|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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 «Технологии виртуализации клиент-серверных приложений»

**Студент группы** ИКБО-20-21 Мухаметшин А. Р.

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

(подпись студента)

**Руководитель практической работы**  Волков М.Ю.

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

(подпись руководителя)

Работа представлена «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2024 г.

Москва 2024

**ОГЛАВЛЕНИЕ**

[Цель работы 3](#_Toc179033484)

[Ход работы 7](#_Toc179033485)

[Выводы 13](#_Toc179033486)

[Ответы на вопросы к практической работе 14](#_Toc179033487)

[Список использованной литературы 16](#_Toc179033488)

# Цель работы

Вам необходимо выполнить все указанные в задании пункты и отразить в отчете в формате снимков экрана.

Для начала работы необходимо установить и запустить minikube в соответствии с установленной ОС с официального сайта Kubernetes: https://kubernetes.io/ru/docs/tasks/tools/install-minikube/

Необходимо создать deployment при помощи файла deployment.yaml используя локальный docker образ с сервером:

* название deployment: Фамилия-НомерГруппы (ivanov-ikbo-99-99)
* используемый образ: Фамилия-НомерГруппы-Образ (ivanovikbo-99-99-obraz)



Рисунок 1 – Конфигурационные файлы

Необходимо посмотреть информацию о Deployment при помощи команды:



Рисунок 2 – Получения списка деплойментов

Далее необходимо посмотреть информацию о поде при помощи команды:

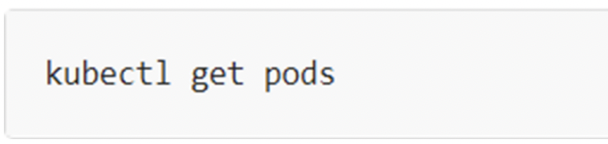


Рисунок 3 – Получения списка подов

После этого нужно посмотреть события кластера при помощи команды:

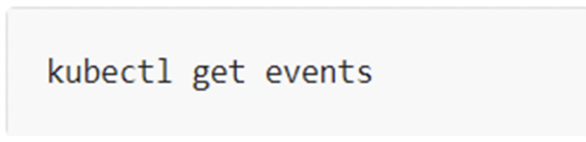


Рисунок 4 – Получение событий

Затем необходимо посмотреть kubectl конфигурацию при помощи команды:

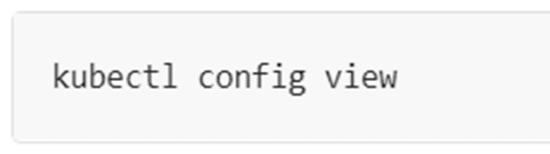


Рисунок 5 – Просмотр конфига

Потом нужно сделать под с deployment Фамилия-НомерГруппы доступным для публичной сети Интернет c помощью команды kubectl expose. После чего необходимо посмотреть только что созданный сервис

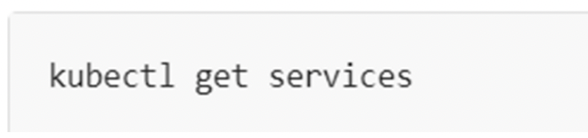


Рисунок 6 – Получения списка сервисов

Далее нужно запустить сервис hello-node:

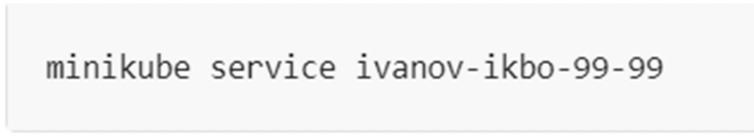


Рисунок 7 – Запуск сервиса

Затем требуется отобразить текущие поддерживаемые дополнения и включить дополнение, например ingress:

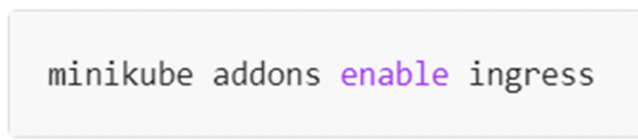


Рисунок 8 – Включение аддона

После этого нужно посмотреть Pod и Service, которые вы только что создали и отключить ingress.



