

İlk Kısım - Dersteki Docker Komutlarının Açıklaması

`sudo docker run [OPTIONS] IMAGE [COMMAND] [ARG...]`

- Docker file çalıştırma komutu
- OPTIONS:
 - -d:
 - docker file'ı arka planda çalıştırır, böylece docker file çalışırken terminali kullanabiliriz
 - --rm:
 - docker file çalışmasını bitirdikten sonra container'ı siler
 - -e:
 - env değişkenleri tanımlama
 - --name:
 - container'a isim verme
- COMMAND:
 - herhangi bir terminal komutu çalıştırılabilir

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker run --rm --name test-container -e MY_ENV_VAR="HelloWorld" alpine sh -c 'echo $MY_ENV_VAR'  
HelloWorld  
emre@emre-VirtualBox:~$
```

`sudo docker ps [OPTIONS]`

- Çalışan container'ları listeler
- OPTIONS:
 - -a:
 - Tüm container'ları listeler

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker ps -a  
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES  
07722805ce6d   alpine    "/bin/sh" 8 minutes ago    Exited (0) 8 minutes ago           affectionate_elba  
kyan  
7447306ae861   hello-world  "/hello"  4 hours ago    Exited (0) 4 hours ago           frosty_leavitt  
28b7028d5ed7   hello-world  "/hello"  4 hours ago    Exited (0) 4 hours ago           stoic_grothendieck  
emre@emre-VirtualBox:~$
```

`sudo docker stop CONTAINER_ID`

- Çalışan bir container'ı durdurur

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker stop 3255b92cd7c1  
3255b92cd7c1  
emre@emre-VirtualBox:~$
```

`sudo docker pull IMAGE_NAME`

- Docker hub'dan image indirme komutu

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker pull alpine  
Using default tag: latest  
latest: Pulling from library/alpine  
c6a83fedfae6: Pull complete  
Digest: sha256:0a4eaa0eecf5f8c050e5bba433f58c052be7587ee8af3e8b3910ef9ab5fbe9f5  
Status: Downloaded newer image for alpine:latest  
docker.io/library/alpine:latest  
emre@emre-VirtualBox:~$
```

`sudo docker images`

- mevcut docker image'lerini listeler

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker images  
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE  
alpine         latest    324bc02ae123   45 hours ago   7.8MB  
hello-world    latest    d2c94e258dcb   14 months ago 13.3kB  
emre@emre-VirtualBox:~$
```

`sudo docker rmi IMAGE_ID`

- mevcut bir docker image'i siler

```
emre@emre-VirtualBox: ~  
emre@emre-VirtualBox:~$ sudo docker rmi d2c94e258dcb --force  
Untagged: hello-world:latest  
Untagged: hello-world@sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc69891e79d84cc4cdceec6  
Deleted: sha256:d2c94e258dcb3c5ac2798d32e1249e42ef01cba4841c2234249495f87264ac5a  
emre@emre-VirtualBox:~$
```

`sudo docker build -t IMAGE_NAME:TAG .`

- docker file'dan image oluşturma komutu

