Департамент образования и науки города Москвы

Государственное автономное образовательное учреждение высшего образования города Москвы

«Московский городской педагогический университет»

Институт цифрового образования

Департамент информатики, управления и технологий

ДИСЦИПЛИНА:

Интеграция и развертывание программного обеспечения с помощью контейнеров

**Лабораторная работа 3.2**

**Развертывание приложения в Kubernetes**

Выполнила: Шведова С.С., группа: АДЭУ-211

Преподаватель: Босенко Т.М.

Москва

2025

**Цель работы:** освоить процесс развертывания приложения в Kubernetes с использованием Deployments и Services.

**Задачи:**

Создать Deployment для указанного приложения.

Создать Service для обеспечения доступа к приложению.

Проверить доступность приложения через созданный Service.

Выполнить индивидуальное задание.

**Вариант 13.** Разверните приложение на Express.js, использующее базу данных MySQL и Redis для сеансов, в Kubernetes. Создайте Deployment для Express.js, MySQL и Redis, а также Service для доступа к приложению

**Технический стек:**

Node.js и Express.js: фреймворк для создания серверного приложения.

MySQL: реляционная база данных.

Redis: хранилище данных в памяти для управления сессиями.

Kubernetes: платформа для управления контейнеризованными приложениями.

Docker: для упаковки приложения и его зависимостей.

Установка миникуб на рисунке 1.



Рисунок 1. Установка миникуб

Дерево проекта на рисунке 2.

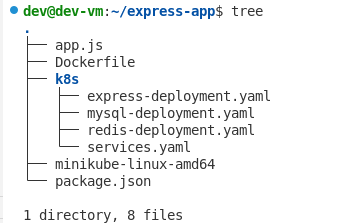


Рисунок 2. Дерево проекта

На рисунке 3 показан файл express-deployment.yaml

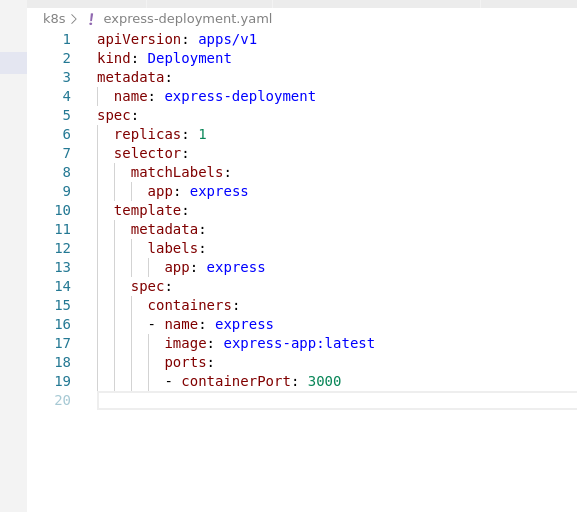


Рисунок 3. Файл express-deployment.yaml

На рисунке 4 сборка образа

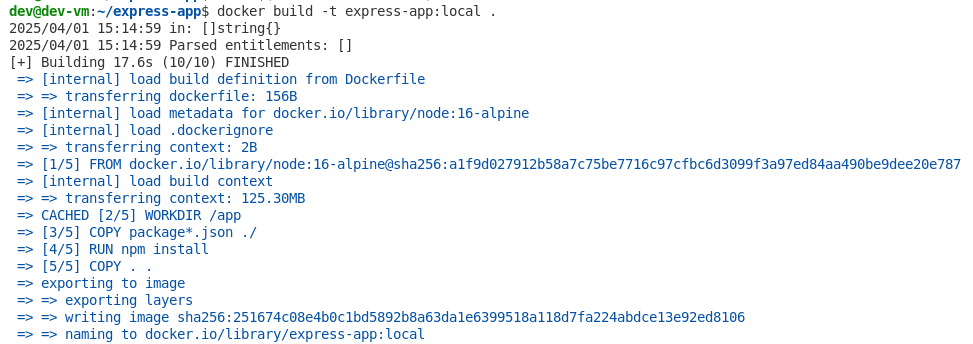


Рисунок 4. Сборка образа

На рисунке 5 показано применение конфигураций

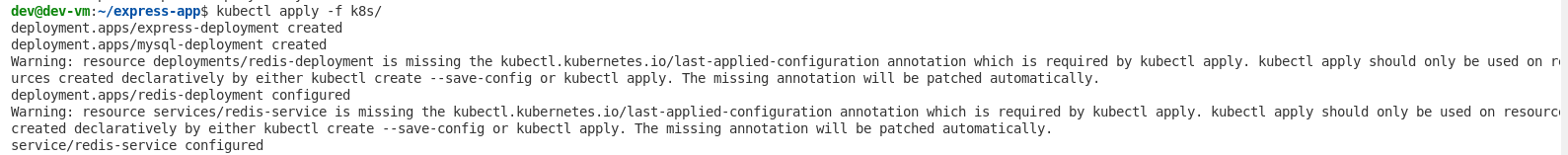


Рисунок 5. Конфигурации

На рисунке 6 показаны все сервисы

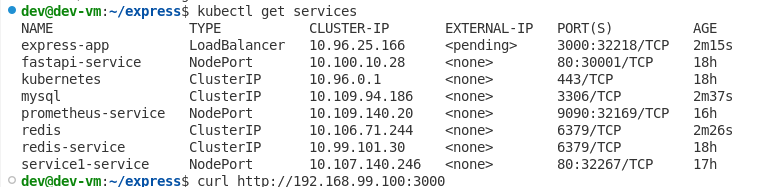


Рисунок 6. Сервисы

На рисунке 7 показана проверка статусов подов

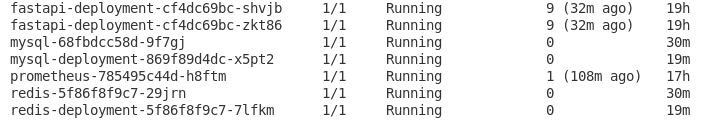


Рисунок 7. Проверка статусов подов

На рисунке 8 очистка ресурсов

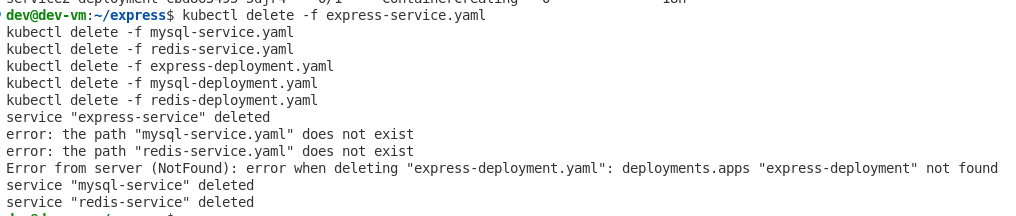


Рисунок 8. Очистка ресурсов

**Выводы:**

1. Создан Deployment для указанного приложения.

2. Создан Service для обеспечения доступа к приложению.

3. Проверена доступность приложения через созданный Service.

4. Выполнено индивидуальное задание.

**Контрольные вопросы**

**1.** Pod это основная единица развертывания в Kubernetes, представляющая собой один или несколько контейнеров с общими ресурсами

Deployment - это объект, управляющий состоянием Pods. Он обеспечивает автоматическое развертывание, обновление и управление жизненным циклом Pods.

Service - это объект, который определяет способ доступа к Pods, обеспечивая стабильный IP-адрес и DNS-имя, а также балансировку нагрузки.

**2. Назначение Deployment:** Обеспечить управление желаемым состоянием Pods, включая их создание, обновление и восстановление в случае сбоев.

**3. Назначение Service:** Обеспечить надежный доступ к Pods, скрывая их изменяемые IP-адреса и позволяя клиентам взаимодействовать через постоянную точку доступа.

**4. Создание Deployment: Используется команда:**

luaCopy code

kubectl create deployment <имя> --image=<имя\_образа>

Также можно создать Deployment через YAML-манифест.

**5. Создание Service: Используется команда:**

phpCopy code

kubectl expose deployment <имя\_deployment> --type=<тип> --port=<порт>

Типы Services:

1. ClusterIP: Доступен только внутри кластера.
2. NodePort: Доступен снаружи кластера через определенный порт на каждом узле.
3. LoadBalancer: Создает внешний балансировщик нагрузки (в облачных средах).
4. ExternalName: Позволяет ссылаться на внешние сервисы через DNS.