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:

Проектный практикум по разработке ETL-решений

**Вебинар 29.03.2025**

**Основы работы с Kubernetes**

Выполнила: Сергеева А. И., 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:**

- Изучить основные концепции Kubernetes через практические вопросы.

- Научиться анализировать и применять манифесты Kubernetes.

**Ход работы:**

Для начала необходимо было дать ответы на вопросы, а именно:

1. В каком пространстве имен работает модуль etcd? Перечислите модули в этом пространстве имен. **Ответ:** Модуль etcd работает в пространстве имён kube-system. Основные модули в kube-system: etcd-<node-name> – хранилище ключ-значение для данных кластера Kubernetes, kube-apiserver-<node-name> – API-сервер Kubernetes, kube-controller-manager-<node-name> – управляет контроллерами (ноды, реплики и т. д.), kube-scheduler-<node-name> – планировщик подов на ноды, coredns-<hash> – DNS-сервис для кластера, kube-proxy-<hash> – обеспечивает сетевую маршрутизацию и балансировку.

2. Предположим, у вас есть модуль под названием «nginx-test». Как его удалить? **Ответ:** k delete po nginx-test.

3. Как проверить, действителен ли манифест? **Ответ:** с флагом --dry-run, который не создаст файл, но протестирует его, и можно будет найти любые синтаксические ошибки. k create -f YAML\_FILE --dry-run

4. Что такое Kubernetes? Почему организации используют его? **Ответ:** Kubernetes — это система с открытым исходным кодом, которая позволяет пользователям управлять, масштабировать и развёртывать контейнерные приложения. Он заменяет ручное управление сотнями контейнеров, самостоятельно перезапускает упавшие сервисы, распределяет нагрузку, работает на любом облаке и локально.

5. Когда и почему НЕ следует использовать Kubernetes? **Ответ:** он не нужен для простых монолитных приложений, если осуществляется управление архитектурой низкого уровня или «голым железом».

6. Что такое kubectl? **Ответ:** Kubectl — это инструмент командной строки Kubernetes, который позволяет выполнять команды в кластерах Kubernetes. Например, с помощью kubectl можно развертывать приложения, проверять ресурсы кластера и управлять ими, а также просматривать журналы.

На рисунке 1 выполнена установка minikube.

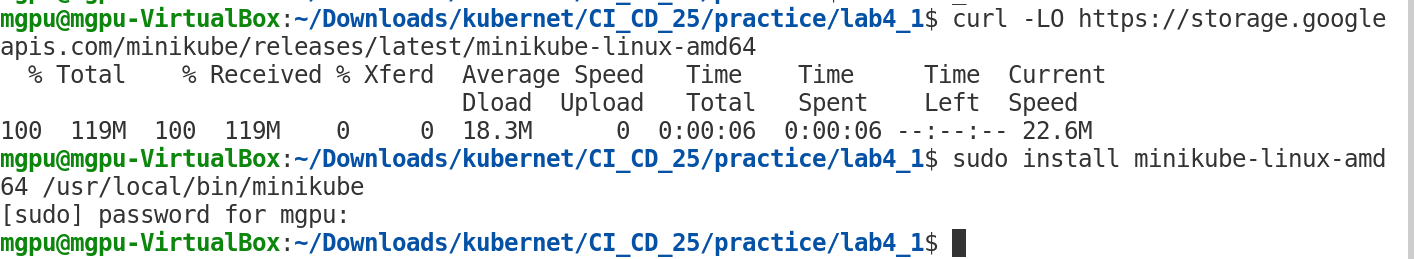


Рисунок 1 – Установка minikube

Добавление пользователя в группу Docker и установка kubectl реализованы на рисунке 2.

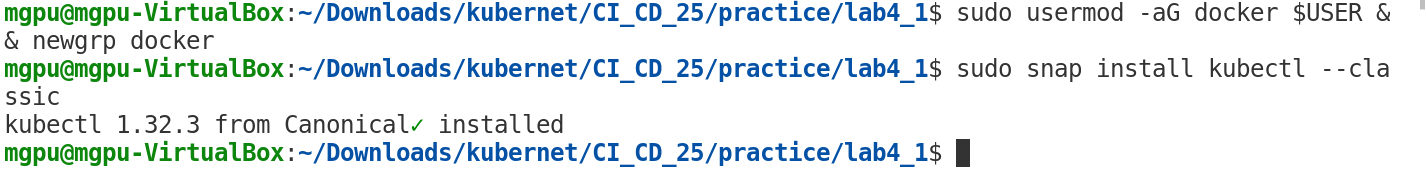


Рисунок 2 - Добавление пользователя в группу Docker и установка kubectl

Далее на рисунке 3 был осуществлён запуск и выбрано ограничение на 2 ГБ оперативной памяти, чтобы он не забрал больше.

