CentOS iptables – systemctl resrat docker.service

<https://github.com/storm39mad/hi-tech/tree/k8s-labs/04-linux-containers/voting-app-docker>

sudo gpasswd -a user docker

newgrp docker newgrp user

id

**Репозитории** docker login

Установка

<https://docs.docker.com/> - downloaf and install – выбор своей версии

**Удаление прежних версий**

sudo apt-get remove docker docker-engine docker.io containerd runc

sudo apt-get update

sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \

$(lsb\_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

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

sudo docker run hello-world

sudo usermod -aG docker $USER

**Контейнеры** - <https://hub.docker.com/> - Explore

docker run -d имя контейнер – запуск в detached mode.

docker attach контейнер - логин в контейнер

docker ps

docker ps -a

docker stop имя контейнера

docker rm - *удаление контейнера*

docker images - *список образов*

docker rmi образ - удаление образа

docker pull образ - скачать образ без запуска

docker run ubuntu sleep 5- запуск с командой

docker rm $(docker ps -aq) - удаление всех контейнеров

docker run redis:5.0 - tag, верия

docker run -it контейнер - t - привязать терминал, i - привязать stdin

docker run -p 80:5000 контейнер - порт хоста:порт К.

docker run - v /opt/datadir/:/var/lib/mysql mysql - привязка папки

docker run inspect K

docker logs K - журнал из stdout этого К.

Docker tag желаемое имя К:tag К.

docker exec имя контейнера cat /etc/hosts - выполнение команды на работающем контейнере

docker volume create data\_volume - создаст dir в /var/lib/docker/volumes

docker run -v том на хосте:том в К образ

**Создание образа**

Dockerfile.yml

FROM ubuntu

RUN apt-get update - запустить команды в образе

RUN apt install -y python3 python3-pip

RUN pip3 install flask

COPY app.py /opt/app.py

ENTRYPOINT [“sleep”]- команда внутри К, которая запустится, когда образ будет запущен в качестве К.

Задаётся без параметров, в отличии от жёсткого CMD. Если при старте не задать sleep 5, будет ошибка.

CMD – команда внутри. = sleep5 в docker run ubuntu sleep 5.

Либо CMD sleep 5, либо CMD [“command”, “param”]

НУЖНО УКАЗЫВАТЬ ОБЕ

Docker run –entrypoint super-sleep 5 – заменяет entrypoint при запуске.

docker history образ - информация о слоях.

docker build . -t rotoro/vebapp - создание образа (. - использовать dockerfile, лежащий в этой директории, -t – имя создаваемого контейнера)

docker login - залогиниться в docker.hub

docker push куда в ркпозиторий на docker.hub - заливка во внешний репозиторий

**Переменная окружения**

если в скрипте, который запускается внутри К

rocket = os.environ.get('ROCKET\_SIZE')

docker run -e ROCKET\_SIZE=average образ

Проверка - docker inspect

**docker-compose.yml**

ИЗ

docker run -d --name=redis redis

docker run -d --name=db postgres:9.4

docker run -d --name=vote -p5000:80 --link redis:redis voting-app (имя К: имя, по котоорму будет обращаться).

docker run -d --name=result -p 5001:80 --link db:db --link redis:redis result-app

docker run -d --name=worker --link db:db worker

docker-compose up -d --scale vote=3 - запуск с репликами определённой службы

docker-compose down

docker-compose ps

docker-compose logs

**Docker-swarm оркестрация**

Установить docker на всех машинах.

На мастере docker swarm init

На workers docker swarm join \

--tocken <tocken>

docker service create --replicas=3 -p 8080:80 --network frontend образ