cat docker-compose.yml

version: "3.9"

services:

db:

image: mariadb:10.10.2

restart: always

environment:

MYSQL\_ROOT\_PASSWORD: 12345

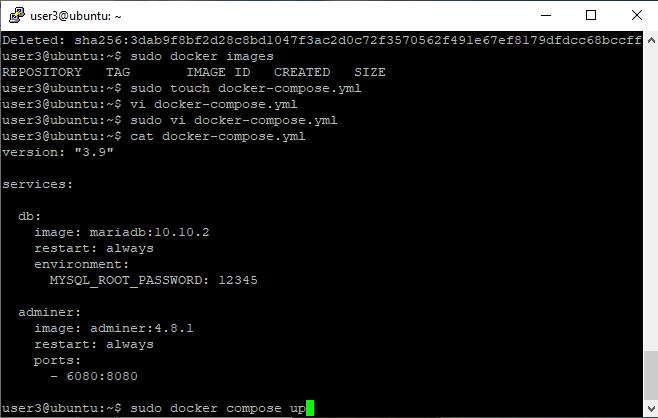
adminer:

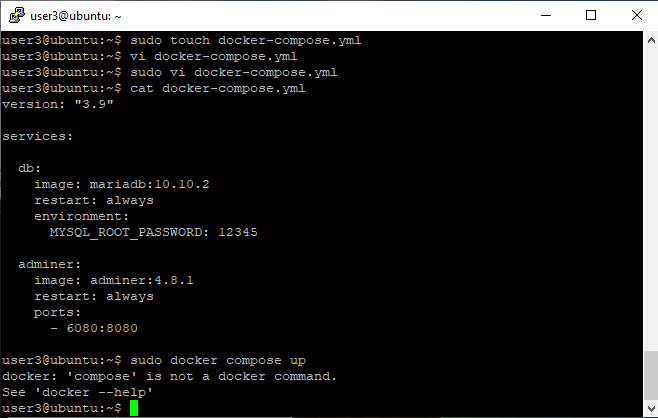
image: adminer:4.8.1

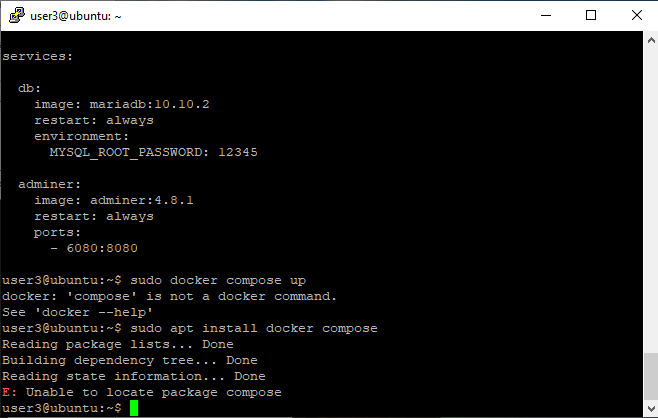
restart: always

ports:

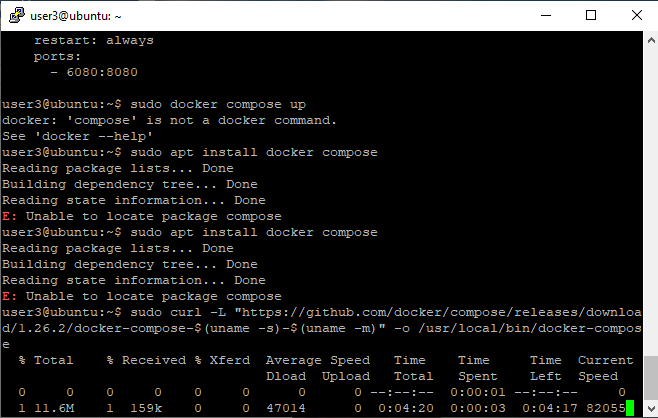
- 6080:8080



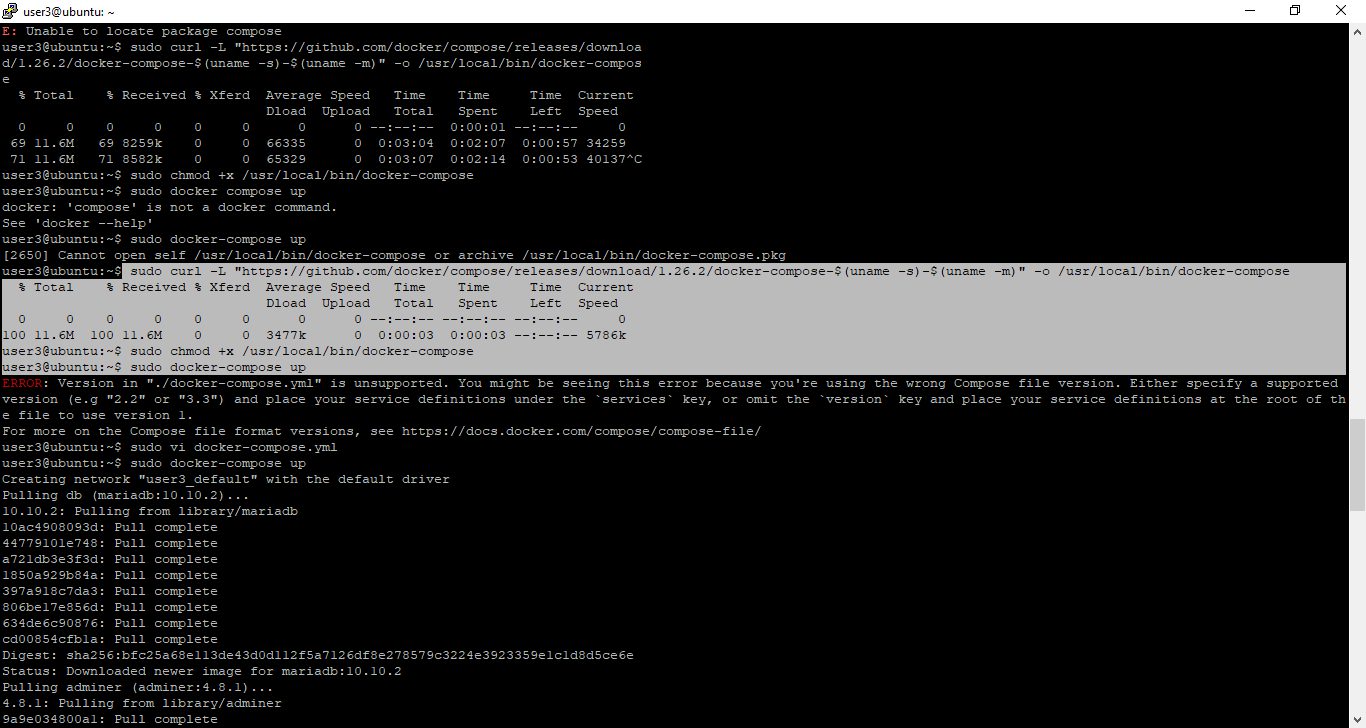




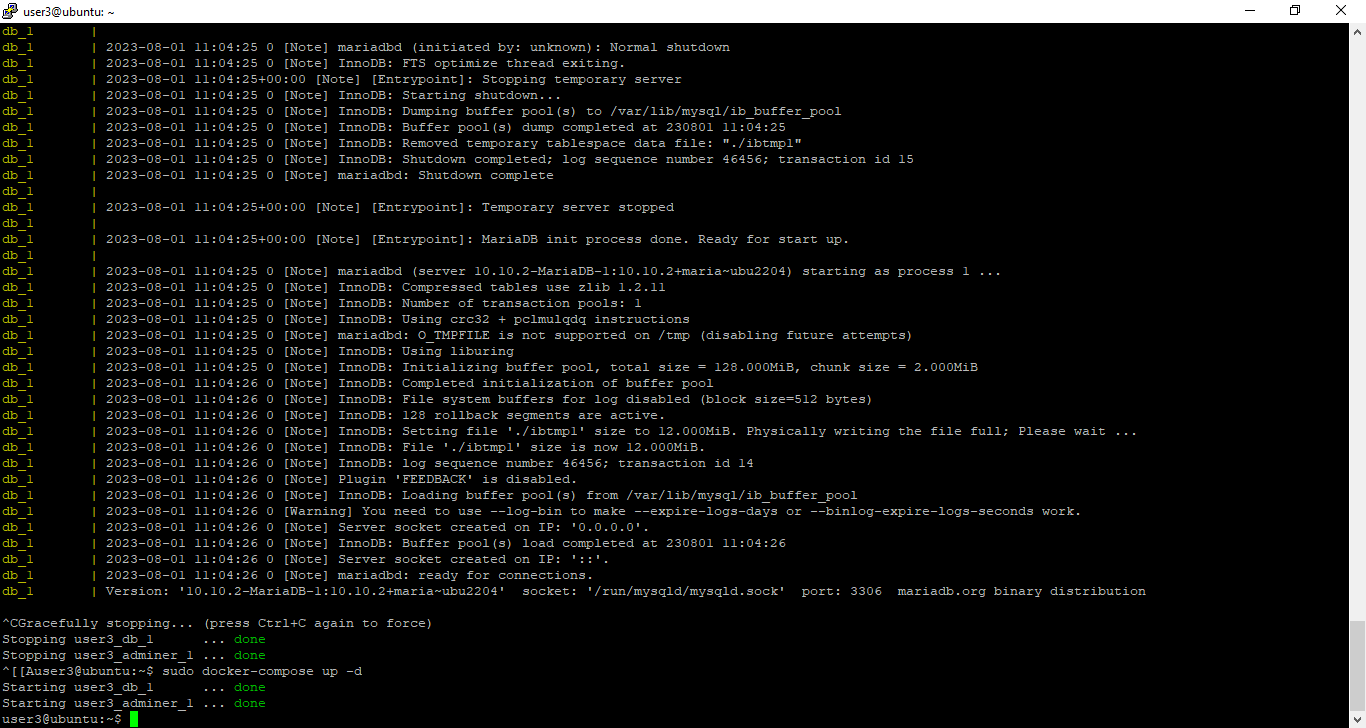
sudo curl -L "https://github.com/docker/compose/releases/download/1.26.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose



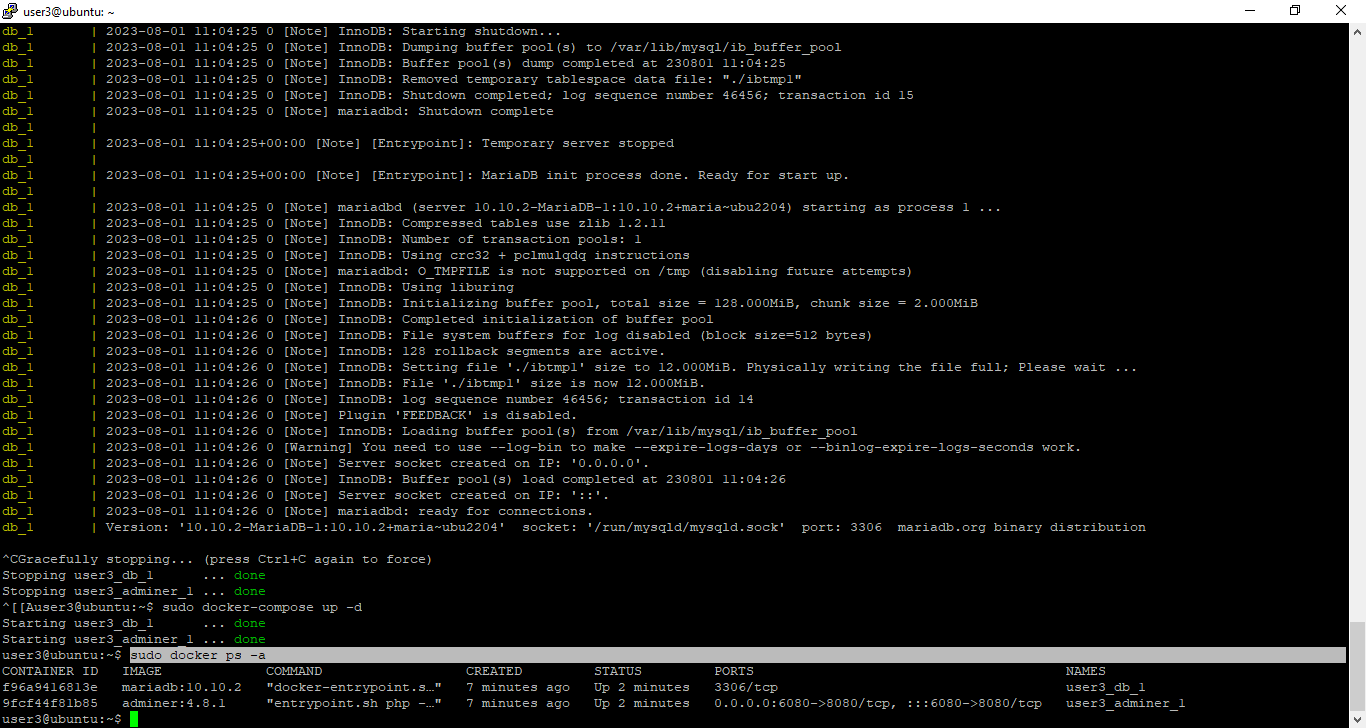
sudo chmod +x /usr/local/bin/docker-compose



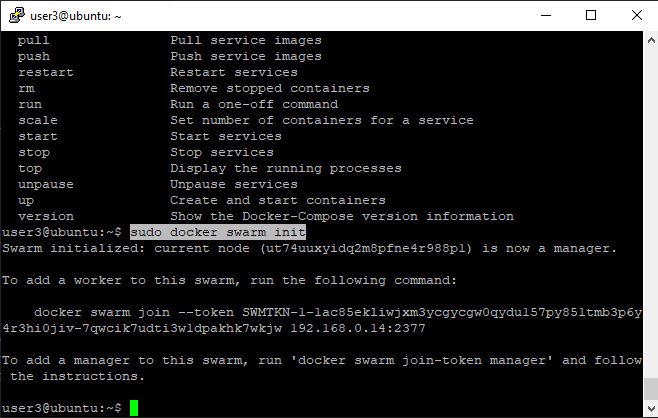
Контейнер запустился в интерактивном режиме, и терминал недоступен. Чтобы запустить в фоновом режиме, добавляем –d:



Статус запущенных контейнеров:



sudo docker swarm init

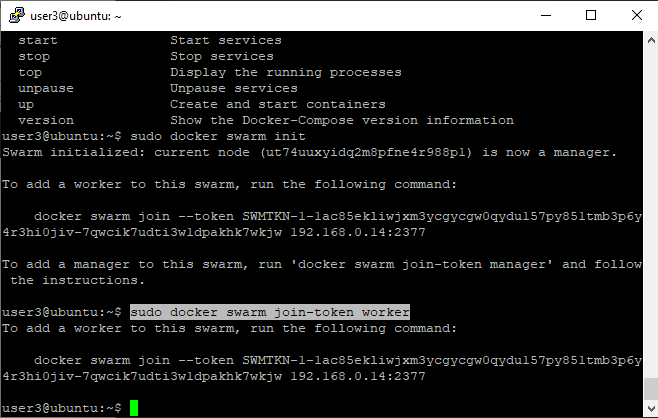


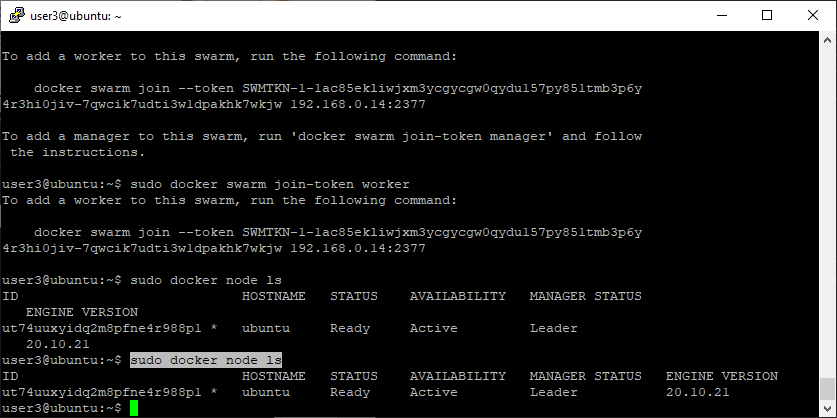
Swarm initialized: current node (ut74uuxyidq2m8pfne4r988p1) is now a manager.