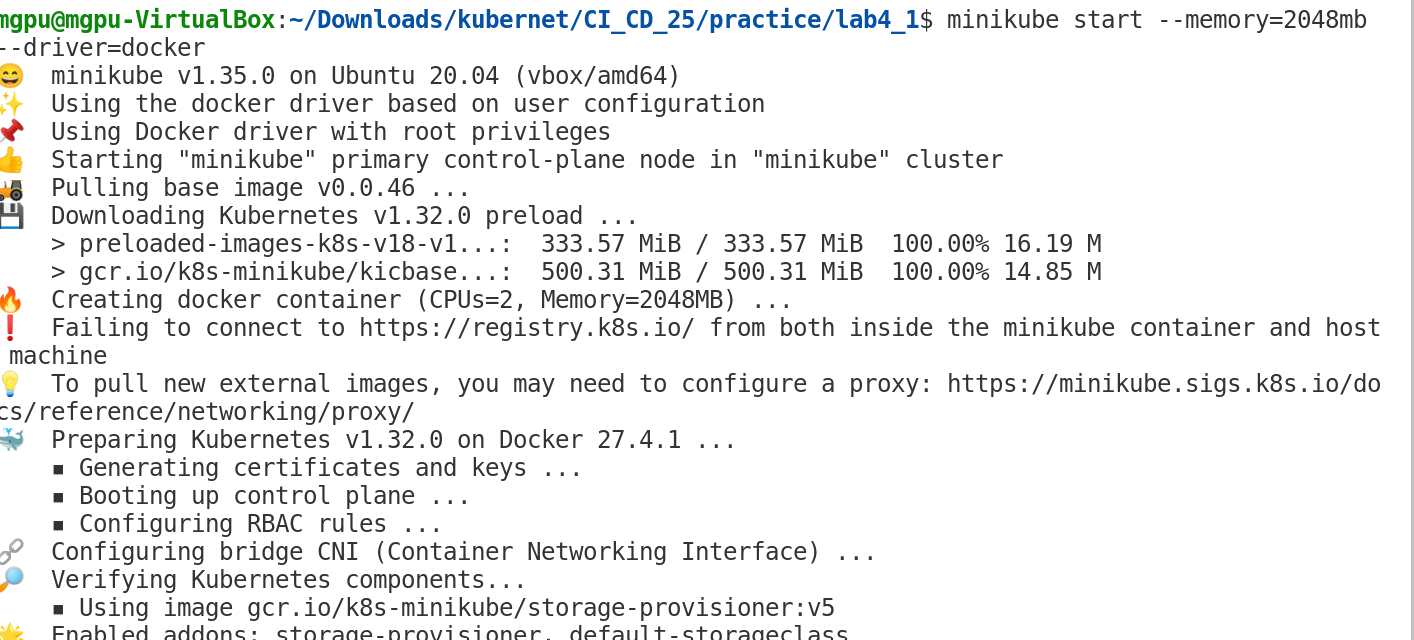


Рисунок 3 – Запуск minikube

Далее была выполнена настройка окружения и билдинг локального образа, загрузка в Minikube на рисунке 4.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 4 – Настройка окружения и билдинг локального образа

Были созданы ресурсы для кластера для хранения конфигурационных, чувствительных данных, управления pod'ами, разворачивания Redis на рисунке 5.

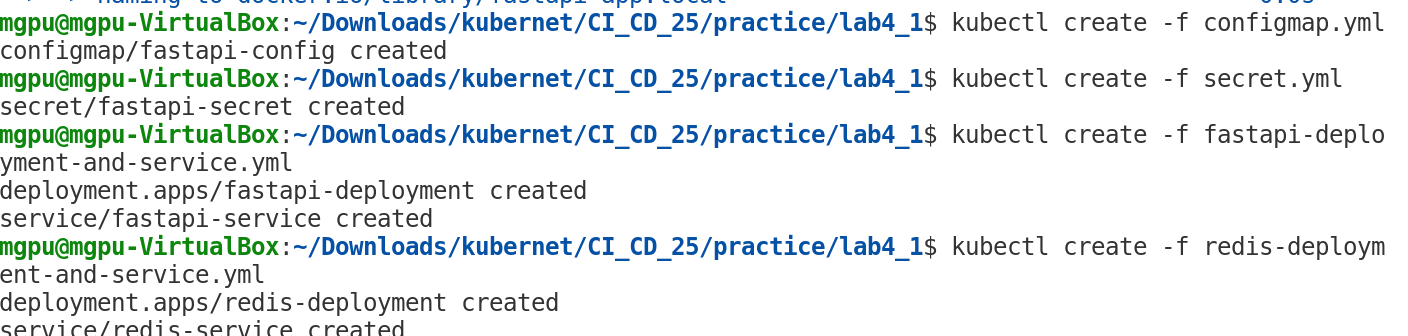


Рисунок 5 - Создание ресурсов в Kubernetes кластере

Далее было просмотрено состояние pod'ов в кластере на рисунке 6.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 6 - Состояние pod'ов в кластере Kubernetes

Далее был выведен URL-адрес сервиса FastAPI, чтобы можно было получить к нему доступ извне Minikube, как видно на рисунках 7–8.

