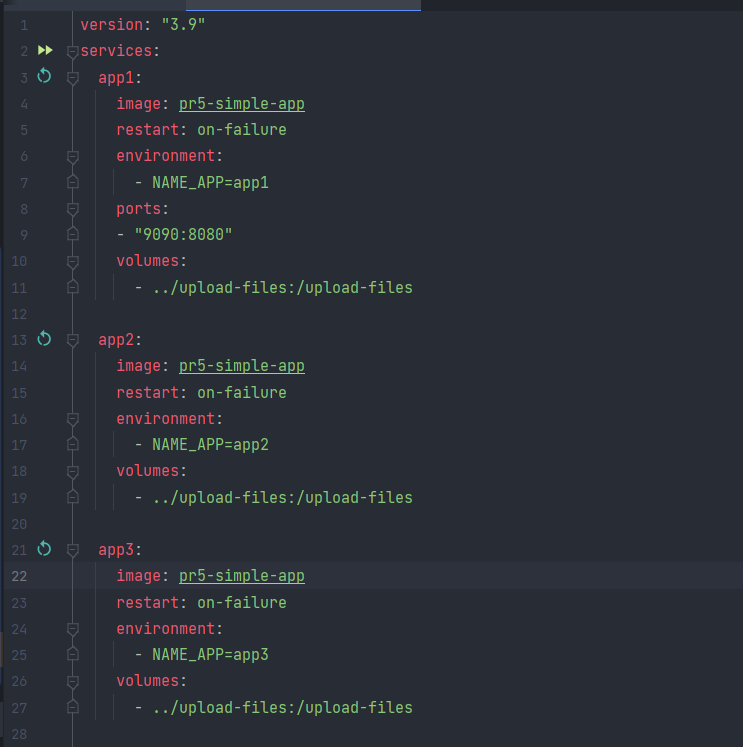


Рисунок 4 – первая часть docker-compose

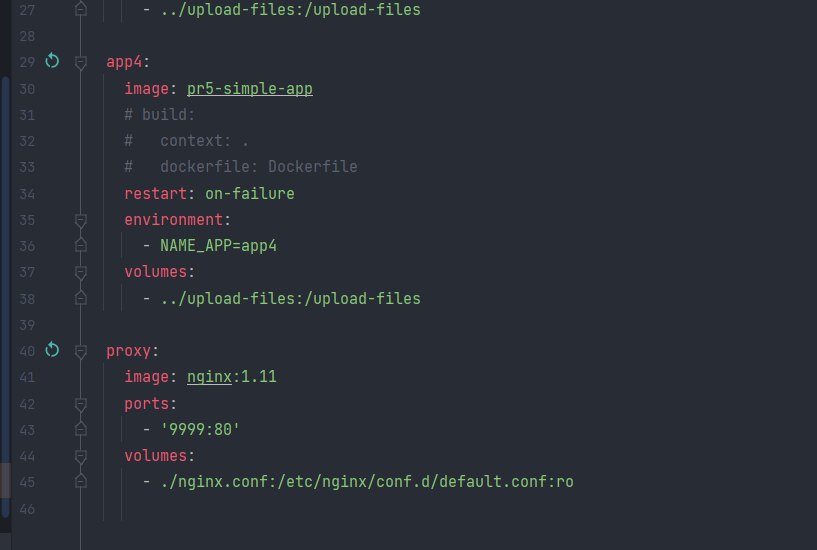


Рисунок 5 – вторая часть docker-compose

Покажем конфигурацию nginx:



Рисунок 6 – конфигурация nginx

Запустим docker-compose и продемонстрируем работоспобность приложения:

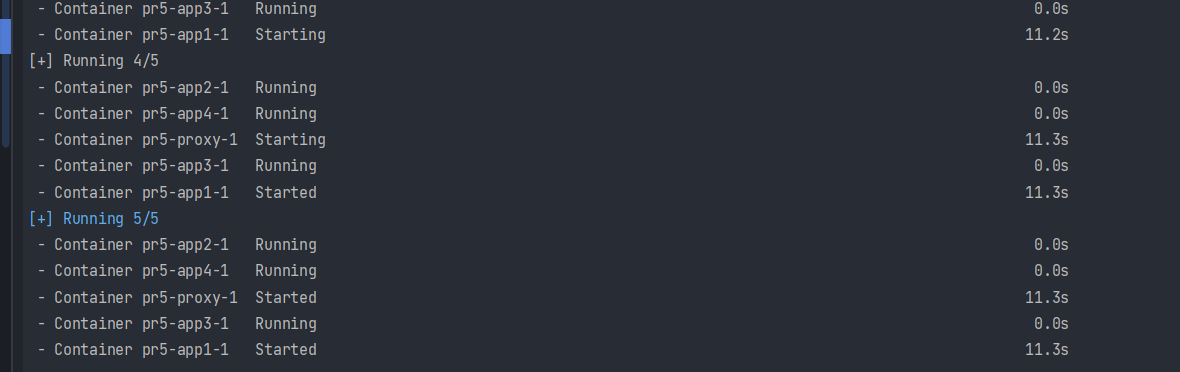


Рисунок 7 – демонстрация запуска всех контейнеров

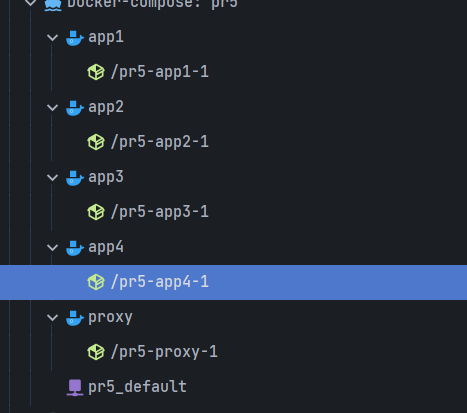


Рисунок 8 – запущенные контейнеры

Попробуем загрузить файл:

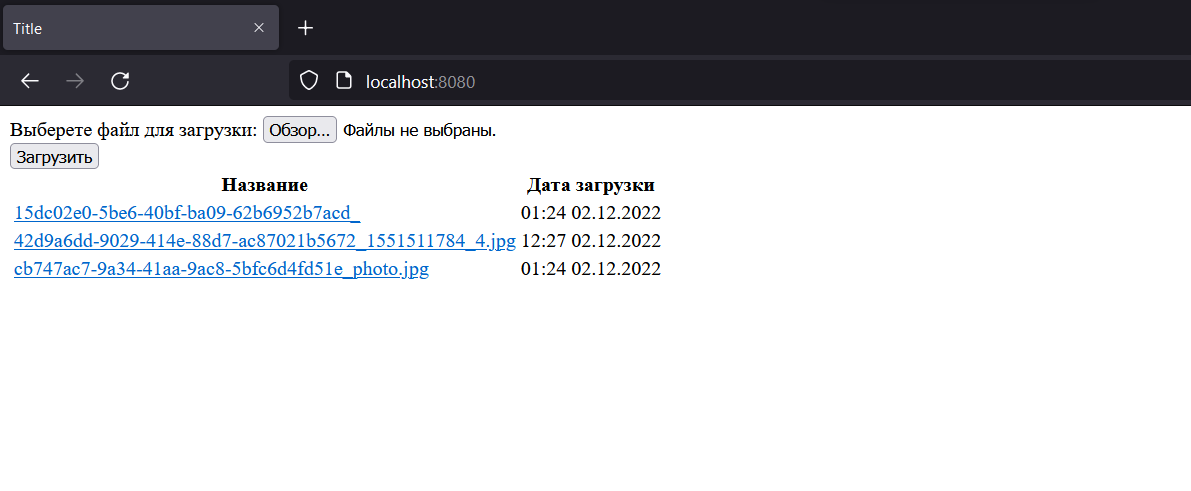


Рисунок 9 –загрузка файла (часть 1)

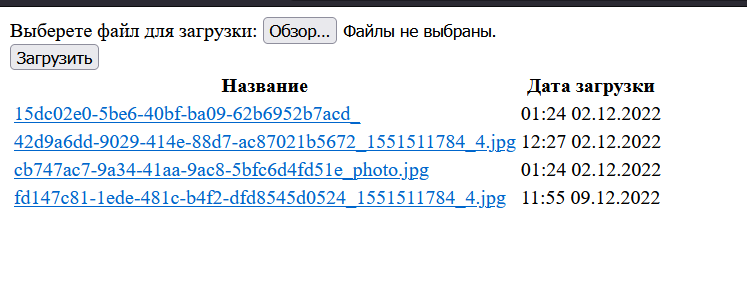


Рисунок 9 –загрузка файла (часть 2)

Попробуем отключить два экземпляра и еще раз загрузить файл:



Рисунок 10 – отключение экземпляров приложения

На рисунке ниже видно, что файл загрузился корректно, даже при двух отключенных экземплярах

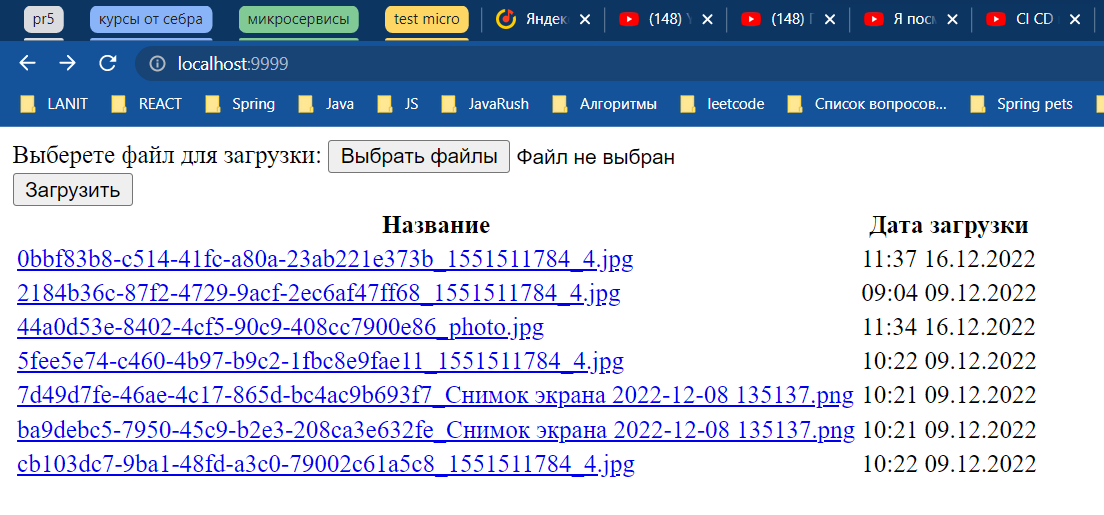


Рисунок 11 – загрузка файла при двух отключенных экземплярах

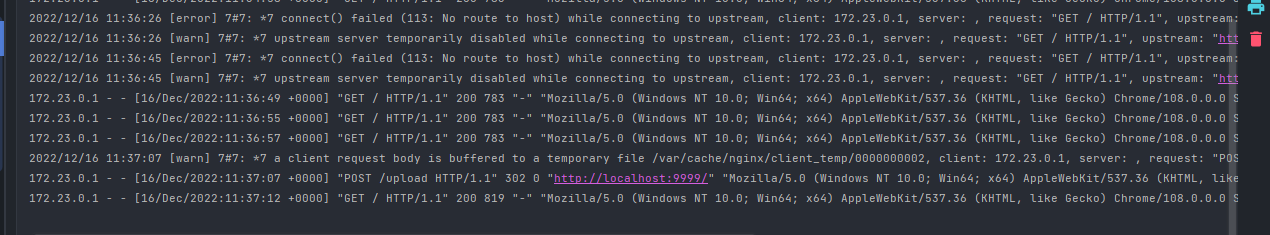


Рисунок 12 – логи nginx

**Вывод:**

В ходе практической работы мы разработали децентрализованную распределенную систему для хранения файлов с использованием Spring/Spring Boot.