Практическое задание: Поднятие веб-приложения на K8s.

Условия реализации:

- В качестве образов для двух контейнеров можно (но не обязательно) использовать уже созданные нами в модулях по Docker образы. Единственное отличие от приведённого примера — наши образы не представлены в публичном репозитории, поэтому их нужно вручную загрузить внутрь кластера:

minikube cache **add** <название локального образа>

- Кроме того, файл index.php нужно чуть переписать, т. к. у нас адрес до БД и логин/пароль прописаны статически в самом файле. А должны браться из переменных окружения.

- Если используете публичные образы, проследите, чтобы образ веб-приложения брал данные для подключения к БД из переменных окружения.

Использовать уже приведённые образы **запрещено**.

- Помимо этого, необходимо сделать таблицу (например, в формате Excel) с инвентаризацией вашего кластера. В кратком виде указать установленное ПО на нодах, количество нод, ОС на нодах, адреса.

Необходимо в качестве отчета предоставить:

- все конфигурационные файлы,

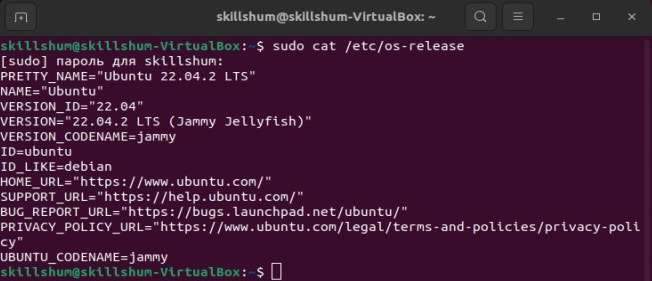
- скриншот из браузера с работающим веб-приложением,

- вывод команды kubectl get all -o wide,

- табличку с инвентаризацией.

Результаты работы загрузите на свой GitHub. В качестве ответа приложите ссылку на репозиторий.

Работа выполнена в виртуальной среде с ОС Ubuntu 22.04



Первоначально подготовим систему к установке Minikube

- Произведем обновление



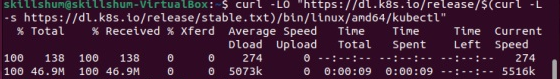
- Устанавливаем необходимые пакеты зависимости