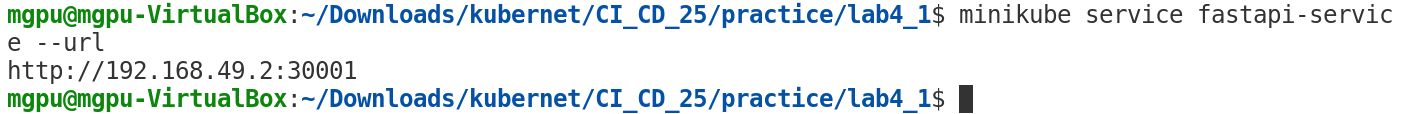


Рисунок 7 - Вывод URL-адреса сервиса FastAPI

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 8 - Проверка работоспособности FastAPI

Был просмотрена информация об одном из pod'ов, выведены список всех серверов, запущенных в кластере, и текущая конфигурация kubectl на рисунках 9–10.

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 9 – Информация об одном из pod'ов

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 10 - Список всех сервисов, запущенных в кластере

Для установки графического интерфейса были выполнены шаги, представленные на рисунках 11-17.

Изображение выглядит как текст, Шрифт, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 11 - Добавление репозитория helm, установка или обновление Dashboard

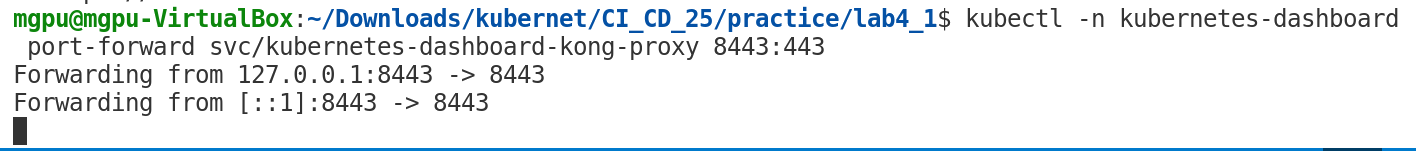


Рисунок 12 – Настройка доступа к Kubernetes Dashboard через HTTPS

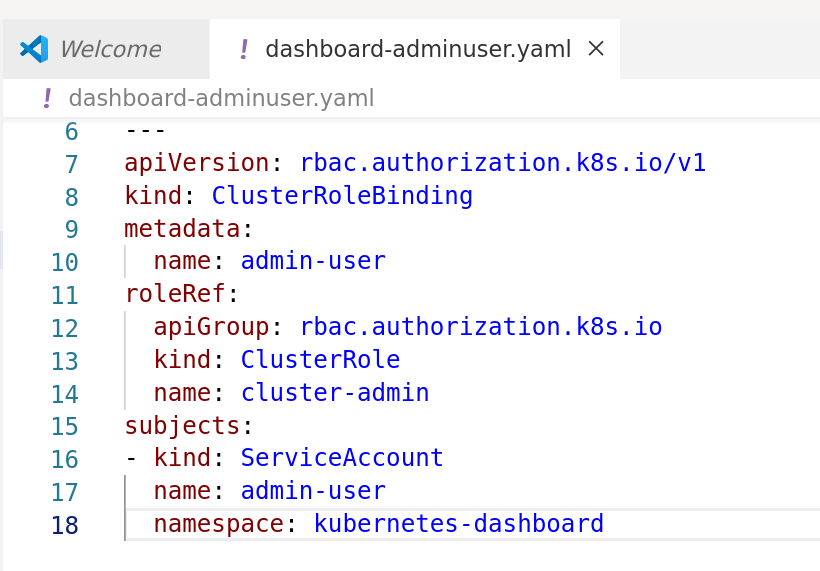


Рисунок 13 – Создание файл манифеста dashboard-adminuser.yaml

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 14 – Применение манифеста

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 15 – Создание долгосрочного токена

Изображение выглядит как текст, снимок экрана, Шрифт, число

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 16 – Получение токена

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 17 – Графический интерфейс

**Индивидуальное задание (12 вариант)**

Чтобы просмотреть все pod была выполнена команда на рисунке 18. Представленные системные pod очень важны: система доменных имён, распределённое хранилище, основной API, kube-controller-manager, сетевой прокси, планировщик, storage-provisioner. Все компоненты в состоянии Running — кластер работает корректно.

Изображение выглядит как текст, Шрифт, число, снимок экрана

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 18 – Список pod

Далее было необходимо в рамках 2-го задания изменить YAML с деплоем для Jenkins: настроить запуск без пароля (или с дефолтными настройками), зафиксировать образ на версии 2.319 и добавить Service для доступа к веб-интерфейсу. Версия была указана в image, в value был задан пароль admin по умолчанию, сервис доступен на порту 8080 внутри кластера и извне на порту 30000. Фрагмент YAML для развертывания Jenkins представлен на рисунке 19.

Изображение выглядит как текст, снимок экрана, дисплей, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 19 - Фрагмент YAML для развертывания Jenkins

Далее необходимо было применить конфигурацию на рисунке 20 и запустить на рисунках 21–22.

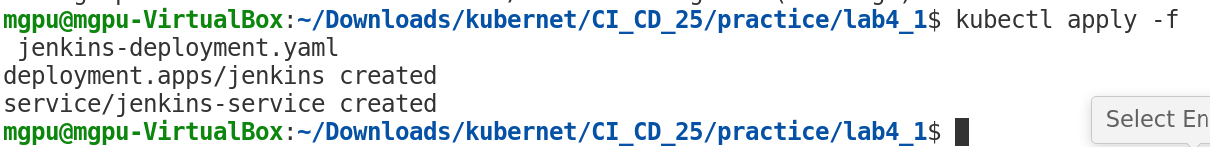


Рисунок 20 - Развёртывание Jenkins в кластере Kubernetes

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 21 – Доступ внутри кластера

Изображение выглядит как текст, снимок экрана, программное обеспечение, веб-страница

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 22 – Запуск Jenkins

Для пода FastAPI были просмотрены все запущенные процессы на рисунке 23. В контейнере работают 2 процесса: основной процесс FastAPI, потребляет 0.2% CPU и временный процесс, который был создан для выполнения команды ps aux.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 23 – Применение команды ps aux внутри pod

Просмотр логов за последние 5 минут выполнен на рисунке 24.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 24 – Просмотр логов за последние 5 минут

Далее был удалён под jenkins на рисунке 25.

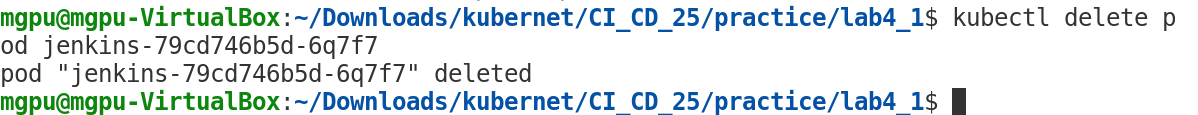


Рисунок 25 – Удаление pod jenkins

Под управляется Deployment, он автоматически пересоздастся, как видно на рисунке 26.

Изображение выглядит как текст, снимок экрана, Шрифт

Контент, сгенерированный ИИ, может содержать ошибки.

Рисунок 26 – Список pod

Проброс пода для отладки на локальный 9090 показан на рисунке 27.

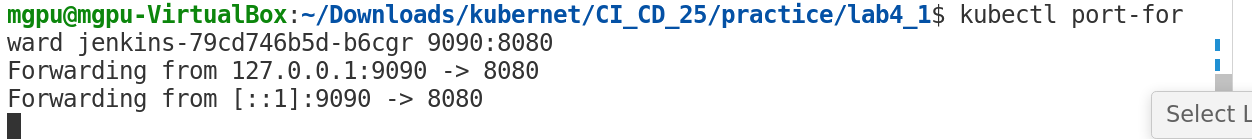


Рисунок 27 – Проброс pod для отладки

Общий вывод:

Kubernetes предоставляет мощный инструментарий для оркестрации контейнеров, включая управление развертываниями, сервисами, конфигурациями и хранилищами. А Minikube — удобный инструмент для локальной разработки и тестирования. В результате работы была ошибка ImagePullBackOff из-за отсутствия доступа к образу, для решения была чуть скорректирована версия jenkins. Также на примере был продемонстрирован перезапуск пода при удалении, как одно из преимуществ для повышения отказоустойчивости. Отсутствие ps/top в образах также одна из выявленных проблем, команда ps была применена только к FastAPI, т. к. версия jenkins не поддерживала данную команду. Рекомендация: выделять 4 ГБ памяти (minikube start --memory=4096), т. к. может использоваться вся оперативная память и при этом её все равно будет мало. При развертывании FastAPI + Redis пришлось проектировать: Service для Redis, Volume для Redis и зависимости. Цель и задачи были выполнены.