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

**Работа на лекции**

**Основы работы с Kubernetes**

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

Преподаватель: Босенко Т.М.

Москва

2025

**Цель работы**

Получить практические навыки работы с кластером Kubernetes, включая развертывание базовых компонентов, настройку мониторинга и работу с service mesh.

**Задачи**

1. Изучить основные концепции Kubernetes через практические вопросы.

2. Научиться анализировать и применять манифесты Kubernetes.

**3 вопроса:**

1. Как проверить, на какой рабочий узел были запланированы модули? Другими словами, как проверить, на каком узле запущен определённый модуль?

kubectl get pods -o wide

2. Как отредактировать развертывание?

kubectl edit deployment <DEPLOYMENT\_NAME>

3. Как удалить развертывание?

Один из способов - указать имя развертывания: kubectl delete deployment [deployment\_name]

Другой способ - использовать файл конфигурации развертывания: kubectl delete -f deployment.yaml

**Общее задание**

На рисунке 1 показано клонирование репозитория

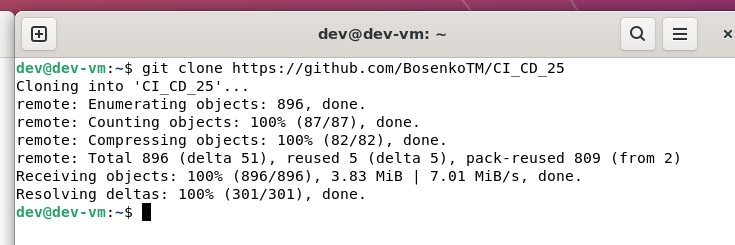
****

Рисунок 1. Клонирование репозитория

На рисунке 2 показана установка kubectl и миникуб

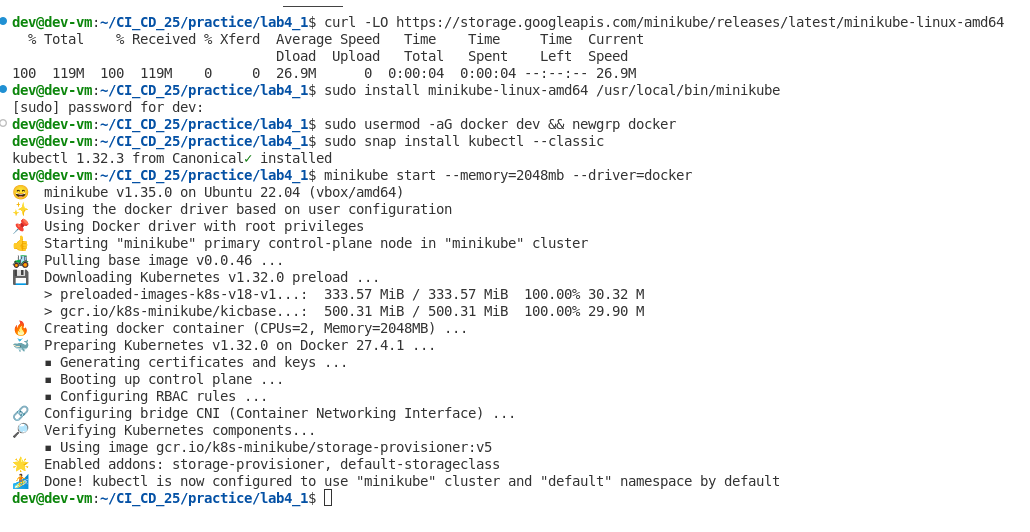
****

Рисунок 2. Установка kubectl и миникуб

На рисунке 3 показана настройка окружения командной строки Ubuntu для работы с Docker, который управляется Minikube, а также сборка образа



