Задание 1:

1) создать сервис, состоящий из 2 различных контейнеров: 1 - веб, 2 - БД

2) далее необходимо создать 3 сервиса в каждом окружении (dev, prod, lab)

3) по итогу на каждой ноде должно быть по 2 работающих контейнера

4) выводы зафиксировать

1. Подгатавливаем файл compose.yml со следующим содержимым:

services:

db:

image: mariadb

networks:

- db-admin

environment:

- MARIADB\_ROOT\_PASSWORD=test123

admin:

image: phpmyadmin

networks:

- db-admin

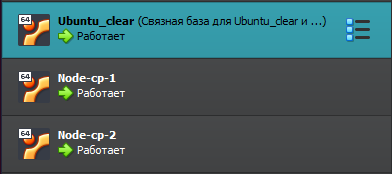
ports:

- "8081:80"

networks:

db-admin:

1. Создаем сервис из 3 нод с уникальными ip



1. На главной ноде инициализируем swarm:

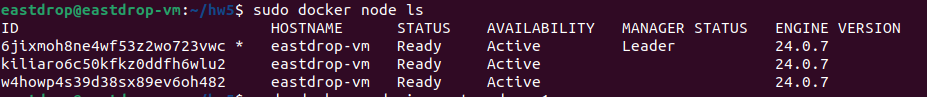
sudo docker swarm init

1. Подключаем к ней оставшиеся с помощью следующей команды, содержащей ключ:

sudo docker swarm join --token SWMTKN-1-0ovhdmofc1ulcxqxdd49hd6y7lr20uowe36m468mnxixj7sjz0-agfprbohqfw1qcmpprj58jl0a 192.168.1.106:2377

1. Проверяем, что все ноды подключены к одному кластеру:

sudo docker node ls

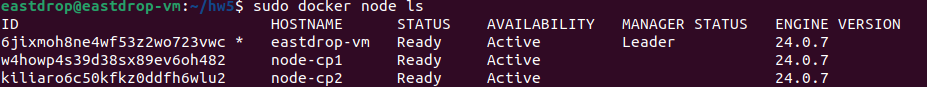


1. Задаем каждой ноде свой хостнейм

sudo hostname node-cp1

sudo hostname node-cp2

И проверяем еще раз, что хостнейм применился



1. Задаем каждой ноде свой label

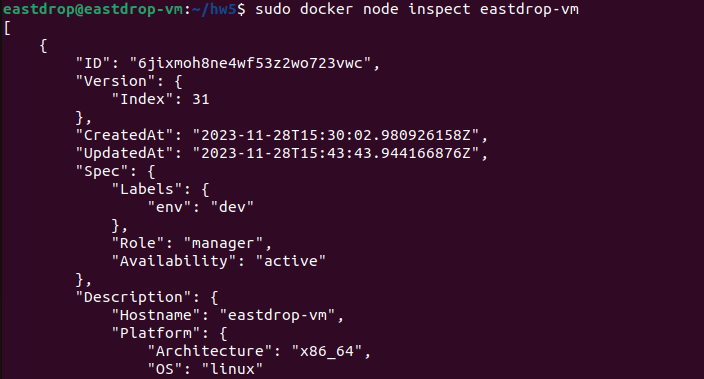
140 sudo docker node update --label-add env=dev eastdrop-vm

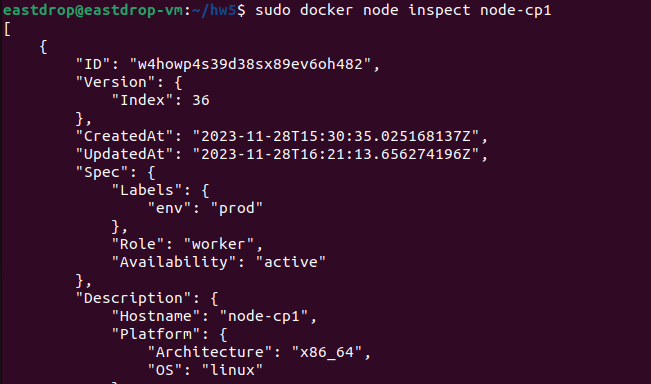
141 sudo docker node update --label-add env=prod node-cp1

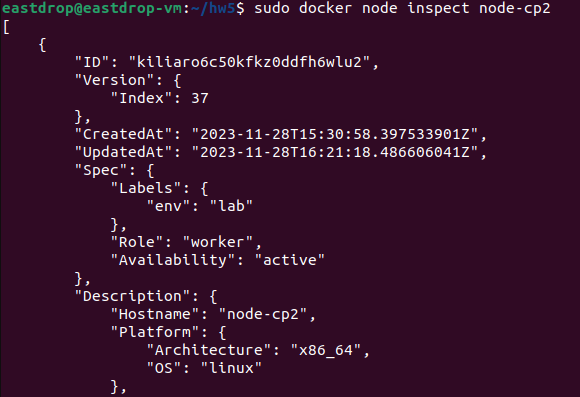
142 sudo docker node update --label-add env=lab node-cp2

1. Проверяем это с помощью команды:

sudo docker node inspect eastdrop-vm







1. Далее мы запускаем на каждой ноде compose.yml из папки с данным файлом

sudo docker-compose up -d

1. В итоге видим, что наша бд с адмайнером корректно запускаются в фоновом режиме:

