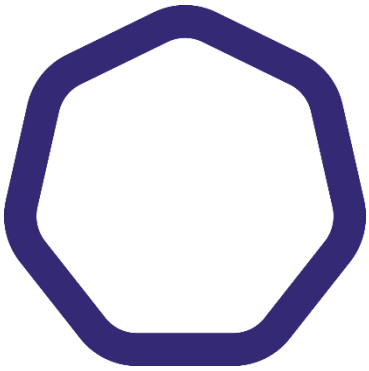
Министерство цифрового развития, связи и массовых коммуникаций Российской Федерации
Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский технический университет связи и информатики»
(МТУСИ)



Кафедра «Сетевые информационные технологии и сервисы»

Лабораторная работа №2

на тему «Создание Kubernetes приложения»

Выполнил:
Студенты 1 курса
Группы М092401(75)
Цыганков Р.О.
Проверил:
к.т.н., Шалагинов А. В.

_____

Москва 2024 г.

Цель работы: изучение создания приложения в Kubernetes, как создавать изображения контейнеров, запуск приложения на Kubernetes, инструменты для улучшения рабочего процесса.

Выполнение работы:

Для выполнения лабораторных работ необходимо:

1. Зайти на сайт: https://learning.kasten.io/profile/
2. Пройти процесс регистрации.
3. Перейти в свой профиль.
4. В разделе «Больше лабораторий для изучения» выбрать лабораторную работу «Создание резервной копии вашего приложения Kubernetes».
5. Нажать на кнопку «Start Lab».

Данный курс состоит из двух частей:

- Теоретический блок, на основании которого созданы вопросы из теста.
- Практический блок.

Теоретический блок:

Команды для создания и масштабирования приложения:

• Обнаружение хранилища K8s

• Kubectl получает класс хранилища

• Изучить вариант использования (Spring PetClinic) со службой передачи данных (MySQL)

• Установка приложения

• Изучение приложения

• Использование его в качестве сервиса для предоставления доступа к приложению

• Запуск нескольких экземпляров приложения

• Масштабирование вашего приложения

• Выполнение текущего обновления

• Добавление данных

• Изучение постоянных требований к объему/объему

Для Поджигателей-

• Внедрить Kubestr, установить другой класс хранилища, использовать Kubestr для сравнения 2 (классы хранения GCP)

Приложения Kubernetes - это готовые для предприятия контейнерные решения со встроенными шаблонами развертывания, отличающиеся переносимостью, упрощенным лицензированием и консолидированным выставлением счетов.

Контейнеры позволяют вам упаковать ваше приложение и его зависимости вместе в один краткий манифест, который может контролироваться версиями, что позволяет легко тиражировать ваше приложение между разработчиками в вашей команде и машинами в вашем кластере.

Контейнеры лучше всего подходят для архитектур, основанных на сервисах. В отличие от монолитных архитектур, где каждая часть приложения взаимосвязана — от ввода—вывода до обработки данных и рендеринга, архитектуры, основанные на сервисах, разделяют их на отдельные компоненты.

С точки зрения приложения создание экземпляра изображения (создание контейнера) аналогично созданию экземпляра процесса, такого как служба или веб-приложение.

Приложение без состояния - это прикладная программа, которая не сохраняет данные клиента, сгенерированные в одном сеансе, для использования в следующем сеансе с этим клиентом, например, службы печати, микросервисы.

Нажмите на стрелку справа, чтобы получить всю необходимую информацию для выполнения задания.

В приложениях с отслеживанием состояния состояние записывается. Под состоянием мы подразумеваем любое изменяемое событие, которое включает внутренние операции, взаимодействие с другими приложениями, переменные среды, пользовательские настройки, содержимое памяти и временное хранилище. Данные, которые хранят такие приложения, зависят от их типов и других факторов, в которых они работают. Обычно приложение с отслеживанием состояния способно записывать ваши предпочтения, отслеживать размер и местоположение вашего окна, а также запоминать файлы, которые вы недавно открыли. Некоторые известные примеры приложений с отслеживанием состояния включают MongoDB, Cassandra и MySQL.

Образ контейнера, в его простейшем определении, представляет собой файл, который извлекается с сервера реестра и используется локально в качестве точки монтирования при запуске контейнеров.

Контейнер - это создание экземпляра изображения во время выполнения.

Механизм контейнера - это часть программного обеспечения, которая принимает пользовательские запросы, включая параметры командной строки, извлекает изображения и с точки зрения конечного пользователя запускает контейнер.

Существует множество контейнерных движков, включая Docker, RKT и CRIO, которые обеспечивают среду выполнения для приложений внутри контейнера.