Рисунок 9 – Выключение аддона

После отключения необходимо включить dashboard:



Рисунок 10 – Включение dashboard

Далее откройте во вкладке deployments созданный под и опишите в отчете отображаемые параметры.

После выполнения освободите ресурсы созданного вами кластера и остановите Minikube:

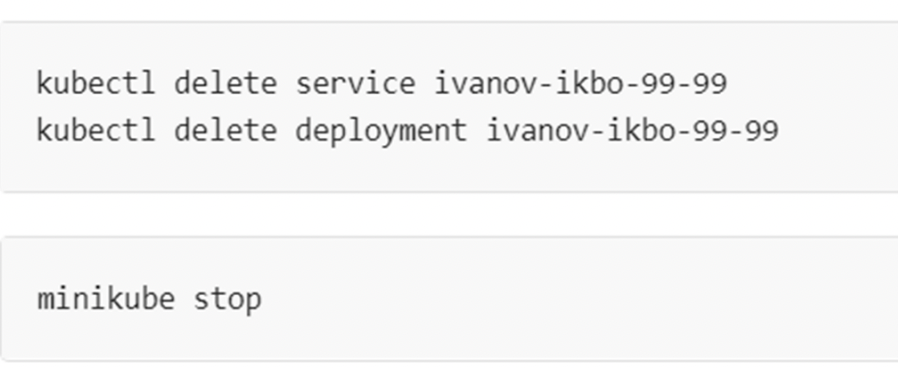