To add a worker to this swarm, run the following command:

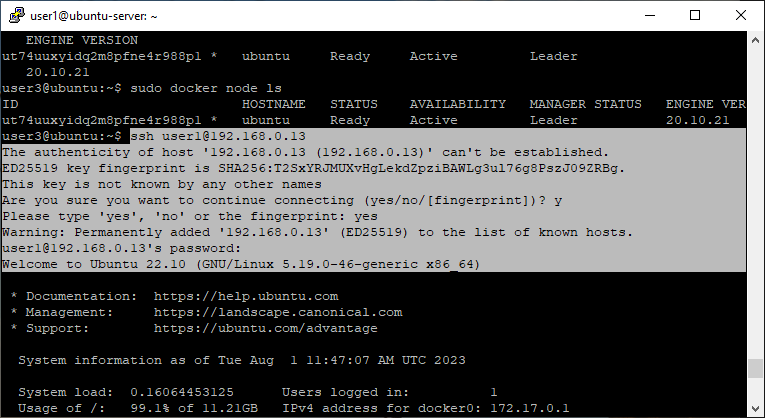
docker swarm join --token SWMTKN-1-1ac85ekliwjxm3ycgycgw0qydu157py851tmb3p6y4r3hi0jiv-7qwcik7udti3w1dpakhk7wkjw 192.168.0.14:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

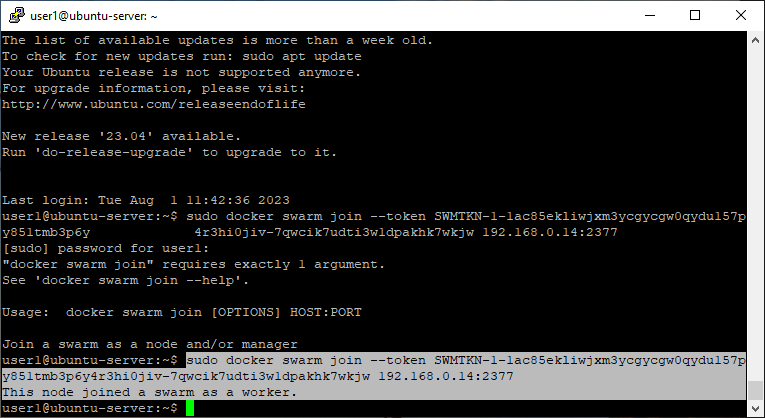


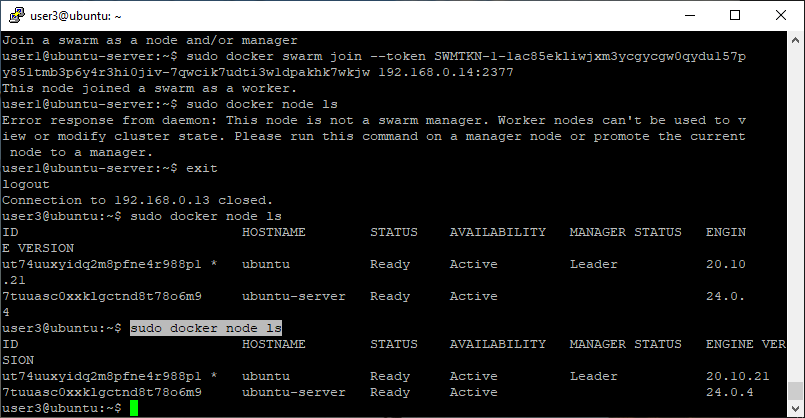


Соединяемся со второй нодой:



Добавляем вторую ноду к swarm в качестве worker:



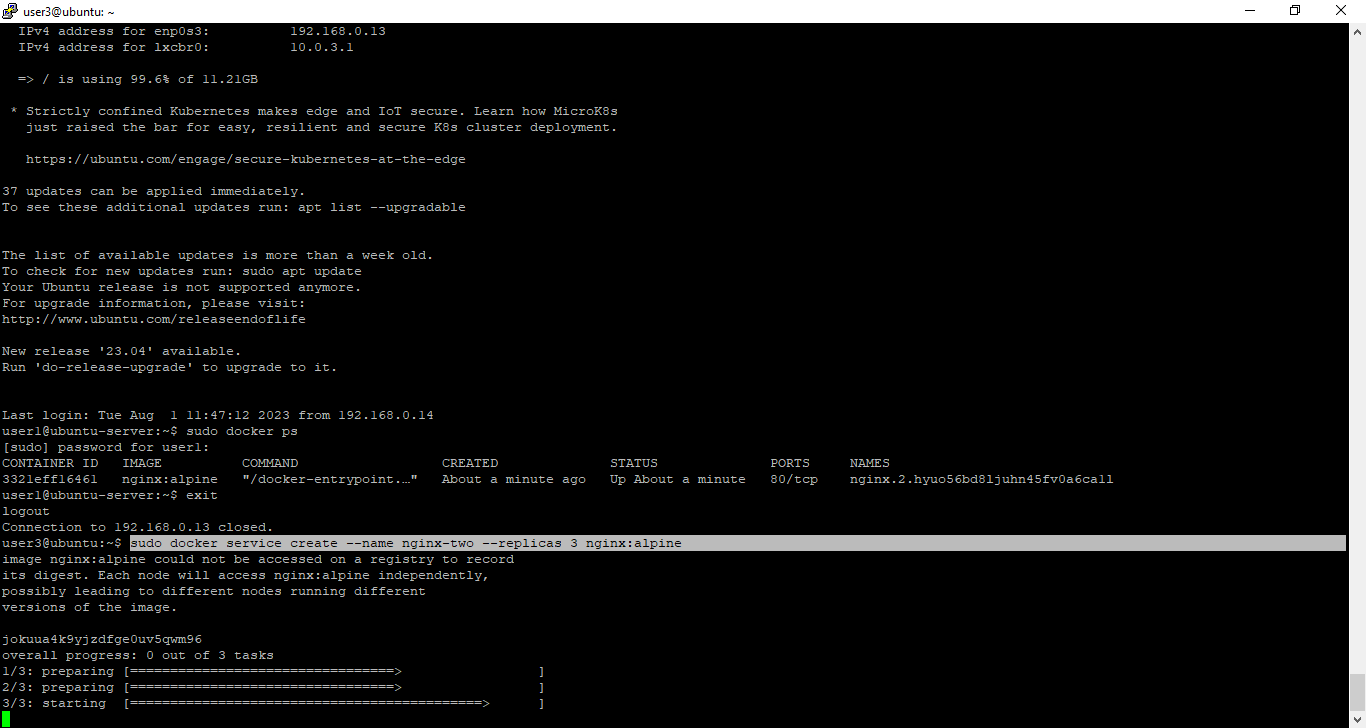


sudo docker node inspect self ut74uuxyidq2m8pfne4r988p1 (id ноды) – информация о ноде!!!