Рисунок 3. Сборка образа

На рисунке 4 показано создание манифестов

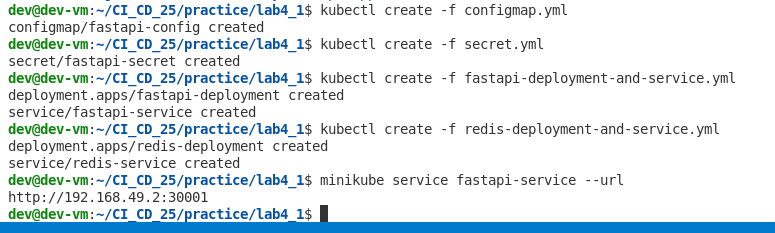
****

Рисунок 4. Сборка манифеста

На рисунке 5 показан OpenAPI

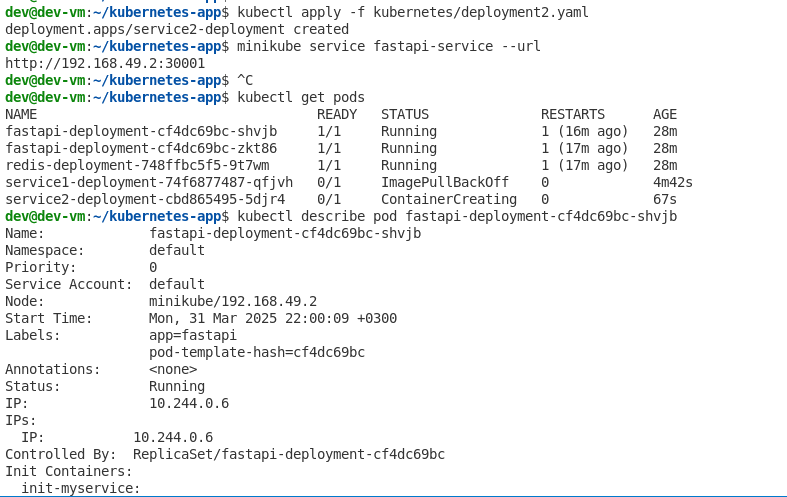


Рисунок 5. OpenAPI

На рисунке 6 показан работающий сайт FastAPI.

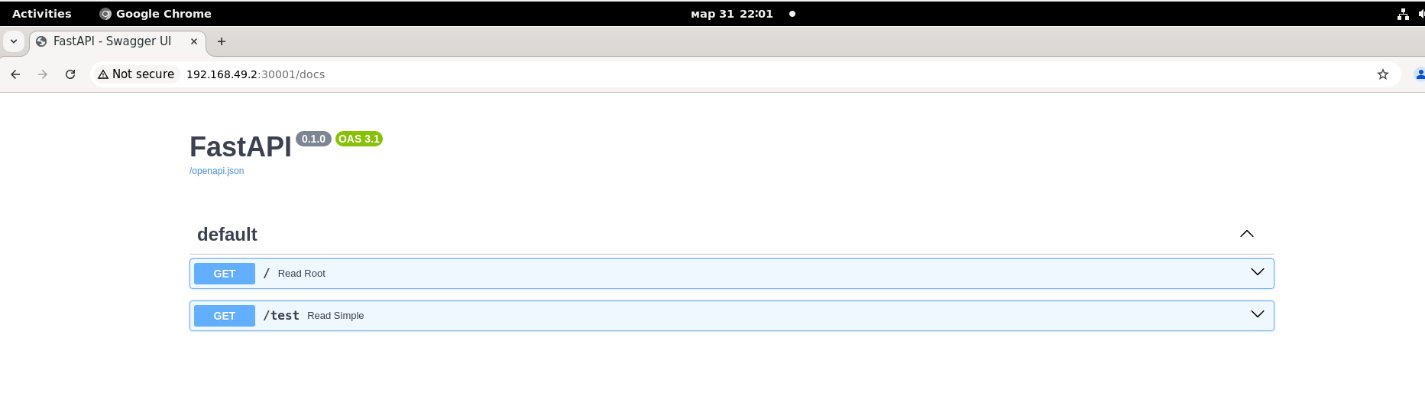
****

Рисунок 6. FastAPI

**Индивидуальное задание**

**Вариант 13. Запустите Kubernetes локально (k3s или minikube). Проверьте работу системных контейнеров и приложите скриншот команды: kubectl get po -n kube-system. Имеется YAML с деплоем для Prometheus. Измените файл:**

**– Настройте запуск без пароля;**

**– Фиксируйте образ на версии 2.30;**

**– Добавьте Service для доступа к метрикам.**

**Приложите итоговый YAML.**

**Напишите команды kubectl для контейнера:**

**– Выполнить команду ps aux внутри pod;**

**– Просмотреть логи за 5 минут;**

**– Удалить pod;**

**– Пробросить порт для отладки.**

Дерево проекта выглядит следующим образом (рисунок 7).

****

Рисунок 7. Дерево проекта

**Технический стек программы**

1. Клиентская часть

HTTP-запросы. Взаимодействие с сервером через RestAPI

2. Серверная часть

Prometheus Server. Он использует HTTP запросы для получения данных и поддерживает конфигурацию через YAML файлы

3. Хранилище данных

Prometheus. Использует собственное хранилище данных, основанное на временных рядах (TSDB)

4. Контейнеризация

Docker. Для упаковки Prometheus приложения и его зависимости в контейнер

5. Оркестрация

Kubernetes. Управление развертыванием и масштабированием контейнеров

Minikube. Локальный кластер Kubernetes для разработки и тестирования

6. Конфигурация

Конфигурационные файлы YAML

7. Мониторинг

Prometheus является инструментом мониторинга

**1 задание.** Проверка системных контейнеров на рисунке 8.

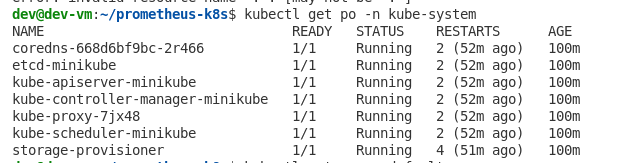
****

Рисунок 8. Проверка системных контейнеров

**2 задание.** Настроен запуск без пароля. Фиксируется образ на версии 2.30. Добавлен Service для доступа к метрикам (рисунок 9).