Рисунок 11 – Остановка кластера

# 

# Ход работы

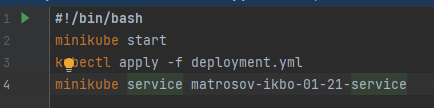


Рисунок 1 – Скрипт для запуска установки minikube

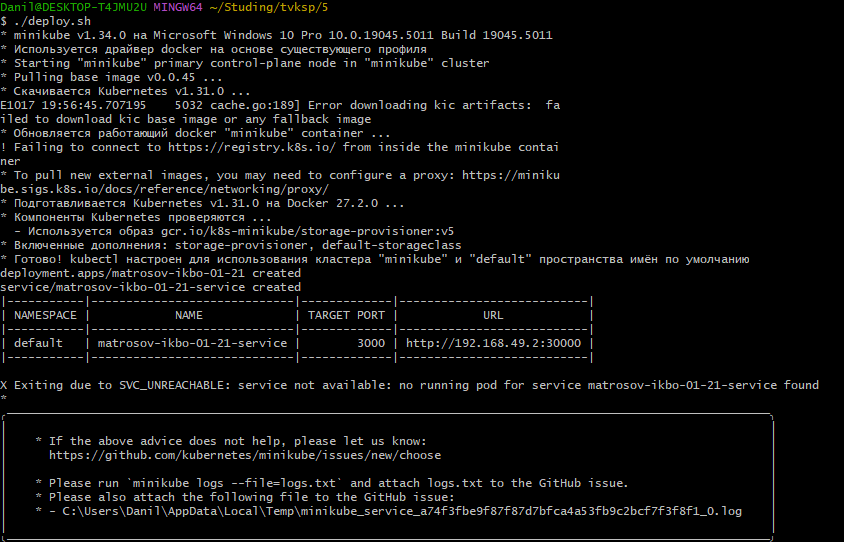


Рисунок 2 - Установка minikube

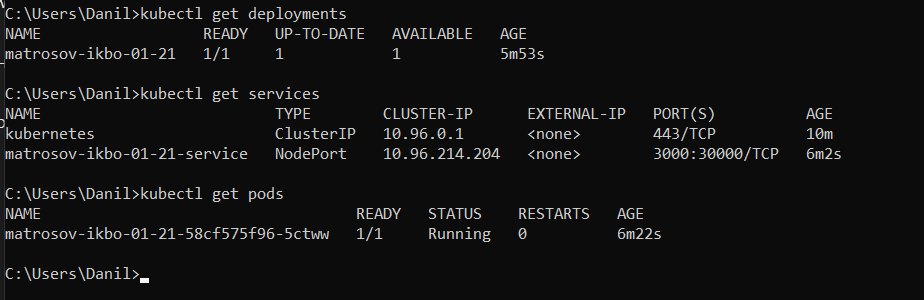


Рисунок 3 – Активные деплойменты и поды