sudo apt install curl

sudo apt install apt-transport-https

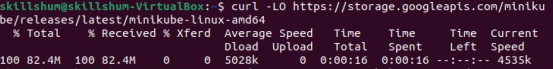
sudo apt install virtualbox

* Устанавливаем kubectl



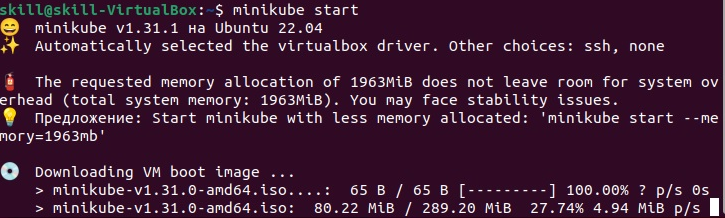


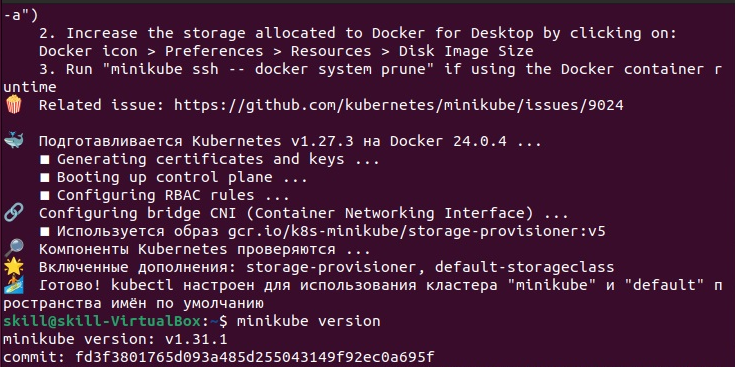
* Устанавливаем minikube





* Проверяем запуск Minikube

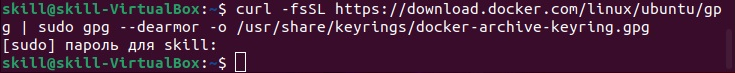




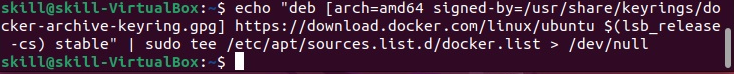
Далее начинаем процесс работы с Docker

* Устанавливаем необходимые зависимости

Добавляем ключ GPG Docker



Добавляем репозиторий Docker



Устанавливаем Docker Engine

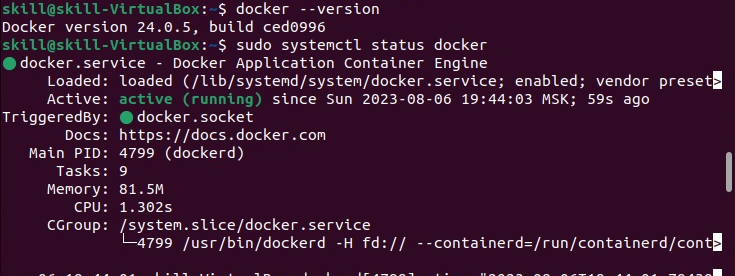
sudo apt update

sudo apt install docker-ce docker-ce-cli containerd.io

Добавляем текущего пользователя в группу Docker

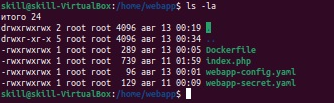
sudo usermod –aG docker $USER

После перезагрузки проверяем состояние Docker

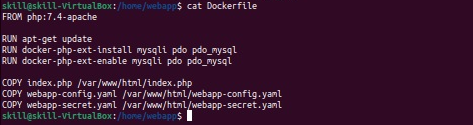


Далее создаем Docker-образ для нашего веб-сервера (файлы в директории /webapp/)

Содержимое директории webapp



Сам Dockerfile

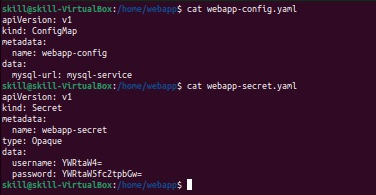


Из файла **webapp-config.yaml** php-скрипт будет извлекать адрес сервера баз данных сервера, а из **webapp-secret.yaml -** логин и пароль.

Файл index.php

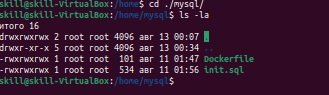


А также вывод содержимого файлов webapp-config.yaml, webapp-secret.yaml

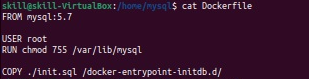


Далее создаем Docker-образ для нашего сервера базы данных (файлы в директории /mysql/)