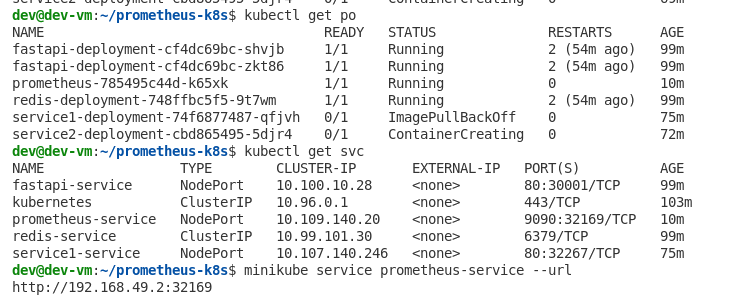
****

Рисунок 9. Запуск Prometheus

На рисунке 10 видно, что он запустился

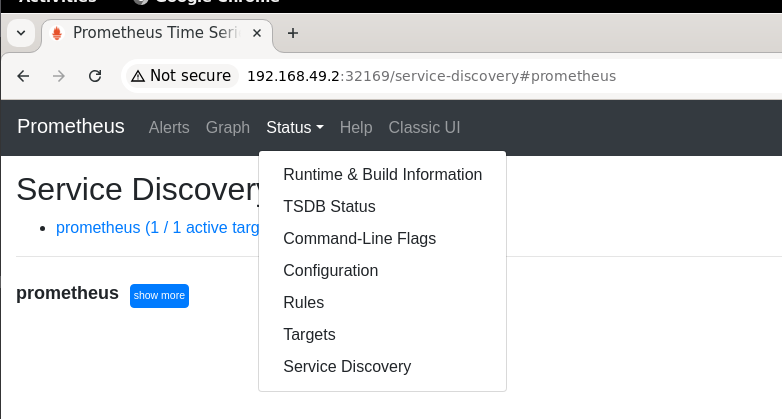
****

Рисунок 10. Prometheus запустился

**3 задание.** Выполнение команды ps aux внутри pod на рисунке 11, удаление pod

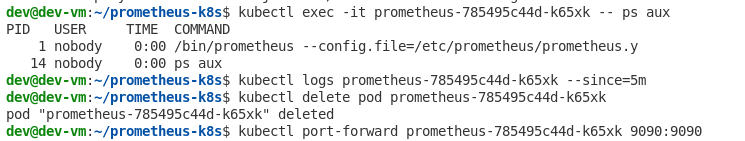
****

Рисунок 11. Выполнение команды ps aux внутри pod, удаление pod

На рисунке 12 показан вывод логов за последние 5 минут.

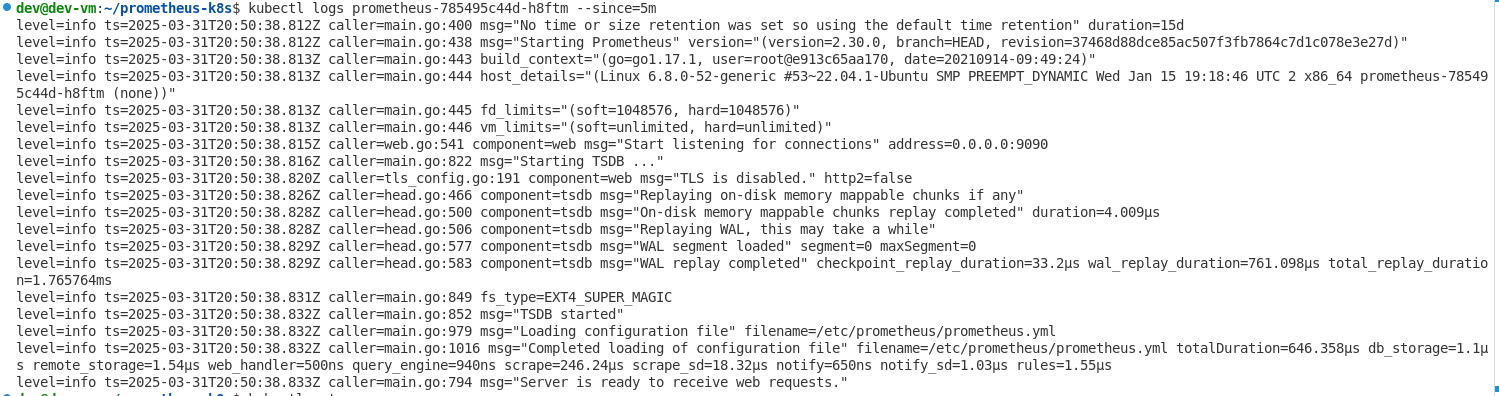
****

Рисунок 12. Вывод логов за последние 12 минут.

Пробросить порт для отладки на рисунке 13.

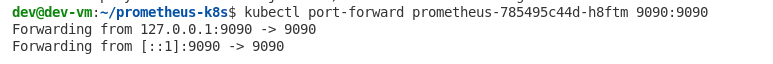
****

Рисунок 13. Порт для отладки

**Выводы:**

1. Изучены основные концепции Kubernetes через практические вопросы.

2. Были проанализированы и применены манифесты Kubernetes.