Рисунок 4 – Конфигурационный файл deployment

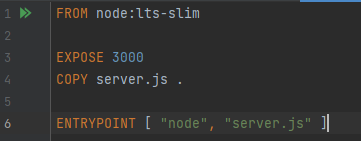


Рисунок 5 – Dockerfile для веб приложения



Рисунок 6 – Файл server.js

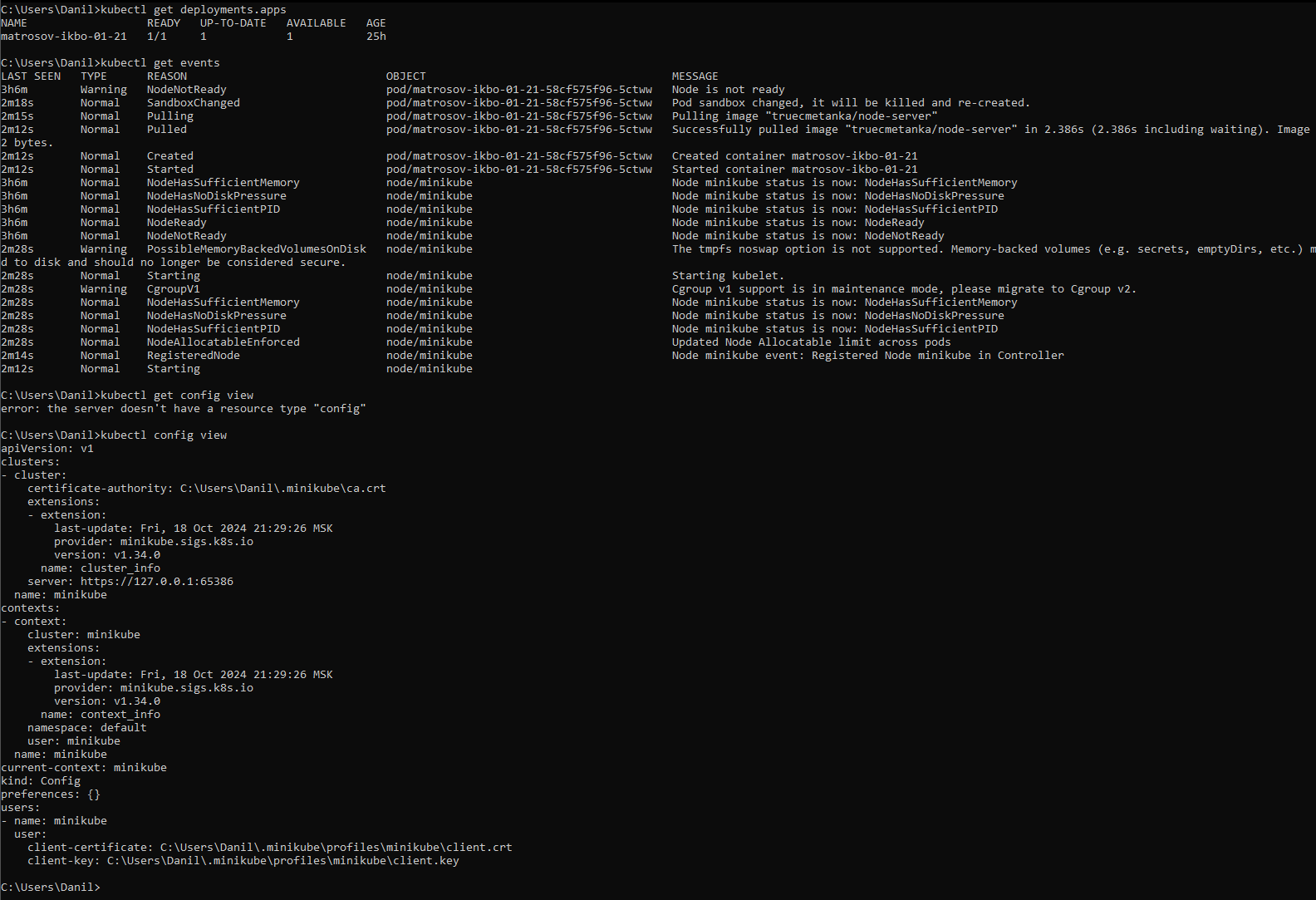


Рисунок 7 – Получение объектов кластера

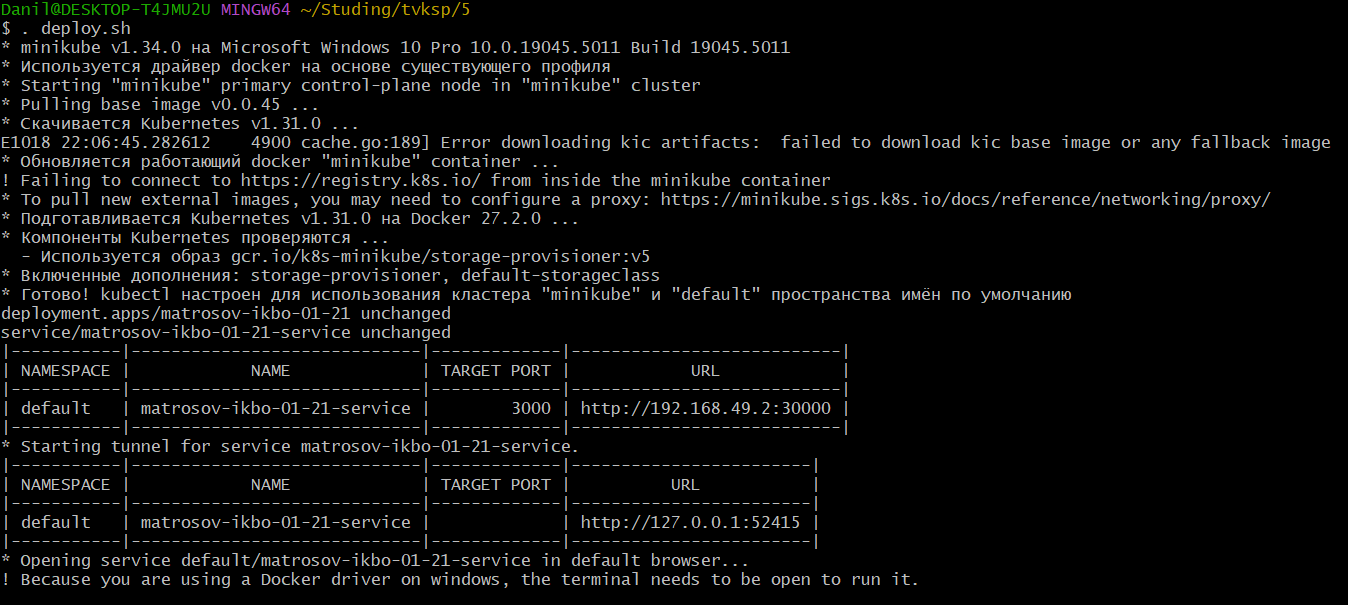


Рисунок 8 – Запуск сервиса

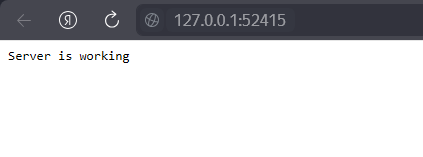