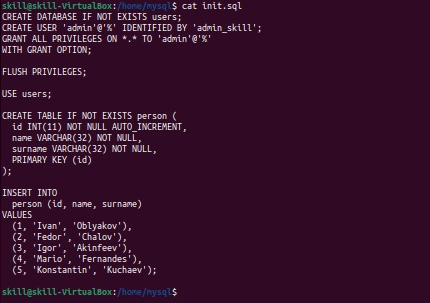
Содержимое директории mysql



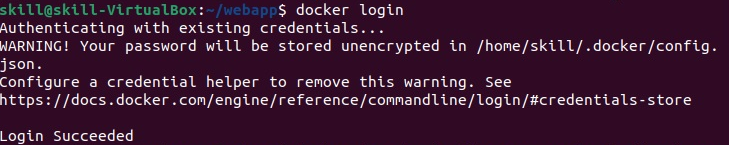
Сам Dockerfile



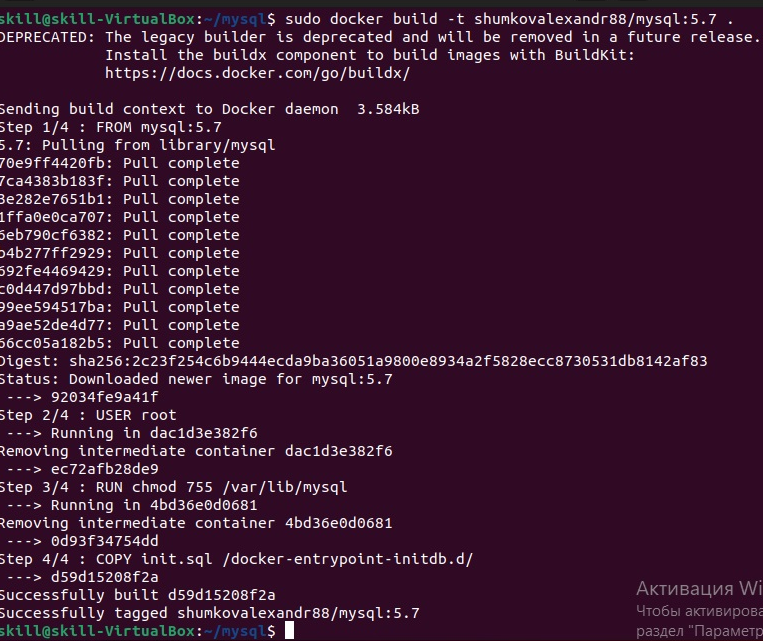
Файл init.sql

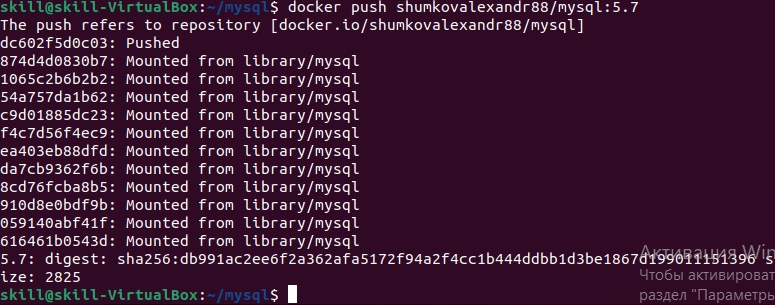


После того, как мы создали поочередно файлы для нашего веб-сервера и сервера базы данных, производим создание образов и выгрузку их на docker-hub (прежде, необходимо зарегистрироваться на сайте, если нет учетной записи)



Образ mysql





Все тоже самое производим и для образа php (webapp)

Применяем манифесты Kubernetes командами:

kubectl apply -f webapp-config.yaml

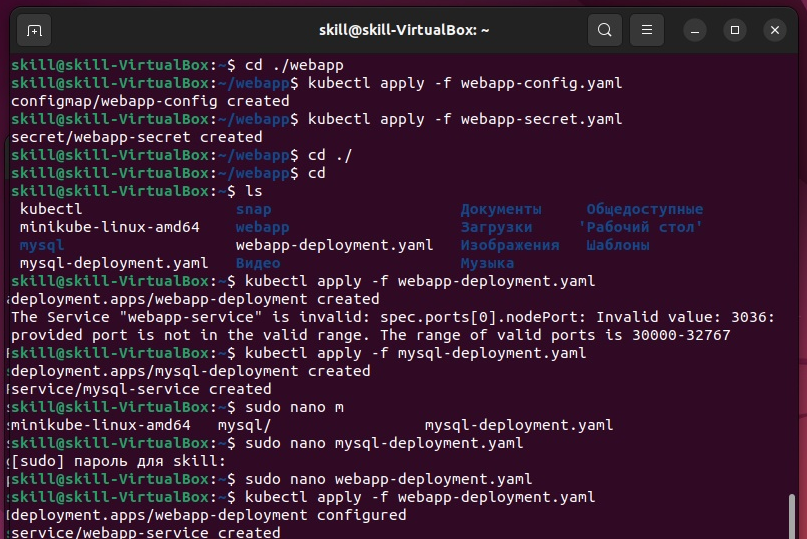
kubectl apply -f webapp-secret.yaml

(находятся оба в директории /webapp)

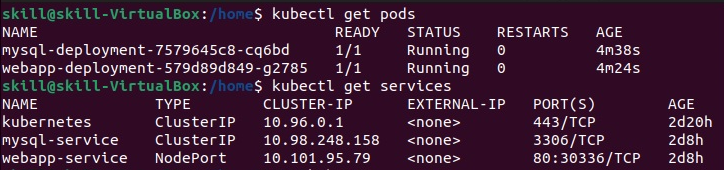
kubectl apply -f mysql-deployment.yaml

kubectl apply -f webapp-deployment.yaml

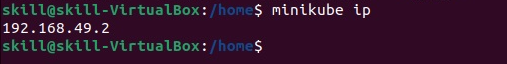
(Файлы представлены в github)