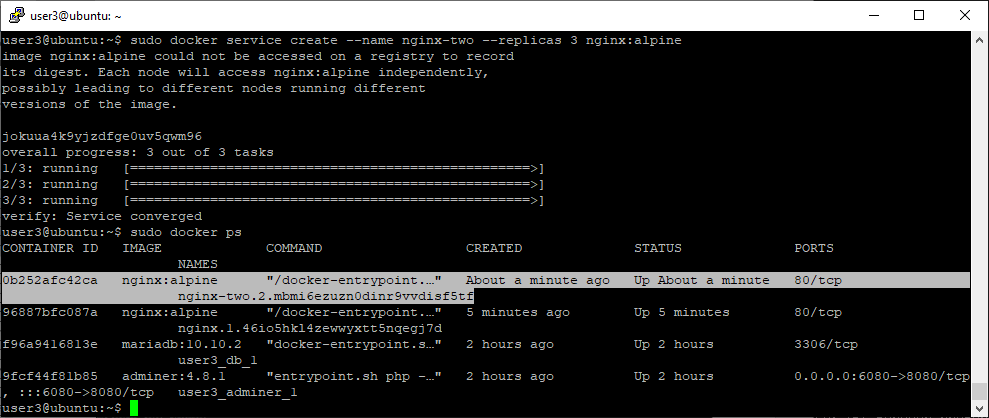
sudo docker node update --role "manager" 7tuuasc0xxklgctnd8t78o6m9 (id ноды) – изменение роли ноды, здесь даем права управления



Создаем 3 реплики одного контейнера в нашем swarm из главной ноды:

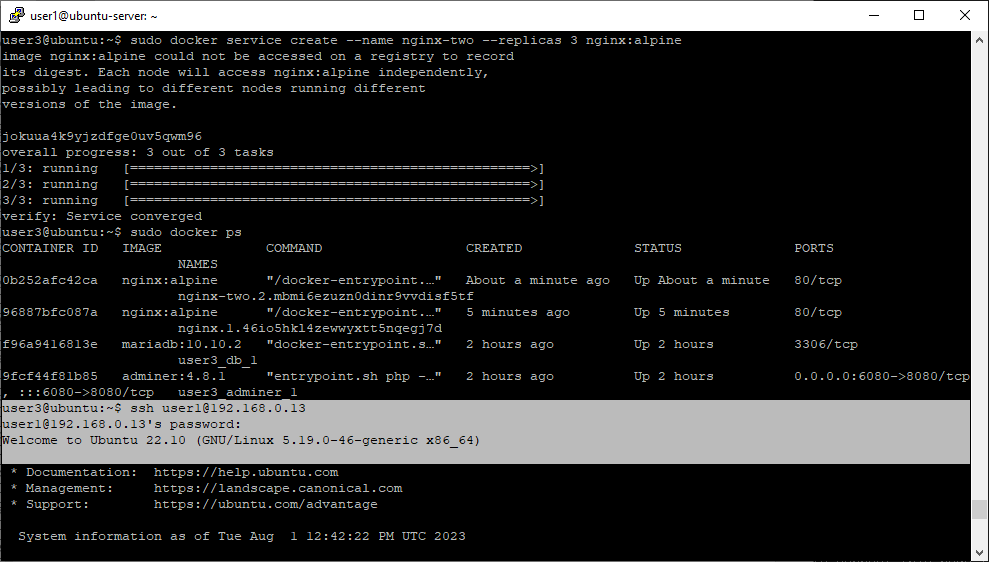


Но только одна из них запустилась в главной ноде(управляющей):

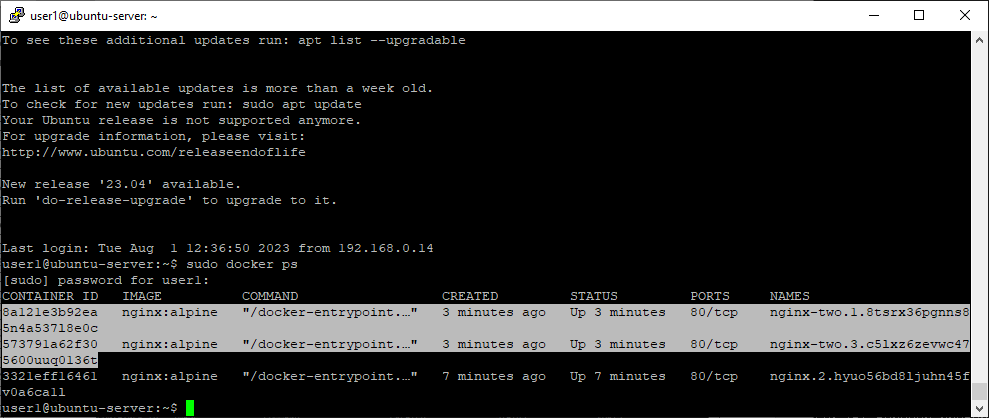


Две другие запустились во второй ноде, т.к. так больше места (перераспределение исходя из ресурсов для поддержания баланса):

1. Заходим во вторую ноду:

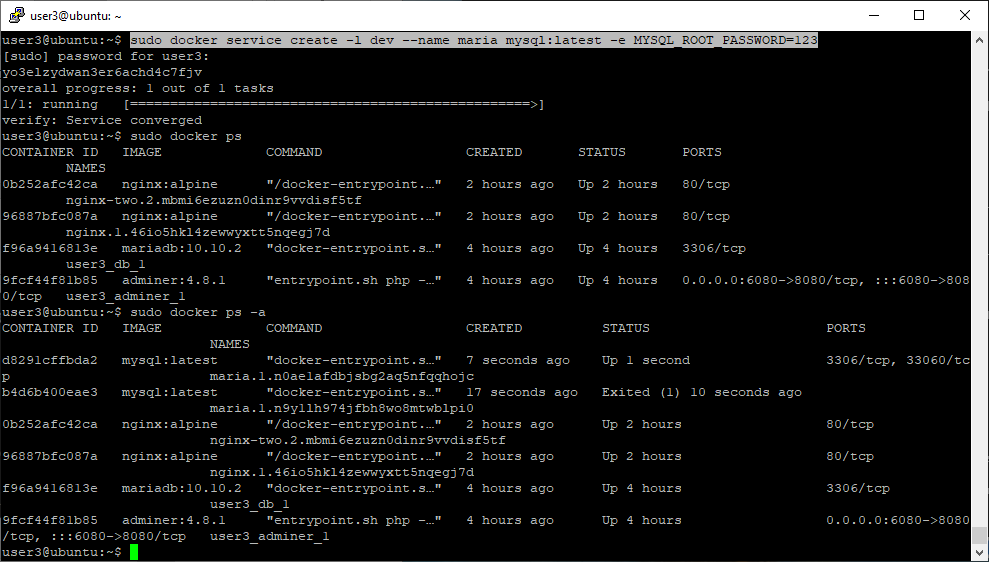


1. И смотрим запущенные контейнеры:

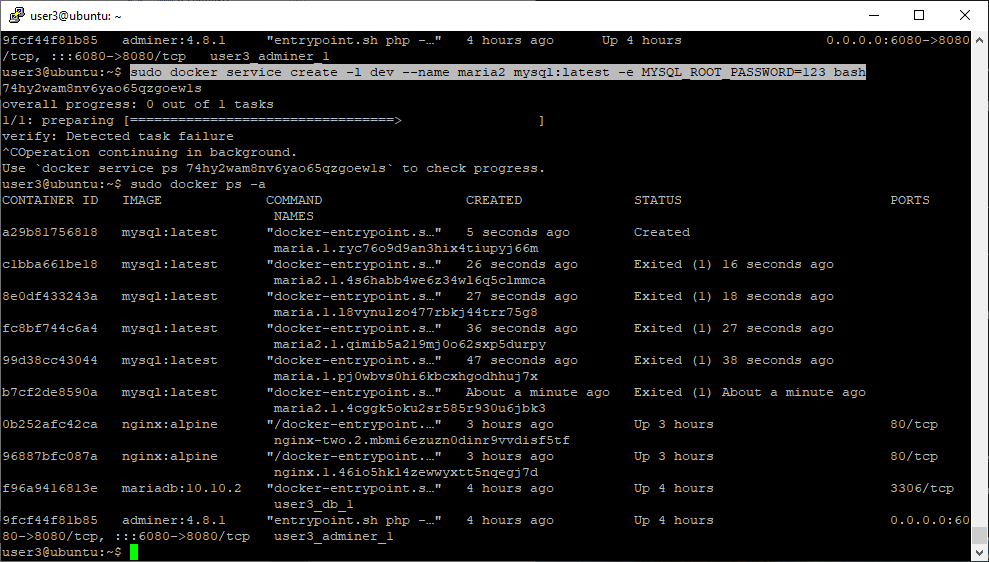


ДОМАШНЕЕ ЗАДАНИЕ

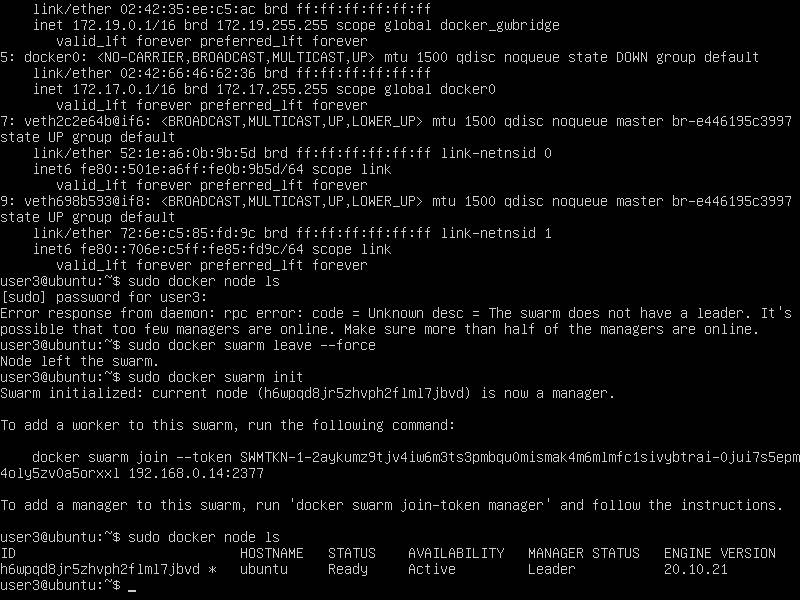
sudo docker service create -l dev --name maria mysql:latest -e MYSQL\_ROOT\_PASSWORD=123



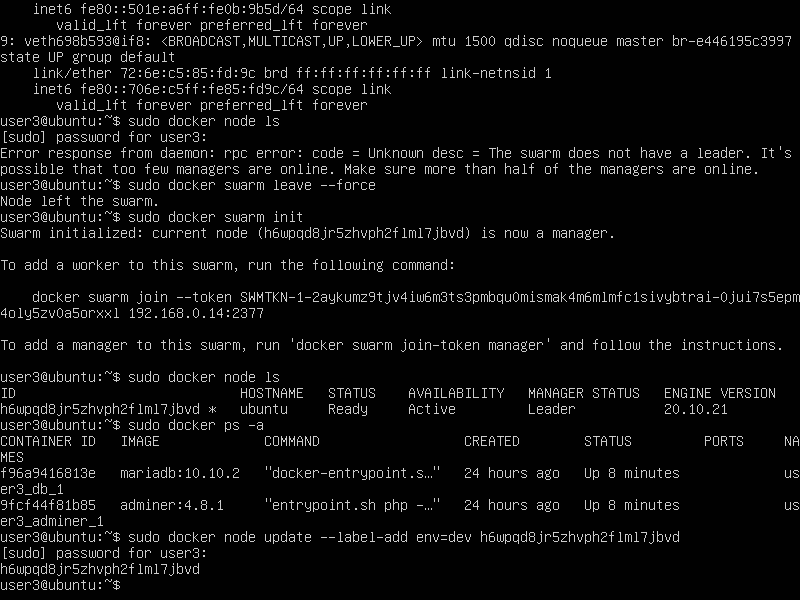
Контейнер все время пытался запуститься, но не смог (preparing->starting->running->preparing)



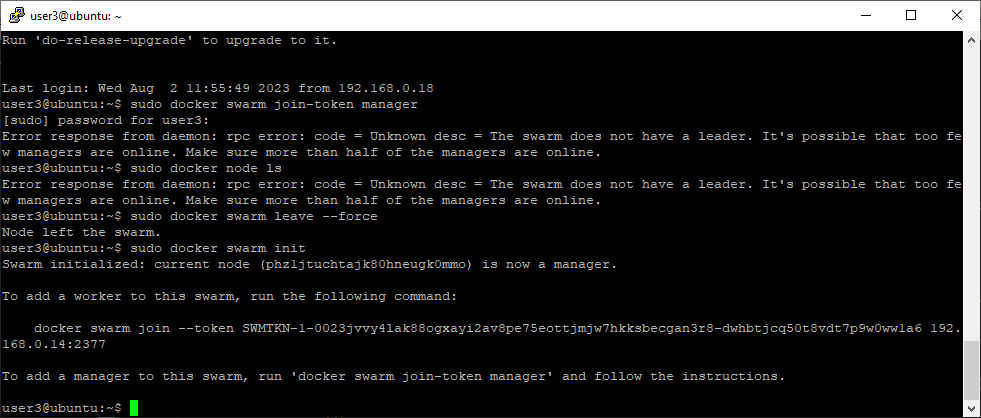
Sudo docker swarm init



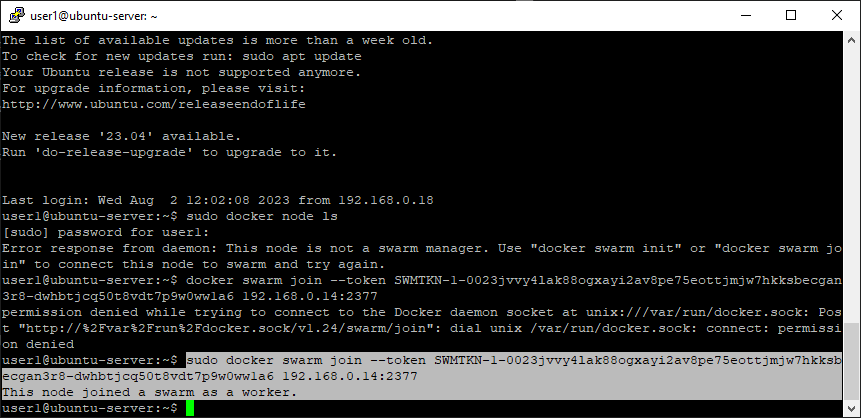
Sudo docker node update –label-add env=dev ID\_NODE:

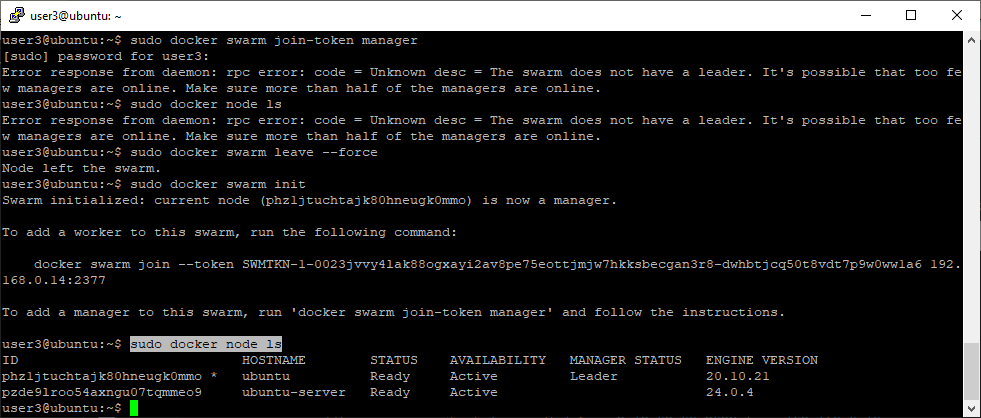


Первая нода:

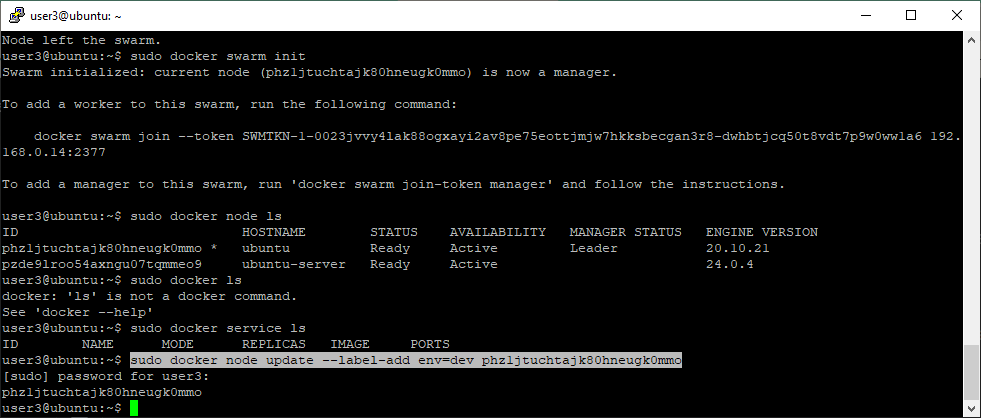


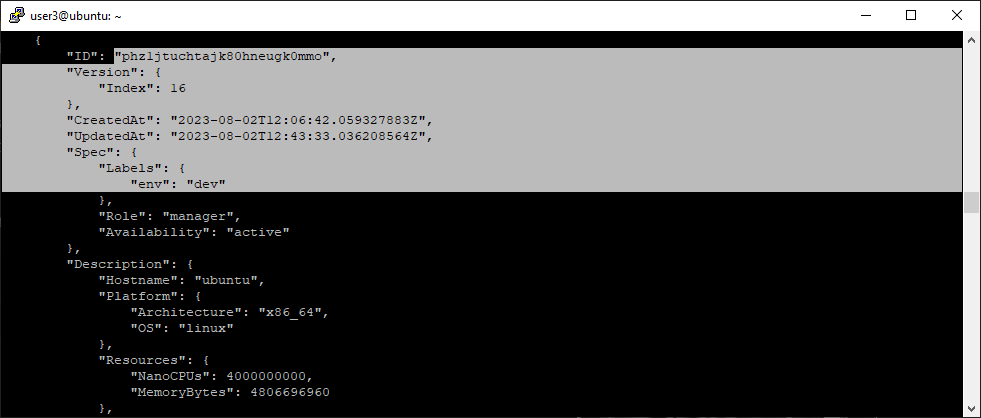
Вторая нода:



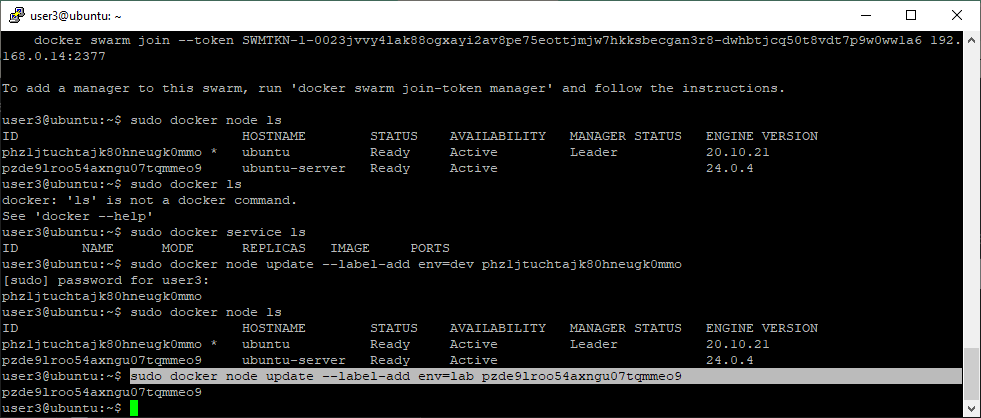


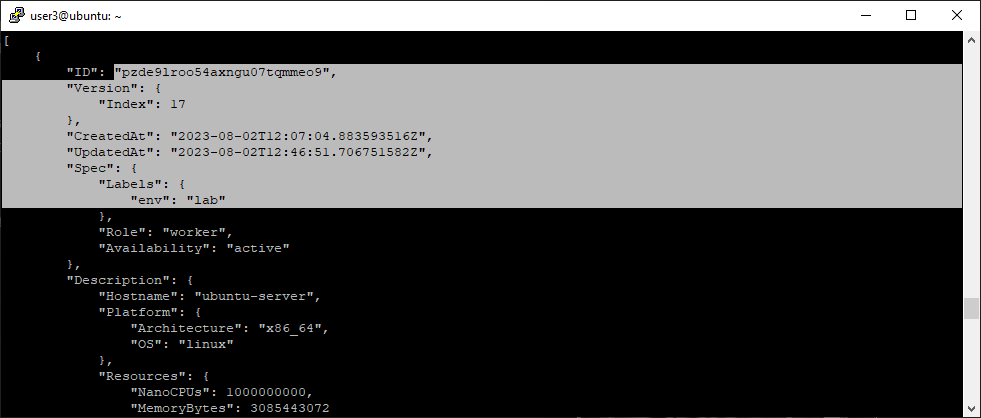
sudo docker node update --label-add env=dev phz1jtuchtajk80hneugk0mmo:



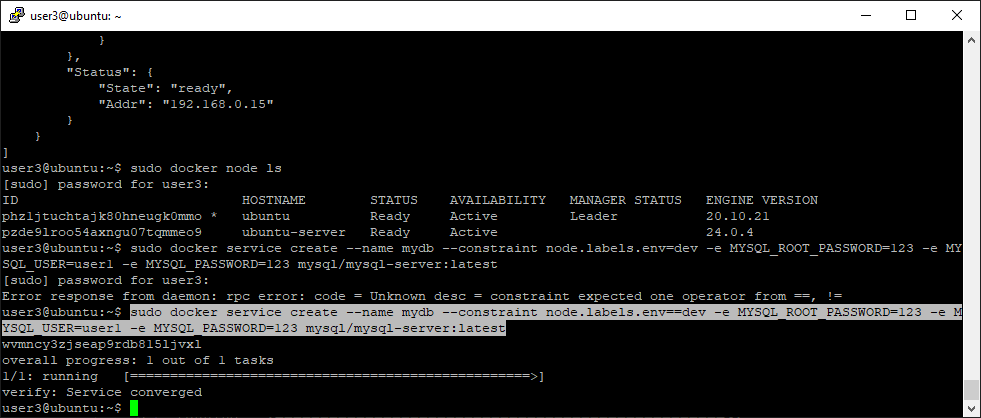


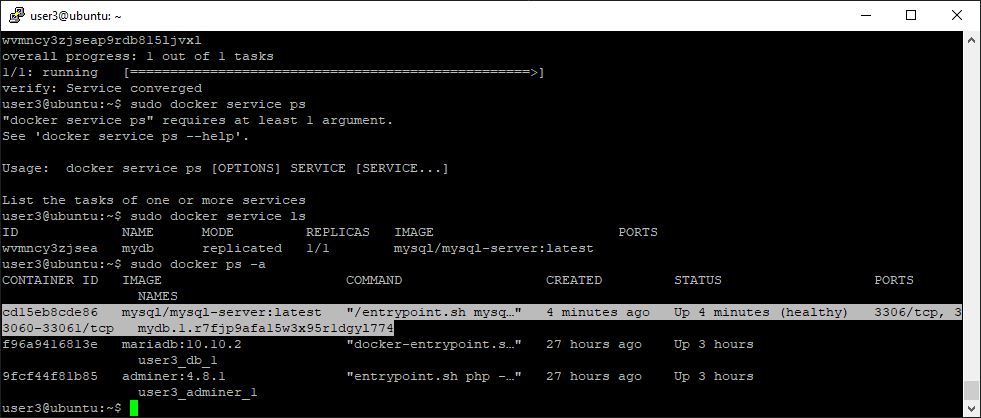
sudo docker node update --label-add env=lab pzde9lroo54axngu07tqmmeo9:



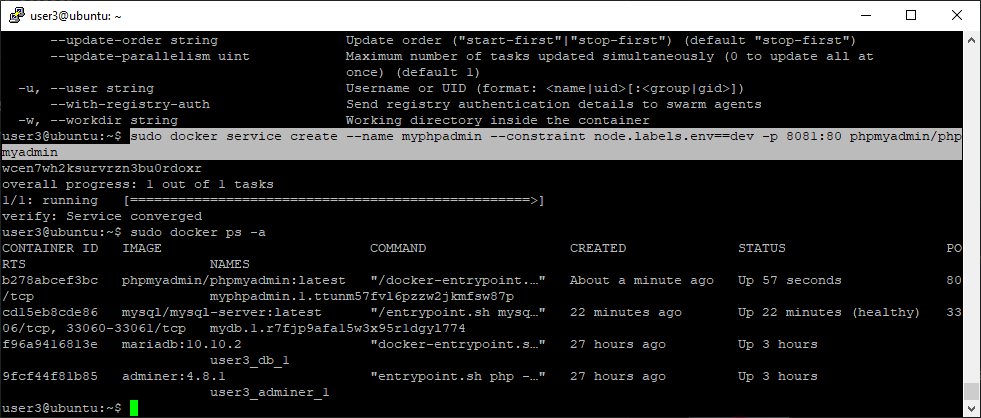


sudo docker service create --name mydb --constraint node.labels.env==dev -e MYSQL\_ROOT\_PASSWORD=123 -e MYSQL\_USER=user1 -e MYSQL\_PASSWORD=123 mysql/mysql-server:latest