Рисунок 9 – Результат запроса

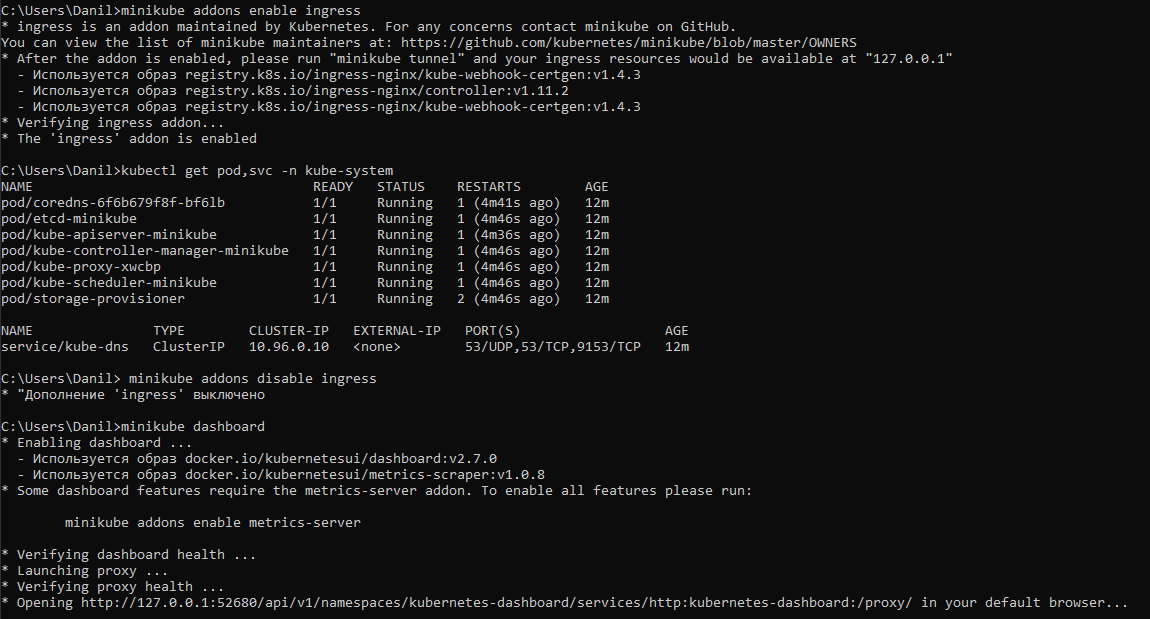


Рисунок 10 – Работа с аддонами

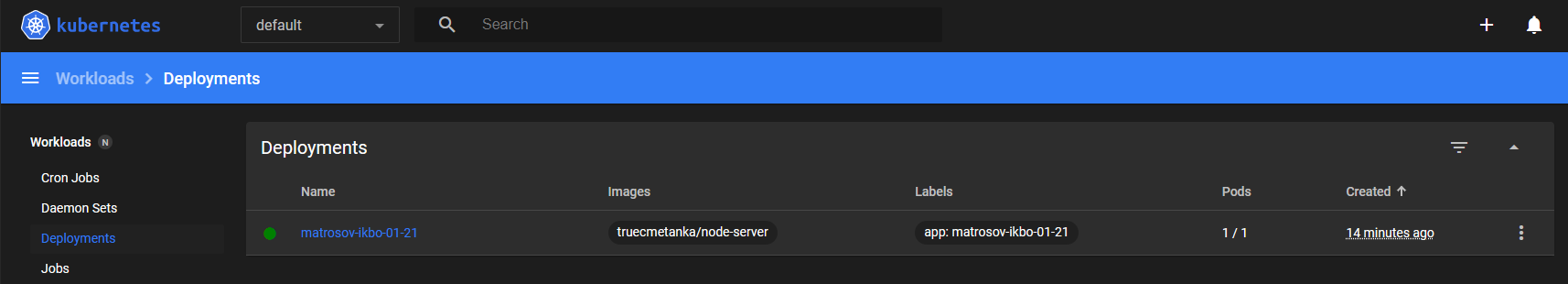


Рисунок 11 – Вывод dashboard

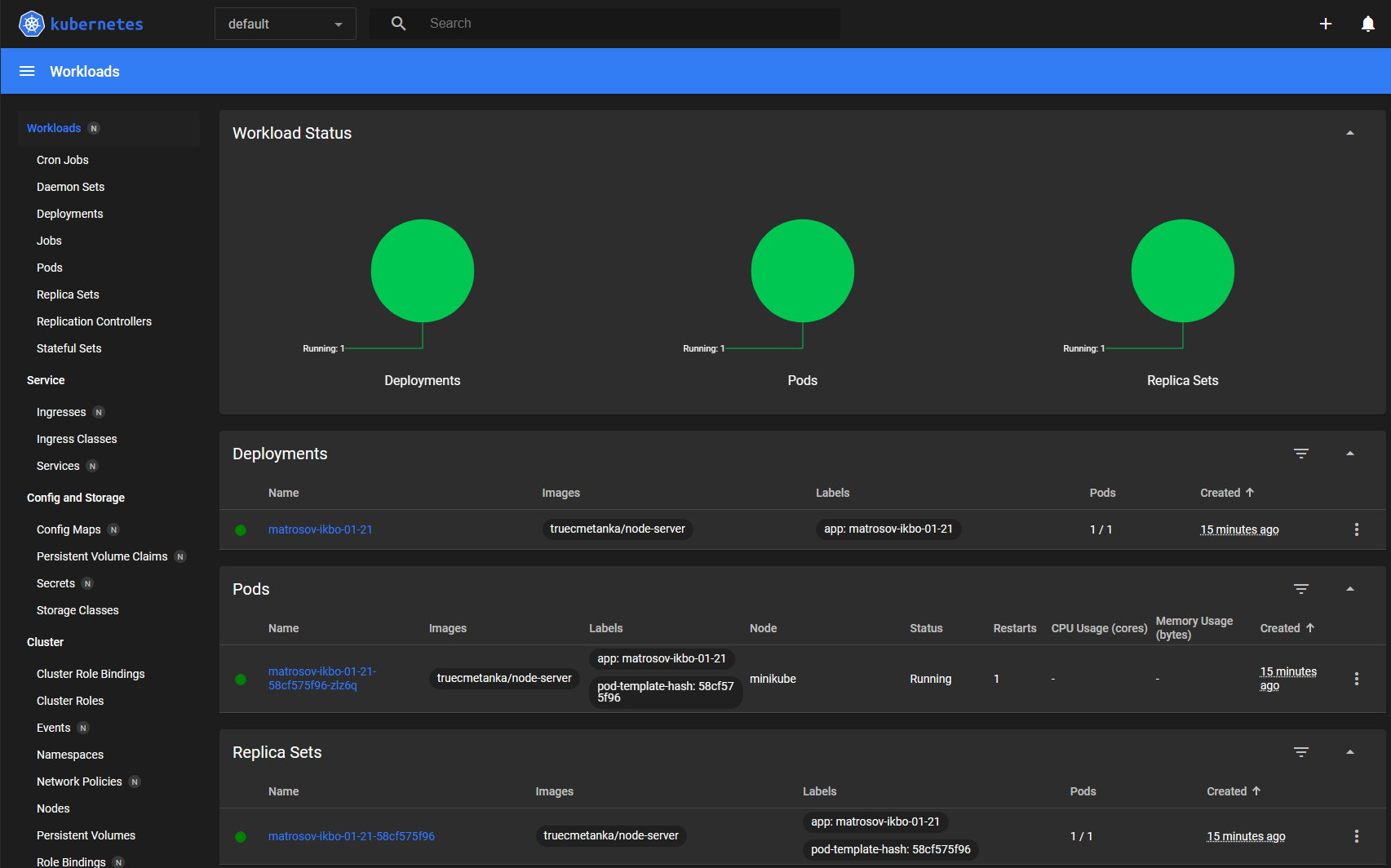


Рисунок 12 – Вывод dashboard

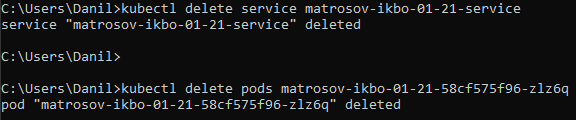


Рисунок 13 – Удаление объектов кластера

# Выводы

В результате выполнения данной практической работы, были получены навыки по работе с оркестровкой контейнеризированных приложений — автоматизации их развёртывания, масштабирования и координации в условиях кластера. А также были изучены на практике различные аддоны локального кластера Kubernetes minikube.

