|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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-20-19 Московка Артём Александрович

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

(подпись студента)

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

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

(подпись руководителя)

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

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

Москва 2022

# 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)

Необходимо посмотреть информацию о Deployment при помощи команды kubectl get deployments

Далее необходимо посмотреть информацию о поде при помощи команды kubectl get pods

После этого нужно посмотреть события кластера при помощи команды kubectl get events

Затем необходимо посмотреть kubectl конфигурацию при помощи команды: kubectl config view

Потом нужно сделать под с deployment Фамилия-НомерГруппы доступным для публичной сети Интернет c помощью команды kubectl expose:

сервис должен быть виден вне кластера;

порт: 8080.

После чего необходимо посмотреть только что созданный сервис kubectl get services

Далее нужно запустить сервис hello-node: minikube service ivanov…

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

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

После отключения необходимо включить dashboard:

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

После выполнения освободите ресурсы созданного вами кластера и остановите Minikube.

# Ход работы

Для начала создадим свой Docker-образ, для этого создадим Dockerfile и Server.js

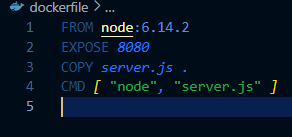


Рисунок 1 - dockerfile



Рисунок 2 – server.js

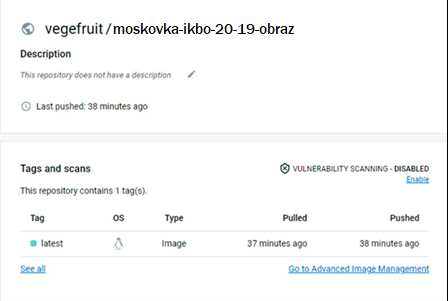


Рисунок 3 - Образ на Docker Hub

Запустим minikube.

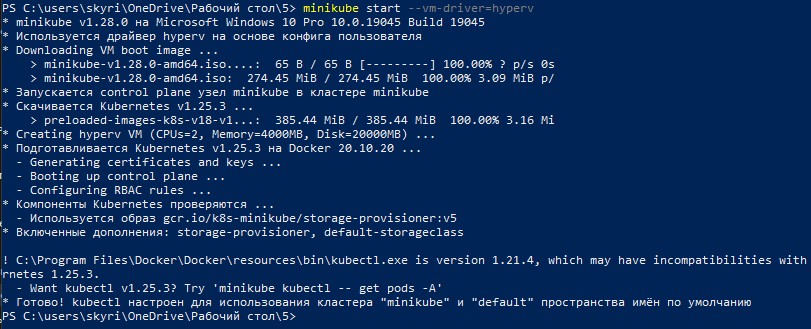


Рисунок 4 - Запуск minikube

Затем создадим yaml для деплоймента. (Рисунок 5)

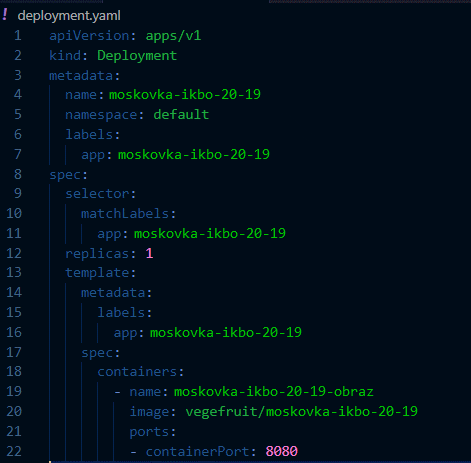


Рисунок 5 – Скриншот yaml файла развертывания приложения

Затем создадим деплоймент и выведем о нем и созданном поде информацию. также выведем список событий, конфигурацию minikube.

# 

Рисунок 6 - Деплоймент и информация о нем

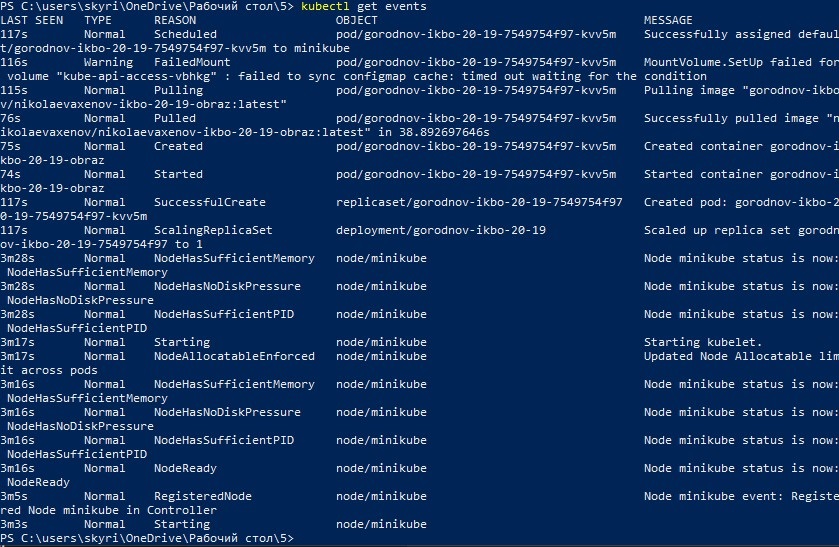


Рисунок 7 - Список событий

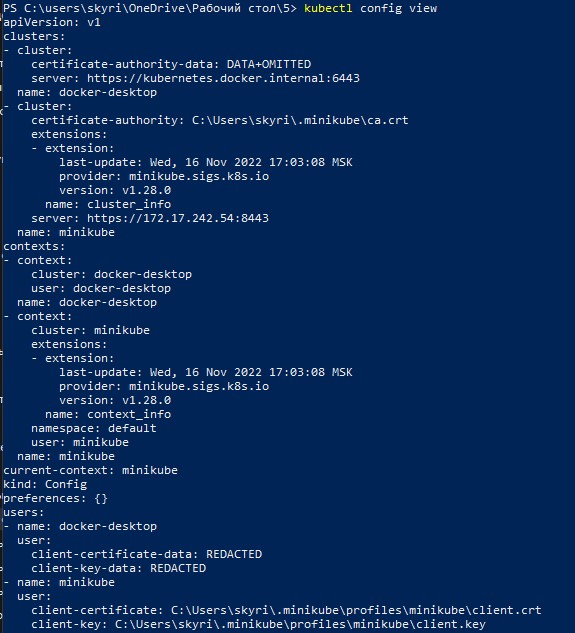


Рисунок 8 - Конфигурация minikube

Создадим и запустим сервис для деплоймента

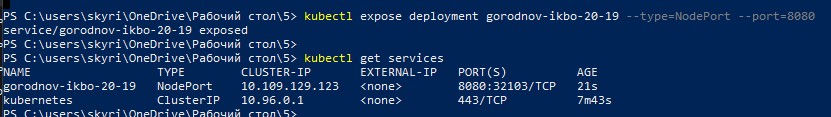


Рисунок 9 - Создание сервиса и просмотр запущенных сервисов

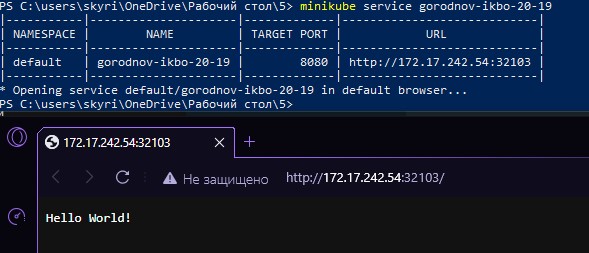


Рисунок 10 - Результат работы сервиса

Затем просмотрим список доступных дополнений и установим одно, Затем просмотрим запущенные сервисы и удалим дополнение.

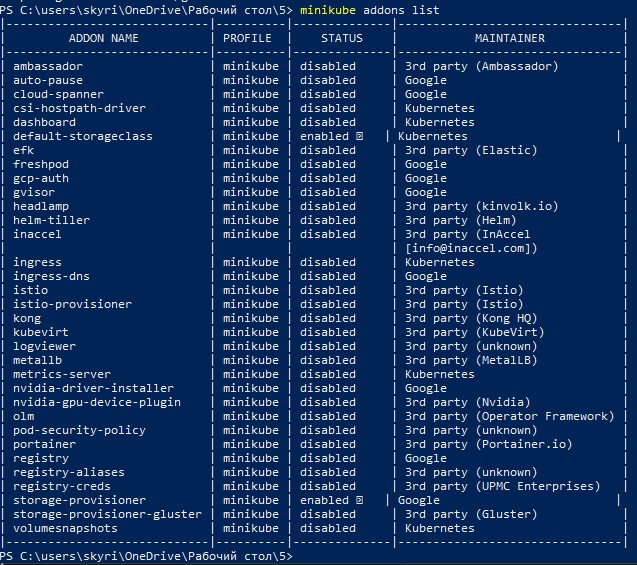


Рисунок 11 - Просмотр доступных дополнений

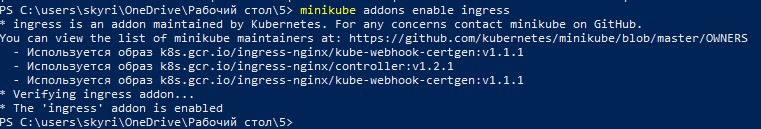


Рисунок 12 - Установка дополнений

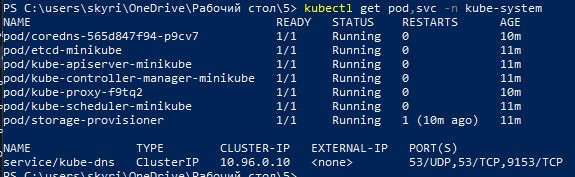


Рисунок 13 - Список подов и сервисов



Рисунок 14 - Отключение дополнения

Завершим работу и освободим ресурсы машины.