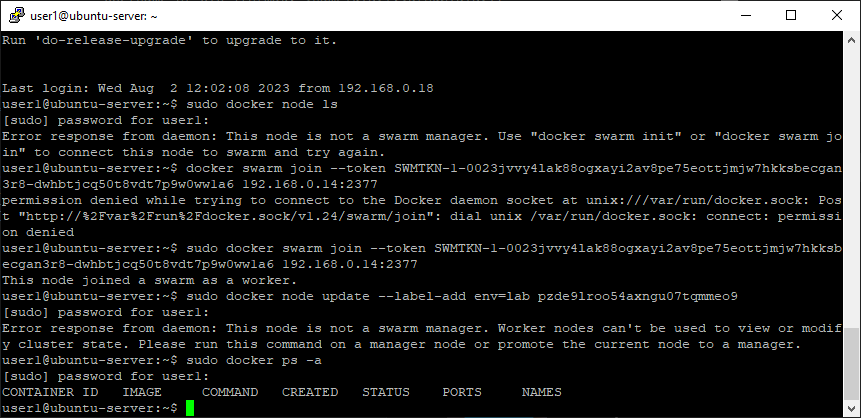




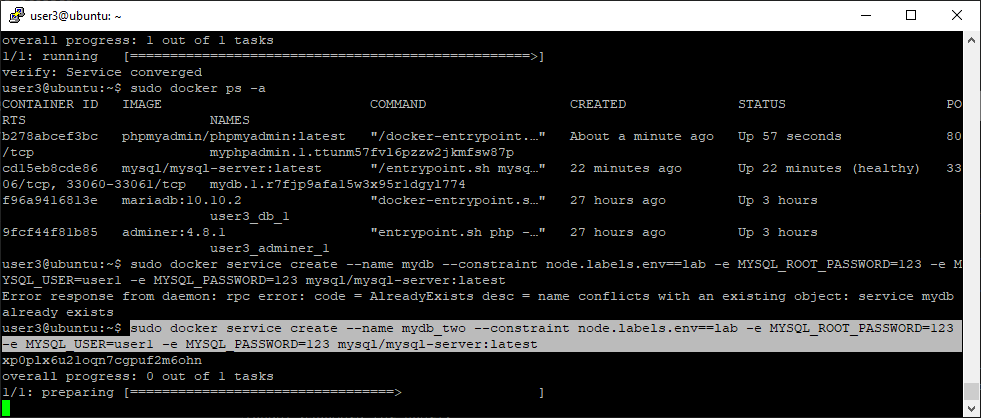
sudo docker service create --name myphpadmin --constraint node.labels.env==dev -p 8081:80 phpmyadmin/phpmyadmin

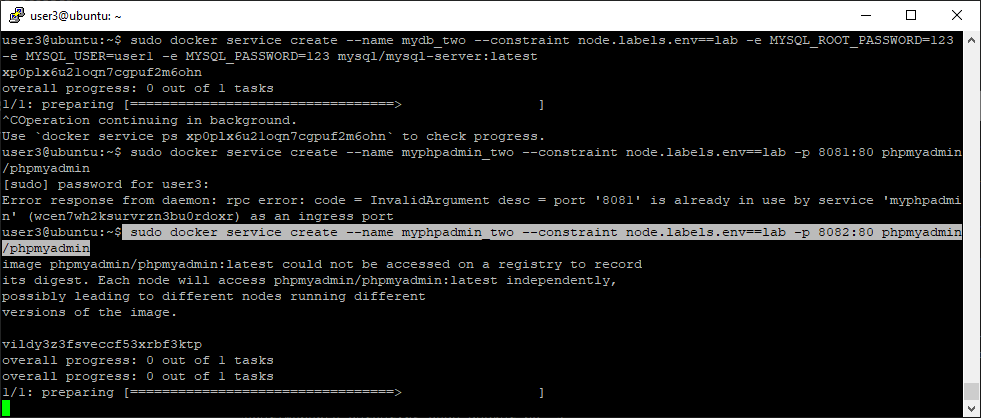


На второй ноде нет запущенных сервисов:

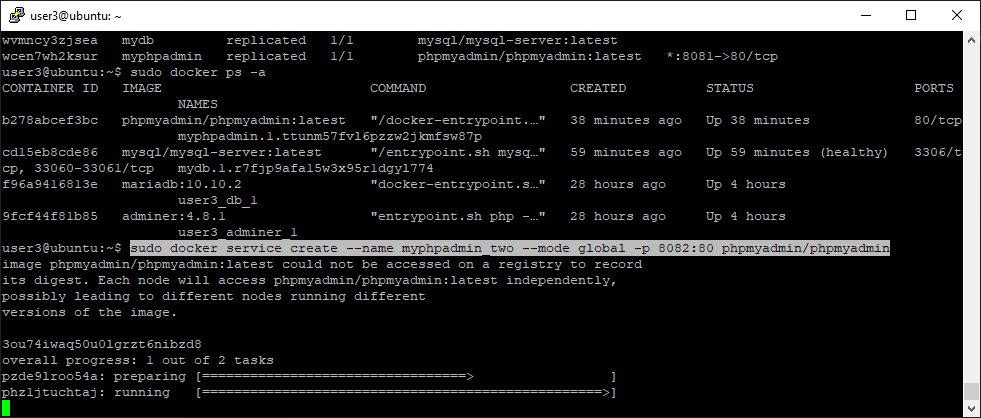


Запустила установку БД на второй ноде lab:

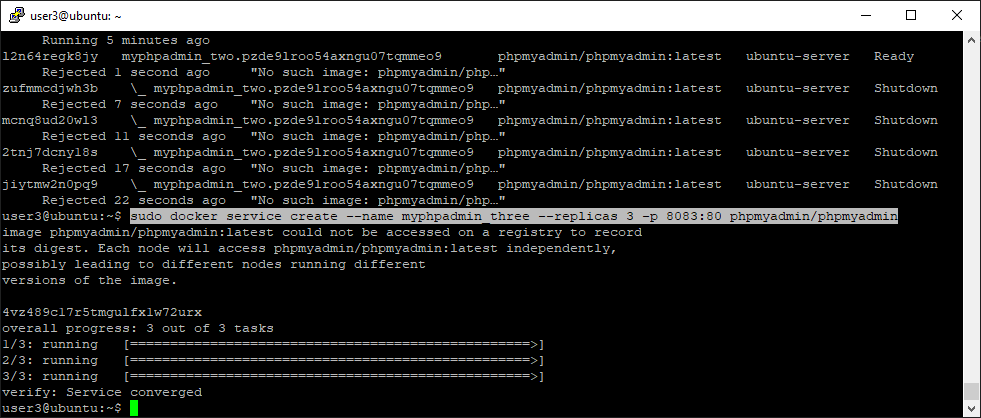




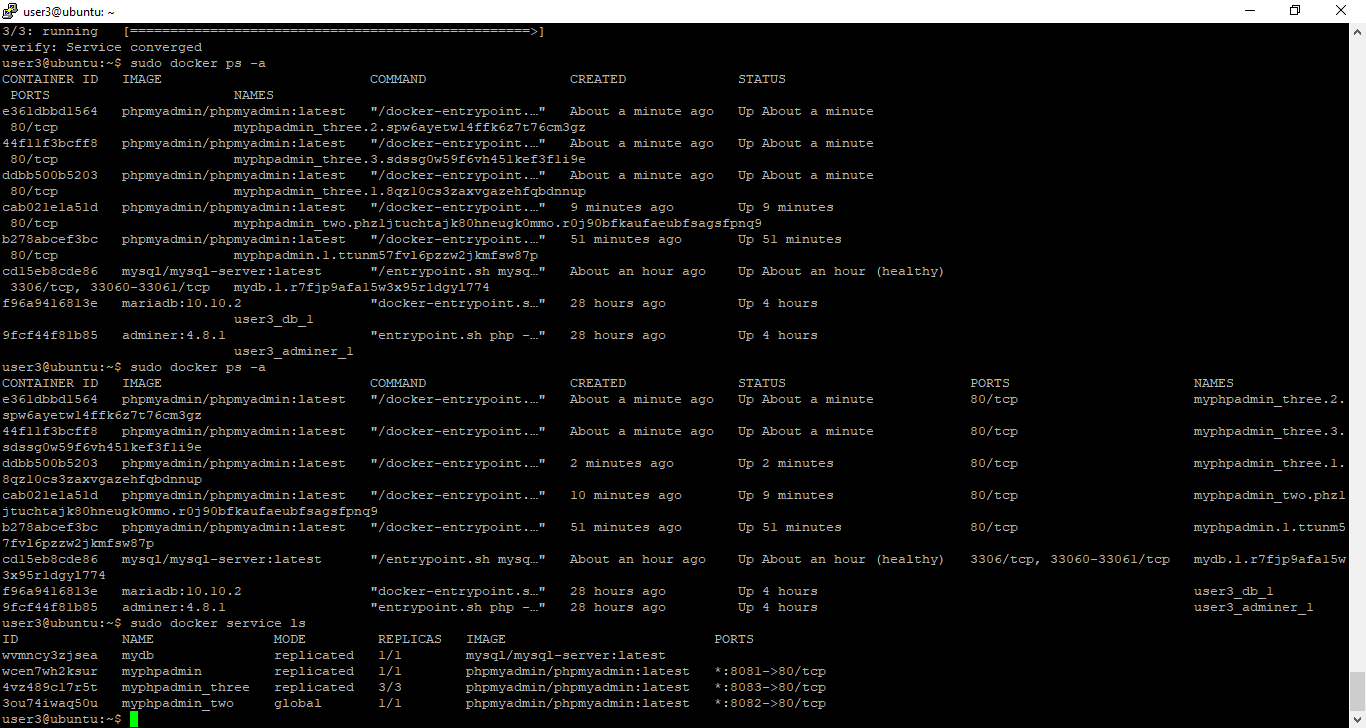
Для запуска сервисов сразу на всех активных нодах используется --mode global (но на рабочей ноде не запустился):



sudo docker service create --name myphpadmin\_three --replicas 3 -p 8083:80 phpmyadmin/phpmyadmin



Все 3 реплики запустились в главной ноде:



2)

mkdir wordpress && cd wordpress

mkdir nginx-conf

nano nginx-conf/nginx.conf

server {

listen 80;

listen [::]:80;

server\_name example.com www.example.com;

index index.php index.html index.htm;

root /var/www/html;

location ~ /.well-known/acme-challenge {

allow all;

root /var/www/html;

}

location / {

try\_files $uri $uri/ /index.php$is\_args$args;

}

location ~ \.php$ {

try\_files $uri =404;

fastcgi\_split\_path\_info ^(.+\.php)(/.+)$;

fastcgi\_pass wordpress:9000;

fastcgi\_index index.php;

include fastcgi\_params;

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

fastcgi\_param PATH\_INFO $fastcgi\_path\_info;

}

location ~ /\.ht {

deny all;

}

location = /favicon.ico {

log\_not\_found off; access\_log off;

}

location = /robots.txt {

log\_not\_found off; access\_log off; allow all;

}

location ~\* \.(css|gif|ico|jpeg|jpg|js|png)$ {

expires max;

log\_not\_found off;

}

}

nano .env

MYSQL\_ROOT\_PASSWORD=123

MYSQL\_USER=user1

MYSQL\_PASSWORD=123

nano .dockerignore

.env

.git

docker-compose.yml

.dockerignore

Установка

sudo curl -L "https://github.com/docker/compose/releases/download/1.26.2/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose

sudo chmod +x /usr/local/bin/docker-compose

nano docker-compose.yml

version: '3'

services:

db:

image: mysql:8.0

container\_name: db

restart: unless-stopped

env\_file: .env

environment:

- MYSQL\_DATABASE=wordpress

volumes:

- dbdata:/var/lib/mysql

command: '--default-authentication-plugin=mysql\_native\_password'

networks:

- app-network

wordpress:

depends\_on:

- db

image: wordpress:5.1.1-fpm-alpine

container\_name: wordpress

restart: unless-stopped

env\_file: .env

environment:

- WORDPRESS\_DB\_HOST=db:3306

- WORDPRESS\_DB\_USER=$MYSQL\_USER

- WORDPRESS\_DB\_PASSWORD=$MYSQL\_PASSWORD

- WORDPRESS\_DB\_NAME=wordpress

volumes:

- wordpress:/var/www/html

networks:

- app-network

webserver:

depends\_on:

- wordpress

image: nginx:1.15.12-alpine

container\_name: webserver

restart: unless-stopped

ports:

- "80:80"

volumes:

- wordpress:/var/www/html

- ./nginx-conf:/etc/nginx/conf.d

- certbot-etc:/etc/letsencrypt

networks:

- app-network

volumes:

certbot-etc:

wordpress:

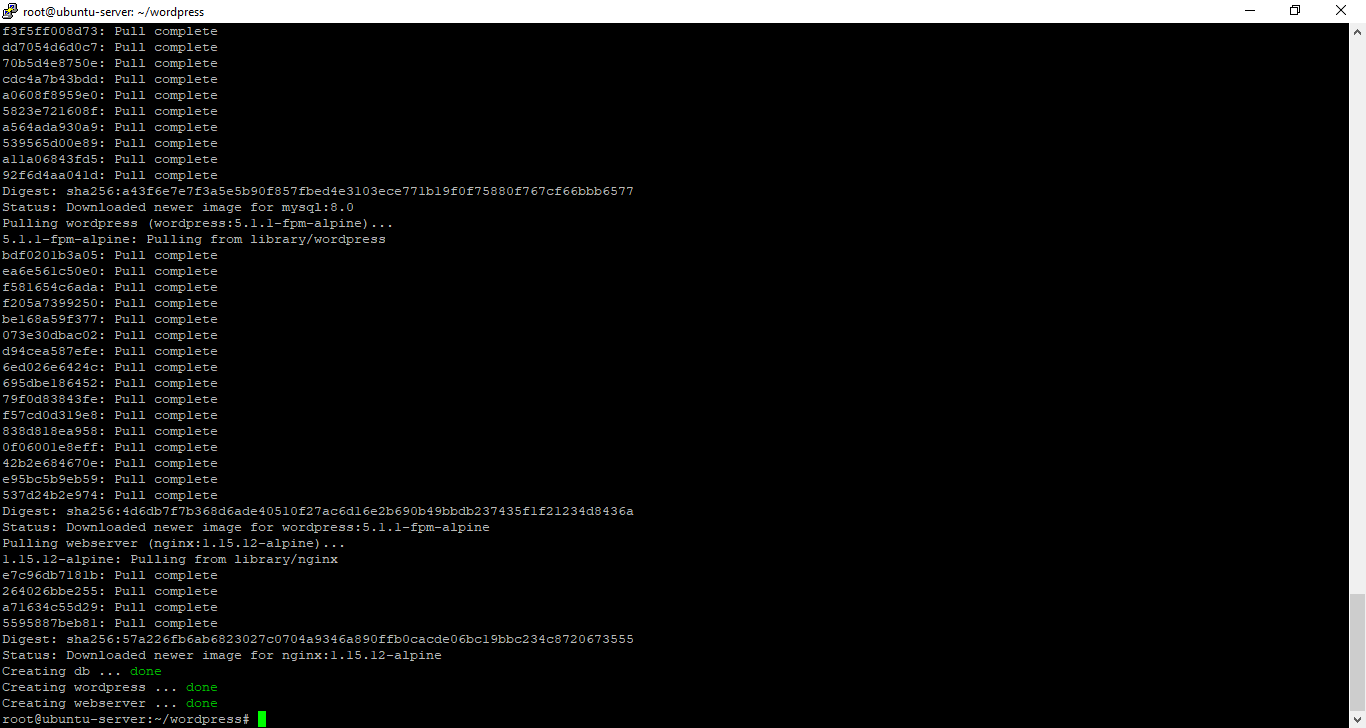
dbdata:

networks:

app-network:

driver: bridge

docker-compose up –d



docker-compose ps

