Государственный Университет Молдовы

Факультет Математики и Информатики

Департамент Информатики

“Inițiere în Cloud Computing și Sisteme de Operare Distribuite”

Лабораторная работа 1   
“Введение в Kubernetes”

Проверил: Препелица Аурелия

Выполнил: Чобану Артём

Кишинев 2021

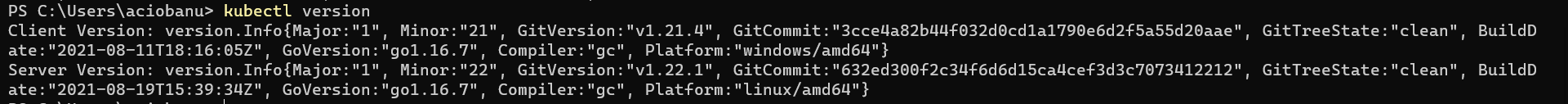
# Требования к лабораторной работе:

* Установка Minikube
* Установка kubectl
* Интерактивное руководство, 6 модулей

# Работа с kubectl

Minikube и kubectl были установлены заранее. Установщики были получены с официального сайта. (Модуль 1)

1. Проверим версию kubectl (возвращает json):



1. Проверим версию minukube:

Text

Description automatically generated

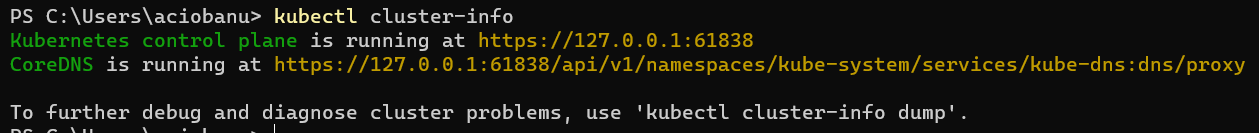
1. Запускаем minikube:

(из-за ошибки с сетью пришлось сначала удалить)

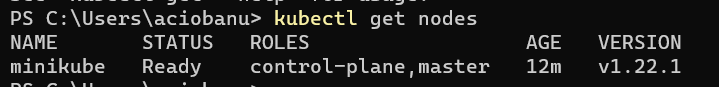
Text

Description automatically generated

1. Узнаем версию кластера kubectl:



1. Получим список доступных узлов узлы(nodes) и данные о них.



Развёртывание приложения (2 модуль):

1. Создание приложения:

kubectl deployment {название} –image={путь до образа приложения}

Text

Description automatically generated

1. Получим информацию о развёртываниях

A picture containing text

Description automatically generated

1. Запустим прокси для частной сети созданного кластера

Text

Description automatically generated

1. Получим имя пода, будем его хранить в переменной POD\_Name

Text

Description automatically generated

1. Получим информацию о поде, осуществляя GET запрос через команду curl:

Text

Description automatically generated

Обзор приложения (Модуль 3)

1. Получим описание подов:

Text

Description automatically generated

1. Получим ответ нашего приложения, используя команду curl, отсылая запрос по адресу прокси. Будем использовать инициализированную заранее переменную POD\_NAME:

Text

Description automatically generated

1. Получим логи контейнера:

Text

Description automatically generated

1. Вызов команд напрямую из контейнера:

Text

Description automatically generated

1. Запустим сессию Bash в контейнере, чтобы взаимодействовать с ним напрямую

Text

Description automatically generated

1. Посмотрим исходный код запущенного приложения в контейнере:

Text

Description automatically generated

Проверим работу:

Text

Description automatically generated

Закроем подключение к контейнеру:

Text

Description automatically generated

Expose your app publicly (Модуль 4)

1. Получим список сервисов:

Text

Description automatically generated

1. Создадим новый сервис и предоставим его внешнему трафику. Для этого используем команду expose и параметр NodePort.

Text

Description automatically generated

Новый список сервисов:

Graphical user interface

Description automatically generated

1. Получим информацию о только что созданном сервисе

A screenshot of a computer

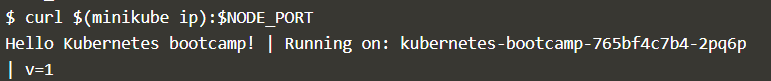
Description automatically generated with medium confidence

1. Создадим переменную окружения NODE\_PORT для хренения значения порта узла:

Text

Description automatically generated

1. Проверим, доступно ли приложение вне кластера:



Использование меток:

1. Получим все запущенные узлы kubernetes-bootcamp

Text

Description automatically generated

1. Установим метку на новую переменную POD\_NAME

Text

Description automatically generated

Отображение самой метки:

Text

Description automatically generated

Получим список подов с меткой:

Text

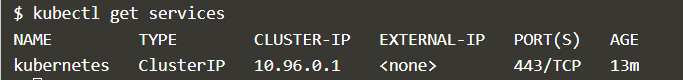
Description automatically generated

1. Удаление сервисов:

Graphical user interface, text

Description automatically generated

Сервис пропал из списка:



В подключении к нему также отказано:



Это показывает, что сервис действительно удалён.

Покажем, что приложение доступно изнутри кластера:



Масштабирование приложения (Модуль 5)

1. Получим развёрнутые приложения

A picture containing diagram

Description automatically generated

NAME – имя приложения в кластере

READY – соотношение реплик CURRENT/DESIRED

UP-TO-DATE – количество обновлённых реплик

AVAILABLE – количество доступных реплик

AGE – время с момента развёртывания приложения

1. Увидим ReplicaSet, созданный развёртыванием

Text

Description automatically generated with low confidence

1. Зададим количество желаемых реплик

A screenshot of a computer

Description automatically generated with medium confidence

Diagram

Description automatically generated

1. Получим изменённые поды:

Text

Description automatically generated

Балансировка нагрузки:

1. Несколько запросов покажут нам распределение между всеми репликами приложения. Как видим, они запущены на разных.

Text

Description automatically generated

1. Уменьшим количество реплик до 2:

Text

Description automatically generated

Text

Description automatically generated

Обновление приложения (Модуль 6)

1. Обновим до второй версии

Text

Description automatically generated

Text

Description automatically generated

1. Проверим обновление:

Text

Description automatically generated

1. Устанавливем другое обновление:

Text

Description automatically generated

Возвращаем на прошлую версию:



# Вывод

В ходе выполнения данной лабораторной работы, я научился работать с Minikube и Kubectl. Я узнал об основных командах для создания, запуска, расширения, обновления и администрирования кластера, как просматривать логи, обновлять версию приложения, откатывать её в случае проблемы.

Kubernetes необходим для упрощения работы с многочисленными веб-сервисами, что безусловно актуально на данный момент. С его помощью упрощается масштабирование, модерация данных приложений, и, конечно, их развёртывание.

# Библиография

<https://minikube.sigs.k8s.io/docs/start/>

<https://kubernetes.io/docs/home/>

<https://kubernetes.io/docs/tutorials/>

<https://kubernetes.io/ru/docs/tasks/tools/install-kubectl/>

<https://docs.docker.com/>

<https://docs.docker.com/get-started/>

<https://aws.amazon.com/ru/docker/>