Вы можете запускать их локально или на удаленном сервере. Контейнеры Docker, работающие на движке Docker, являются легкими, портативными и безопасными. Они работают везде: Linux, Windows, Центры обработки данных, Облачные, Бессерверные и т.д.

Сборка Docker лежит в основе того, что делает Docker таким популярным. Это позволяет вам легко создавать и обмениваться образами переносимых контейнеров Docker с использованием открытых стандартов, а

также создавать образы для нескольких архитектур процессоров и ОС и делиться ими в вашем личном реестре или в Docker Hub.

Docker Desktop - это приложение для компьютеров macOS и Windows для создания и совместного использования контейнерных приложений и микросервисов.

Docker Compose - это инструмент для определения и запуска многоконтейнерных приложений Docker. С помощью Compose вы используете файл YAML для настройки служб вашего приложения. Затем с помощью одной команды вы создаете и запускаете все службы из своей конфигурации.

Helm - это менеджер пакетов для Kubernetes, который позволяет вам упаковывать, совместно использовать и управлять жизненным циклом ваших контейнерных приложений Kubernetes. По сути, вы создаете структурированные пакеты приложений, которые содержат все необходимое для запуска в кластере Kubernetes, включая зависимости, необходимые приложению.

Helm использует диаграммы для упаковки всех необходимых компонентов K8S для развертывания, запуска и масштабирования приложения. Диаграмма - это набор файлов, описывающих связанный набор ресурсов Kubernetes. Одна диаграмма может использоваться для развертывания чего-то простого, например, модуля memcached, или чего-то сложного, например, полного стека веб-приложений с HTTP-серверами, базами данных, кэшами и так далее.

Шаблоны Helm - это подкаталог на диаграмме, который объединяет компоненты K8S, например, Сервис, Набор реплик, Развертывание, Вход и т.д.

Значения руля описаны в значениях.файл yaml, который позволяет пользователям динамически развертывать свои контейнерные приложения. Его структура Yaml содержит значения, соответствующие шаблонам, определенным в манифесте приложения.

Практический блок

Создание приложения осуществляется в несколько этапов

Рисунок 1

Рисунок 2

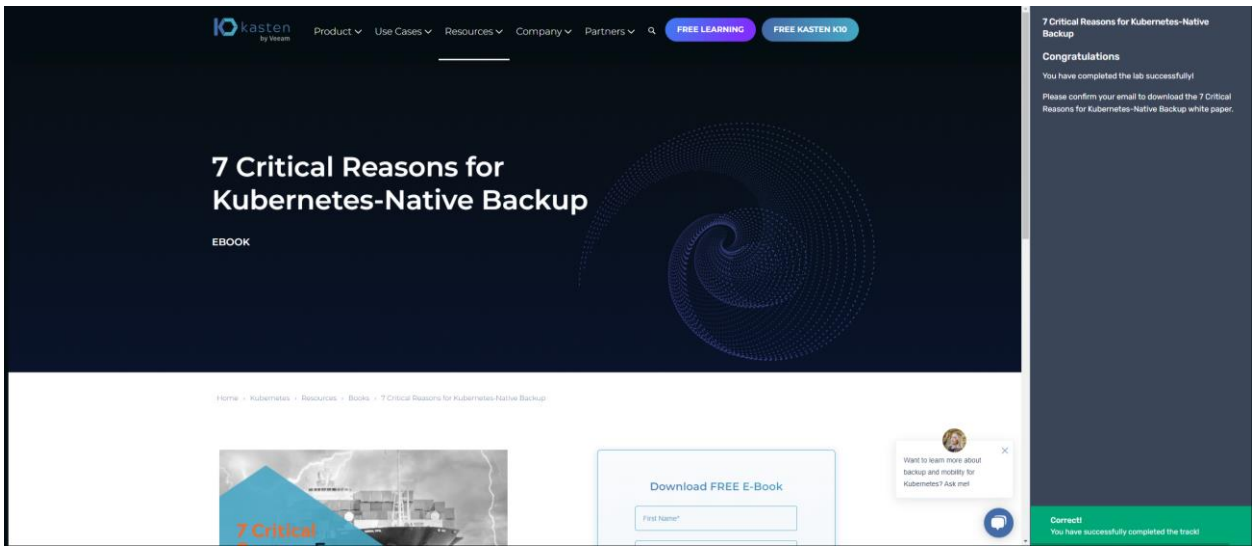


Рисунок 1

Заключение

В процессе лабораторной работы были изучены основы для создания приложения в Kubernetes, а также особенности создания изображения контейнеров, запуск приложения на Kubernetes и инструменты для улучшения рабочего процесса.