DOCKER COMMAND

docker images => oluşturulan image’lerin listesi

docker ps => çalışan docker container’ların listesi

docker pull “service:version” => kullanılacak servis için mage oluşturmak. Docker Hub’da istenilen servisin komutu vardır.

docker run “service:version” => image’I çalıştırmak ve yeni container oluşturmak

docker run –d “service:version” =>detach flagiyle image’in bloklanmadan çalışmasını sağlar ve loglama arka planda yapılır.

docker logs “container id” => container içinde çalışan servislerin log bilgisi gösterilir

docker stop “container id” => container’I durdurmak için kullanılır

docker run-d -p “servisin çalışması istendiği port:servisin varsayılan portu service:version” => p flag’iyle containerın localhostta hangi porta çalışacağı yani port binding işlemi ayarlanır. Örnek: docker run -d -p 9120:80 nginx:1.25.2

docker ps => bir image üzerinde açılmış çoklu containerlar dahil tüm containerları gösterir

docker start “container id”=> durdurulan containerı tekrardan çalıştırmak

docker run --name webapp -d -p 9120:80 nginx:1.25.2 =>yeni container oluştururken container ismi verilmek için –name flag’I kullanılır.

docker network create “networkname” => docker ağı oluşturmak için

docker network ls => docker ağı listesi

docker network rm “networkname” => networkü silmek

FLAG

a- => all

-dit =>

--rm => run command flag ile çalıştırıldıktan sonar tamamlandığında container silinir.

--name => container adı

--network=> hangi docker ağında çalışacağı

-d => detach mod. Log bilgileri çalıştırıldığı terminalde gözükmez

-e => environment flagi

docker version => docker versiyonu

DOCKER COMPOSE

docker-compose –f “yaml. File” up => oluşturulan docker compose dosyasını ayağa kaldırır

docker-compose –f “yaml. File” up –d=> detach modda ayağa kalkar.

docker-compose –f “yaml. File” down =containerları ve oluşturulmuş networkü durdurur ve siler.

docker-compose –f “yaml. File” stop => containerları sadece durdurur.

docker-compose –f “yaml. File” start=> containerları çalıştırır.

docker-compose --project–f “yaml. File” up

version: '3' //Docker-Compose versiyonu

services: //projede kullanılacak servisler

my-app: //custom oluşturulan servis adı

build: . //dockerfile’ın bulunduğu yerde build edilir.

ports: //servis portunun aktarılacağı docker portu

-3000:3000

environment: //servis ayağa kaldırılırken kullanılacak konfigürasyonlar

MONGO\_DB\_USERNAME: admin

MONGO\_DB\_PASSWORD: supersecret

mongodb: //servis adı

image: mongo //image ismi

ports: //servis portunun aktarılacağı docker portu

-27017:27017

environment: //servis ayağa kaldırılırken kullanılacak konfigürasyonlar

MONGO\_INITDB\_ROOT\_USERNAME: admin

MONGO\_INITDB\_ROOT\_PASSWORD: supersecret

mongo-express: //servis ismi

image: mongo-express

ports: //servis portunun aktarılacağı docker portu

-8081:8081

environment: //servis ayağa kaldırılırken kullanılacak konfigürasyonlar

ME\_CONFIG\_MONGODB\_ADMINUSERNAME: admin

ME\_CONFIG\_MONGODB\_ADMINPASSWORD: supersecret

ME\_CONFIG\_MONGODB\_SERVER: mongodb

depends\_on: //srevis ayağa kalkma bağımılılığı.

-“mongodb”

  lvserver:

    build:

      context: .

      dockerfile: Dockerfile

    ports:

      - "9898:9898"

    networks:

      - mongodb\_network

      - influxdb-grafana-trick-integration\_default

networks:

  mongodb\_network:

    external: true

  influxdb-grafana-trick-integration\_default:

    external: true