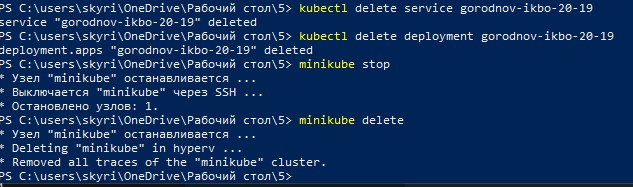


Рисунок 15 - Завершение работы и освобождение ресурсов

# Вывод

В результате выполнения четвертой практической работы были получены навыки работы с Kubernetes и minikube , были созданы поды и сервисы для их работы.

# Ответы на вопросы к практической работе

1. Назовите виды контроллеров в Kubernetes.

*Deployments* - контроллер, который управляет состоянием развертывания подов, которое описывается в манифесте, следит за удалением и созданием экземпляров подов. Управляет контроллерами ReplicaSet.

*ReplicaSet* - гарантирует, что определенное количество экземпляров подов всегда будет запущено в кластере.

*StatefulSets* - так же как и Deployments, управляет развертыванием и масштабированием набора подов, но сохраняет набор идентификаторов и состояние для каждого пода.

*DaemonSet* - гарантирует, что на каждом узле кластера будет присутствовать экземпляр пода.

*Jobs* - создает определенное количество подов и смотрит, пока они успешно не завершат работу. Если под завершился с ошибкой, повторяет создание, которое мы описали определенное количество раз. Если под успешно отработал, записывает это в свой журнал.

*CronJob* - запускает контроллеры Jobs по определенному расписанию.

2. Как называется командная строка в Kubernetes?

Kubectl — это инструмент командной строки для управления кластерами Kubernetes. kubectl ищет файл config в директории $HOME/.kube. Вы можете указать другие файлы kubeconfig, установив переменную окружения KUBECONFIG или флаг --kubeconfig.

3. Что такое под?

Pods или поды — это абстрактный объект в кластере K8S, который состоит из одного или нескольких контейнеров с общим хранилищем и сетевыми ресурсами, а также спецификации для запуска контейнеров.

Это главный объект в кластере, в нем прописаны, какие контейнеры должны быть запущены, количество экземпляров или реплик, политика перезапуска, лимиты, подключаемые ресурсы, узел кластера для размещения.

kube-scheduler планирует размещение пода на узлах кластера

kubelet на рабочем узле кластера запускает под

4. Назовите 2 типа ресурсов, из которых состоит кластер Kubernetes.

Сам кластер K8S состоит из, барабанная дробь, рабочих узлов. В узлах или нодах (Nodes, Worker nodes), помимо контейнеров компонентов самого кластера, размещаются контейнеры наших проектов и сервисов.

Worker nodes состоит из компонентов:

kubelet - сервис или агент, который контролирует запуск компонентов (контейнеров) кластера

kube-proxy - конфигурирует правила сети на узлах

Плоскость управления (Master nodes) управляет рабочими узлами и подами в кластере. Там располагаются компоненты, которые управляют узлами кластера и предоставляют доступ к API.

Control plane состоит из компонентов:

kube-apiserver - предоставляет API кубера

etcd - распределенное key-value хранилище для всех данных кластера. Необязательно располагается внутри мастера, может стоять как отдельный кластер

kube-scheduler - планирует размещение подов на узлах кластера

kube-controller-manager - запускает контроллер

kubelet - сервис или агент, который контролирует запуск основных компонентов (контейнеров) кластер

5. Чем Kubernetes отличается от Docker Swarm?

Kubernetes уже включает множество инструментов оркестровки, например, автомасштабирование.

# Список использованной литературы

1. Установка Kubernetes с помощью Minikube — Текст: электронный [сайт]. — URL: <https://kubernetes.io/ru/docs/setup/learningenvironment/minikube/>

2. K8S для начинающих. Первая часть — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/589415/>

3. Kubernetes или с чего начать, чтобы понять что это и зачем он нужен — Текст: электронный [сайт]. — URL: <https://habr.com/ru/company/otus/blog/537162/>

4. Основы Kubernetes — Текст: электронный [сайт]. — URL: <https://habr.com/ru/post/258443/>

5. Kubernetes — Текст: электронный [сайт]. — URL: <https://kubernetes.io/ru/>