```
emre@emre-VirtualBox: ~/Desktop/dockerlidoockerize
emre@emre-VirtualBox: ~/Desktop/dockerlidoockerize_me/dockerize_me$ sudo docker build -t homework:test .
[+] Building 191.0s (11/11) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 232B
=> [internal] load metadata for docker.io/library/php:7.4-apache
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/6] FROM docker.io/library/php:7.4-apache@sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905 156.4s
=> => resolve docker.io/library/php:7.4-apache@sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905ff 0.0s
=> => sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73deaaab460f8f33ebc6b64e2 31.41MB / 31.41MB 65.5s
=> => sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e877d580bd0ec9e763 226B / 226B 1.5s
=> => sha256:18b3497ee7f2099a90b66c23a0bc3d5261b12bab367263e1b40e9b004c39e882 3.04kB / 3.04kB 0.0s
=> => sha256:c9d7e608f73832673479770d66aacc8100011ec751d1905ff63fae3fe20ca6d 1.86kB / 1.86kB 0.0s
=> => sha256:20a3732f422b7b28dcf99e8597f093a8c135efca62ff0dc02a2d92d916369413 12.51kB / 12.51kB 0.0s
=> => sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376adf9169351a1265382e39de 91.63MB / 91.63MB 144.8s
=> => sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea421750bc9025aface98d8 270B / 270B 2.0s
=> => sha256:25f85b498fd5bfc6cce951513219fe480850daba71e6e997741e984d10483971 19.25MB / 19.25MB 34.3s
=> => sha256:9b233e420ac7bbca645bb82c213029762acf1742400c076360dc303213c309d5 475B / 475B 35.3s
=> => sha256:fe42347c4ecfc9033acd9cad13912387eea39d13827a25cfa78727fa5d200e9 514B / 514B 35.7s
=> => sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829e2cf443fc409342fa 10.76MB / 10.76MB 60.1s
=> => sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171ca5caf81cb8b1c40cb 491B / 491B 61.2s
=> => sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fbb042de5017e43afe8 10.20MB / 10.20MB 79.0s
=> => extracting sha256:a603fa5e3b4127f210503aaa6189abf6286ee5a73deaaab460f8f33ebc6b64e2 3.5s
=> => sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade985fe31e59f15deee8 2.46kB / 2.46kB 65.9s
=> => sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f7662a31498644c9a813ef 246B / 246B 66.3s
=> => sha256:05e465aaa99a358add4acecdade8f39843089069f31fea0201533d3a09a98c9a 892B / 892B 66.7s
=> => extracting sha256:c428f1a494230852524a2a5957cc5199c36c8b403305e0e877d580bd0ec9e763 0.0s
=> => sha256:156740b07ef8a632f9f7bea4e57e4ee5541ade376adf9169351a1265382e39de 7.7s
=> => extracting sha256:fb5a4c8af82f00730b7427e47bda7f76cea2e2b9aea421750bc9025aface98d8 0.0s
=> => extracting sha256:25f85b498fd5bfc6cce951513219fe480850daba71e6e997741e984d10483971 1.3s
=> => extracting sha256:9b233e420ac7bbca645bb82c213029762acf1742400c076360dc303213c309d5 0.0s
=> => extracting sha256:fe42347c4ecfc9033acd9cad13912387eea39d13827a25cfa78727fa5d200e9 0.0s
=> => extracting sha256:d14eb2ed1e17ae00f5fcb44b0d562e2867c401c20372829e2cf443fc409342fa 0.3s
=> => extracting sha256:66d98f73acb62e86c0c226f9eedcbc7eda305df0c1e171ca5caf81cb8b1c40cb 0.0s
=> => extracting sha256:d2c43c5efbc861f83ee6565c7102ca660d6f35e158324fbb042de5017e43afe8 1.1s
=> => sha256:ab590b48ea476386dd7b07c34de9eff7cf2103c4668ade985fe31e59f15deee8 0.0s
=> => extracting sha256:80692ae2d067c8358112c56490a2a97f69ef395fd8f7662a31498644c9a813ef 0.0s
=> => extracting sha256:05e465aaa99a358add4acecdade8f39843089069f31fea0201533d3a09a98c9a 0.0s
=> [internal] load build context
=> => transferring context: 1.58kB
=> [2/6] WORKDIR /var/www/html
=> [3/6] COPY ./app .
=> [4/6] RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
=> [5/6] RUN apt-get update
=> [6/6] RUN docker-php-ext-install pdo pdo_mysql
=> exporting to image
=> => exporting layers
=> => writing image sha256:2e149a457a15c1d061efe13301b847ebef2df4f6bd9e7b319c897cfba90cf954
=> => naming to docker.io/library/homework:test
emre@emre-VirtualBox: ~/Desktop/dockerlidoockerize_me/dockerize_me$
```

sudo docker push IMAGE_NAME:TAG

- docker hub'a image yükleme komutu

sudo docker exec -it CONTAINER_ID COMMAND

- Çalışan bir container'da komut çalıştırmamızı sağlar

```
emre@emre-VirtualBox: ~
emre@emre-VirtualBox: ~$ sudo docker exec -it 4a832c4d23a3 pwd
/
emre@emre-VirtualBox: ~$ sudo docker exec -it 4a832c4d23a3 ls
```

sudo docker version

- docker versiyonunu döner

```
emre@emre-VirtualBox: ~
emre@emre-VirtualBox: ~$ docker -v
Docker version 27.1.1, build 6312585
emre@emre-VirtualBox: ~$
```

sudo docker-compose [OPTIONS] [SERVICES]

- OPTIONS:
 - up:
 - docker-compose.yml file'ındaki servisleri çalıştırır
 - down:
 - docker-compose.yml file'ındaki çalışan servisleri durdurur
 - build:
 - docker-compose.yml file'ındaki servislerden image oluşturur
 - ps:
 - docker-compose.yml file'ındaki çalışan servisleri listeler

```
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$ sudo docker compose -f ./docker-compose.yml down
WARN[0000] /home/emre/Desktop/dockerlidocontainerize_me/dockerize_me/docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confusion
[+] Running 3/3
✓ Container dockerize_me-app-1 Removed 0.0s
✓ Container dockerize_me-db-1 Removed 0.0s
✓ Network dockerize_me_net Removed 0.3s
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$
```

sudo docker network ls

- Docker network'lerini listeler

```
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$ sudo docker network ls
NETWORK ID      NAME      DRIVER      SCOPE
c35c0b9f405c    bridge   bridge      local
4b909b1b260d    host     host         local
c7378d8ba2ce    none     null         local
```

sudo docker network create NETWORK_NAME

- docker network oluşturur

sudo docker network rm NETWORK_NAME

- docker network silme komutu

```
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$ sudo docker network create homework
d4e7fd0d39847f247b9733d67fad72f6cb3197097c65bf8d74bf425e910b897c
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$ sudo docker network rm homework
homework
emre@emre-VirtualBox: ~/Desktop/dockerlidocontainerize_me/dockerize_me$
```

sudo docker volume ls

- docker volume'lerini listeleme(volume: backup tarzı bir olay)

sudo docker volume create VOLUME_NAME

- docker volume oluşturma komutu

sudo docker volume rm VOLUME_NAME

- docker volume silme komutu

```
emre@emre-VirtualBox: ~/Desktop/dockerlidockerize_me/dockerize_me
emre@emre-VirtualBox: ~/Desktop/dockerlidockerize_me/dockerize_me$ sudo docker volume ls
DRIVER      VOLUME NAME
local       dockerize_me_db_data
emre@emre-VirtualBox: ~/Desktop/dockerlidockerize_me/dockerize_me$ sudo docker volume create homework
homework
emre@emre-VirtualBox: ~/Desktop/dockerlidockerize_me/dockerize_me$ sudo docker volume rm homework
homework
emre@emre-VirtualBox: ~/Desktop/dockerlidockerize_me/dockerize_me$
```

İkinci Kısım - Gruba Atılan Dosya İçin Açıklamalar

Docker File

FROM php:7.4-apache

- Base image'i belirtmek için yazılan satır

WORKDIR /var/www/html

- diğer tüm komutların çalışacağı directory'i belirtiyoruz

COPY ./app .

- Ana host'taki directory ve file'ları, container'ın file sistemine yükler, yani üstte belirttiğimiz container'ın /var/www/html dizini

RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf

- Container'daki apache konfigurasyonunda ServerName'i localhost olarak ayarlıyoruz, servername localhost olmayınca hata aldığımız için

RUN apt-get update

- container'daki paketleri günceller, amaç: kullanılan paketlerin son versiyonlarını kullanmak

RUN docker-php-ext-install pdo pdo_mysql

- php'nin mysql ile iletişim kurması için gerekli extention'ların kurulumu

EXPOSE 80

- container 80 port'unu dinlemeye başlar ama server 80 portunda başlatılmaz, container çalıştırılacağında -p argümanı ile host için belirleyeceğimiz port'ta başlatılır.

docker-compose.yml

version: '3'

- docker-compose file format'ını belirtir

```
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
    depends_on:
      - db
    ports:
      - "80:80"
    networks:
      - net
```

- **services içinde**, hangi container'ların çalışacağını belirtiyoruz, buna göre sadece app container'ı çalıştırılacak
 - **app** -> servisin adı
 - **build içinde**, app servisi için nasıl image build(oluşturmak) edileceği belirtiliyor, buna göre bulunan directory'deki Dockerfile'a göre oluşturulacak
 - **depends_on içinde**, app servisinin hangi servisler çalıştıktan sonra çalışacağını belirtiyoruz, buna göre db servisi başladıktan sonra app servisi başlatılacak
 - **ports içinde**, host:container şeklinde, app servisinin host'ta çalışacağı port'u belirtiyoruz, buna göre app servisi host'un 80 port'unda çalışacak
 - **networks içinde**, app servisinin hangi network'e bağlanacağı belirtiliyor, buna göre app servisi net network'üne bağlanacak

```
db:
  image: mysql:latest
  environment:
    - MYSQL_DATABASE=yavuzlar
    - MYSQL_ROOT_PASSWORD=1
  volumes:
    - db_data:/var/lib/mysql
    - ./yavuzlar_messages.sql:/docker-entrypoint-initdb.d/yavuzlar_messages.sql
  ports:
    - "8080:3306"
  networks:
    - net
```

- .
 - **db** -> servisin adı
 - **image** ile db servisinde kullanılacak image'ı belirtiyoruz, buna göre db servisinde kullanılacak image mysql
 - **environment** ile environment variable'larını belirtiyoruz, buna göre db servisi başlatıldığında yavuzlar adında bir db oluşturulacak ve root password'u 1 olacak
 - **volumes** ile volume belirtilir; buna göre yavuzlar_messages.sql file, host /var/lib/mysql içinde de tutulacak, yapılan her değişiklikten etkilenecek şekilde

- **ports** ile host:container şeklinde, db servisinin host'ta çalışacağı port belirleniyor, buna göre db servisi host üzerinde 8080 portunda çalışacak
- **networks**, db servisinin hangi network'e bağlanacağı belirtiliyor, buna göre app servisi net network'üne bağlanacak

```
networks:  
  net:  
    driver: bridge
```

- **networks** ile servisler için özel bir network oluşturuluyor
 - **net** -> network'ün adı
 - **driver** ile kullanılacak network sürücüsünü belirtiyoruz, buna göre network'te bridge driver kullanılacak, bu servislerin birbirleri ile private bir network'te iletişim kurmasına olanak sağlar

```
volumes:  
  db_data:
```

- db_data adında volume oluşturuyoruz, böylece container'lar kapansa, durdurulsa bile data kaybının önüne geçiyoruz