# Ответы на вопросы к практической работе

1. **Назовите виды контроллеров в Kubernetes.**

В Kubernetes существует несколько видов контроллеров:

* ReplicaSet контроллер для поддержания желаемого количества реплик подов
* Deployment контроллер для управления обновлением приложений и их конфигурацией
* StatefulSet контроллер для управления приложениями с состоянием DaemonSet контроллер для управления запуском по одной копии пода на каждом узле
* Job и CronJob контроллеры для выполнения задач и работы по расписанию

1. **Как называется командная строка в Kubernetes?**

Командная строка в Kubernetes называется kubectl. Она предоставляет интерфейс командной строки для взаимодействия с Kubernetes кластером.

1. **Что такое под?**

Под (Pod) в Kubernetes представляет собой минимальную управляемую единицу, которая объединяет один или несколько контейнеров, хранилище для этих контейнеров, уникальный IP-адрес, а также настройки для запуска в контексте узла (например, монтаж томов и переменных среды).

1. **Назовите 2 типа ресурсов, из которых состоит кластер Kubernetes.**

Кластер Kubernetes состоит из различных ресурсов, включая:

Node (узел): Физический или виртуальный компьютер, на котором работают поды

Pod (под): Группа одного или нескольких контейнеров, совместно использующих ресурсы и сетевое пространство

1. **Чем Kubernetes отличается от Docker Swarm?**

Основные различия между Kubernetes и Docker Swarm: Kubernetes - это оркестратор контейнеров, который управляет развертыванием, масштабированием и управлением контейнеризированными приложениями в крупных средах с учетом автоматического восстановления и масштабируемости. Docker Swarm, с другой стороны, также является оркестратором контейнеров, но является частью Docker, обеспечивая более простое развертывание и управление контейнерами в небольших и средних масштабах. Kubernetes широко используется в распределенных узлах и предоставляет более мощные функции оркестрации, тогда как Docker Swarm обеспечивает более простую настройку и управление для небольших проектов.

# Список использованной литературы

1. Virtualization and Containers: An Overview [Электронный ресурс]. – URL: https://www.example.com/virtualization-containers-overview (дата обращения: 06.10.2024)
2. Docker Documentation. [Электронный ресурс]. – URL: https://docs.docker.com/ (дата обращения: 06.10.2024)
3. VMware Official Website. [Электронный ресурс]. – URL: https://www.vmware.com/ (дата обращения: 06.10.2024)
4. "Virtual Machines Versus Containers: What's the Diff?" by John Smith. Virtualization Journal, vol. 25, no. 3, 2019, pp. 45-58.
5. "Understanding the Performance Trade-Offs in Virtualization" by Jane Doe. Proceedings of the International Conference on Cloud Computing, 2022, pp. 112-120.
6. 50 вопросов по Docker, которые задают на собеседованиях, и ответы на них | Хабр. — Текст: электронный [сайт]. — URL: https://habr.com/ru/company/southbridge/blog/528206/
7. Docker Documentation | Docker Documentation — Текст: электронный [сайт]. — URL: https://docs.docker.com/
8. Что такое режим Docker Swarm и когда его использовать? — CloudSavvy ИТ | Cpab. — Текст: электронный [сайт]. — URL: https://cpab.ru/chtotakoe-rezhim-docker-swarm-i-kogda-ego-ispo4lzovat-cloudsavvy-it/
9. Dockerfile reference | Docker Documentation — Текст: электронный [сайт]. — URL: https://docs.docker.com/engine/reference/builder/