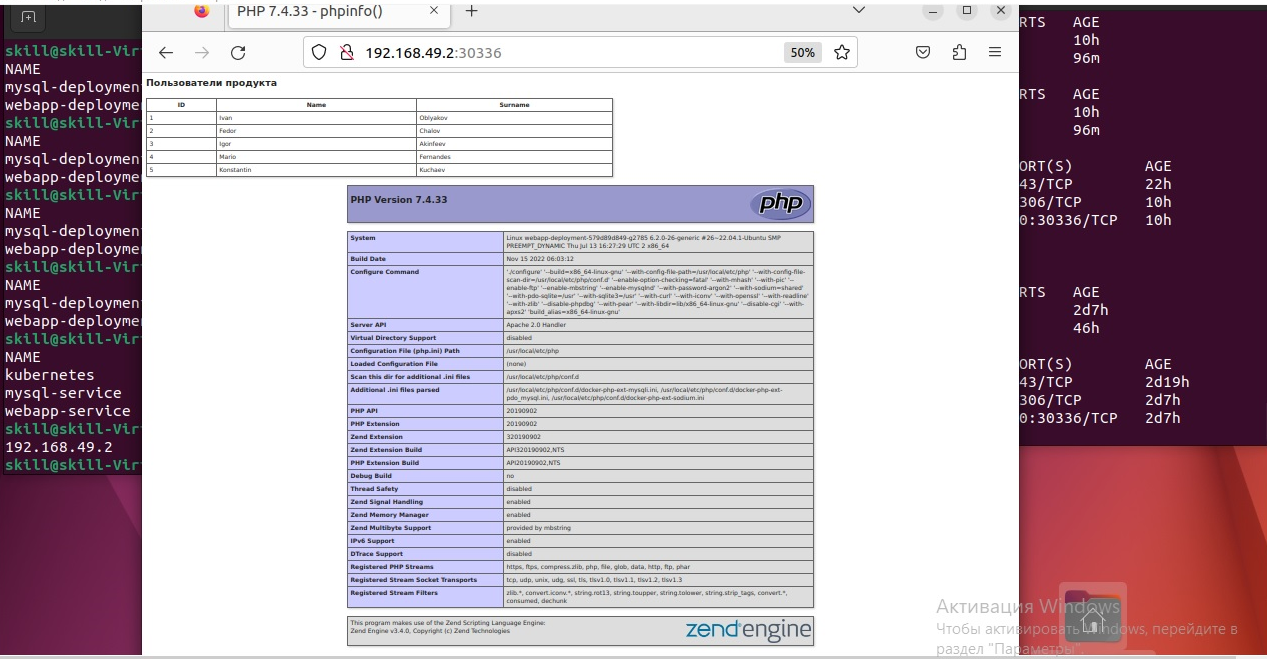
Проверяем статус развертывания:



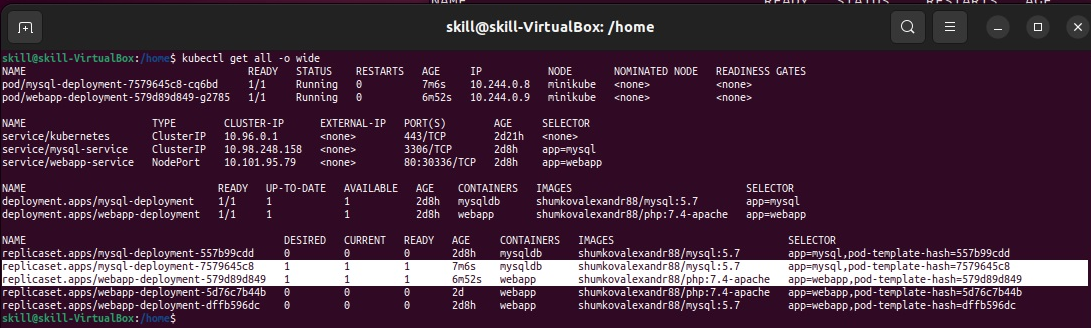
Узнаем ip Minikube



Проверяем браузер, убеждаемся в работоспособности нашего приложения



Также предоставлен вывод и состояние всего кластера



Все конфигурационные файлы загружены, инвентаризация кластера отображена в файле inventarization\_cluster.xlsx