Отчет 10. Деплой сервисов в k8s (minikube)

Задача: «Куберизировать» приложение Flask+Redis из предыдущей практики и обновить

1. Разворачиваем на vm мини-кластер k8s- minikube

2. (опционально) Подключаемся к кластеру с хостовой машины через приложение Lens

3. Собираем образы будущих контейнеров и прокидываем их внутрь minikube

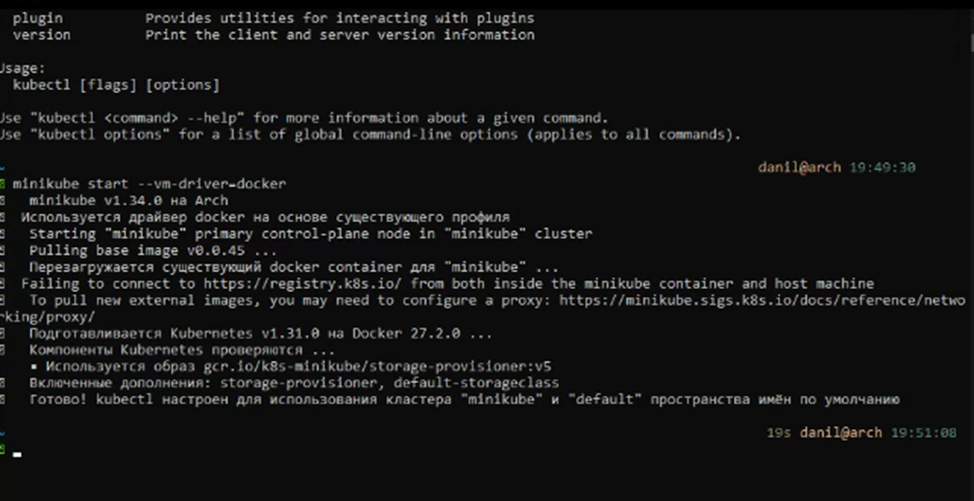
4. Создаем манифесты– описания приложений и сервисов

5. Разворачиваем deployment/ReplicaSet с application-серверами и БД и сервисы

6. Активируем балансировщик входящих запросов (служба LoadBalancer)

7. Проверяем работу сервисов 8. Обновляем веб сервисы на новую версию (RollingUpdate)

Запускаем Minicube

Проверяем, какие у нас есть ноды и сервисов в кластере.



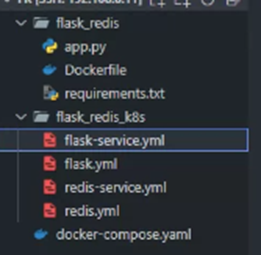
Производим сборку образа из исходников прямо внутри minikube с помощью команды "minikube image build -t flask:v1 flask\_redis/"

Загружаем готовый образ redis внутрь кластера с помощью команды "minikube image load redis:alpine"

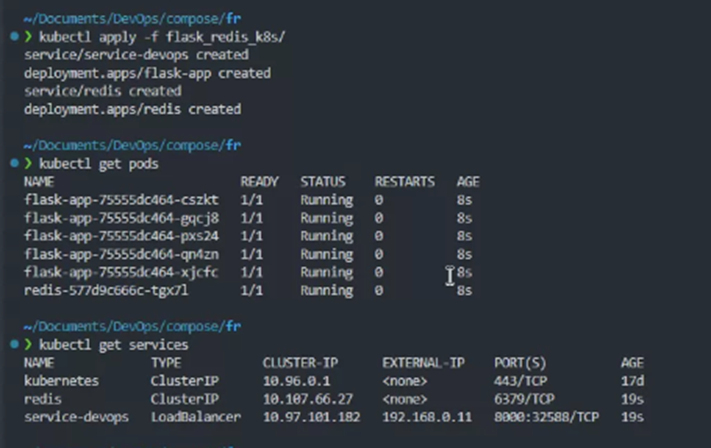
Проверяем, что образы стали доступны внутри кластера



Создаём манифесты для каждого сервиса.

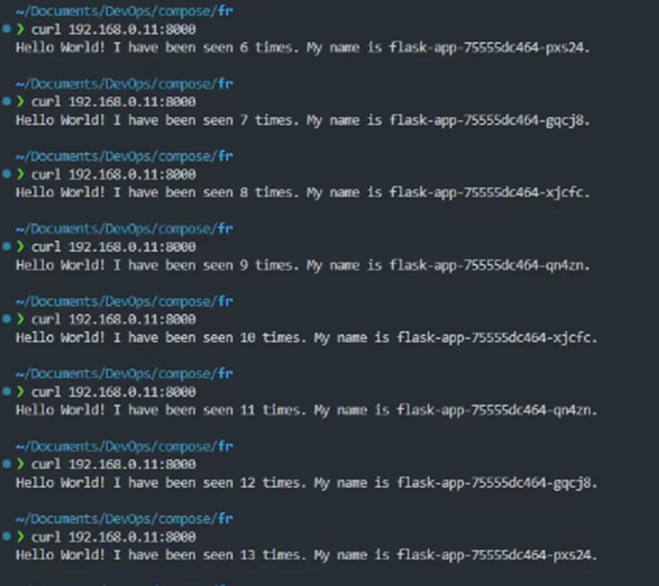


Применяем каталог манифестов целиком.



Открываем порт 8000 и пробрасываем трафик внутрь машины с помощью команды "minikube tunnel --bind-address 10.0.2.15".

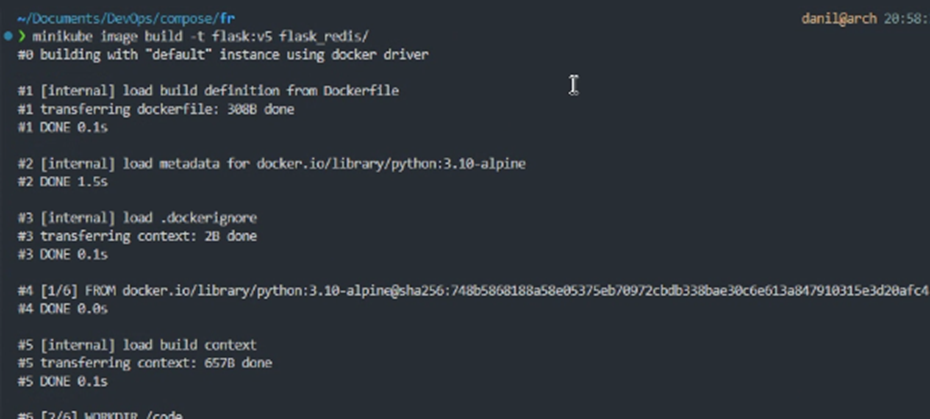
9. Проверяем http://192.168.0.11:8000. Как видно, мы подключились и попадаем на разные поды при подключениях.



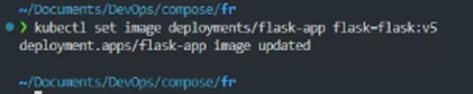
Обновим наше приложение. (добавим version 5 в текст).



Пересобираем образ с новым тегом :v5 и делаем новый образ доступным в кластере minikube.



Обновляем поды на новую версию – патчим деплоймент.



Поды обновились, задача выполнена.

