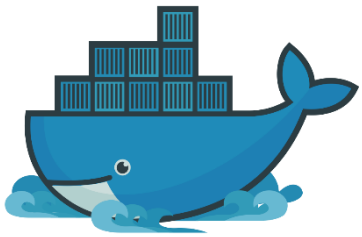
1,2 ve 3üncü maddeleri sırasıyla aynı pdf içerisinde yaptım.



DOCKER KOMUTLARI

İlk öncelikle ödevi hiç bilmeyen birinin okuduğunda neler yapacağını tam manasıyla kavrayıp yapabilmesini düşünerek yapacağım.

DOCKER NEDİR?

Docker, yazılım uygulamalarını ve onların bağımlılıklarını izole edilmiş ve taşınabilir konteynerler içinde çalıştırmayı sağlayan bir platformdur. Konteyner ise yazılım uygulamalarını ve onların bağımlılıklarını bir araya getirip izole eden, hafif ve taşınabilir bir çalışma ortamıdır.

DOCKER KOMUTLARI:

- docker --version

  Bu komut, Docker'ın hangi sürümünün sisteminizde kurulu olduğunu hızlıca öğrenmenizi sağlar.

```
C:\Users\huriy>docker --version
Docker version 26.1.4, build 5650f9b
```

- docker info

  Docker kurulumunuz hakkında detaylı bilgi sağlar. Bu komut, Docker daemon'u, konteynerlar, görüntüler, ağlar ve daha fazlası hakkında kapsamlı bir genel bakış sunar.

```
C:\Users\huriy>docker info
Client:
 Version:    26.1.4
 Context:    desktop-linux
 Debug Mode: false
 Plugins:
  buildx: Docker Buildx (Docker Inc.)
    Version:  v0.14.1-desktop.1
    Path:     C:\Program Files\Docker\cli-plugins\docker-buildx.exe
  compose: Docker Compose (Docker Inc.)
    Version:  v2.27.1-desktop.1
    Path:     C:\Program Files\Docker\cli-plugins\docker-compose.exe
  debug: Get a shell into any image or container (Docker Inc.)
    Version:  0.0.32
    Path:     C:\Program Files\Docker\cli-plugins\docker-debug.exe
  dev: Docker Dev Environments (Docker Inc.)
    Version:  v0.1.2
    Path:     C:\Program Files\Docker\cli-plugins\docker-dev.exe
  extension: Manages Docker extensions (Docker Inc.)
    Version:  v0.2.24
    Path:     C:\Program Files\Docker\cli-plugins\docker-extension.exe
  feedback: Provide feedback, right in your terminal! (Docker Inc.)
    Version:  v1.0.5
    Path:     C:\Program Files\Docker\cli-plugins\docker-feedback.exe
  init: Creates Docker-related starter files for your project (Docker Inc.)
    Version:  v1.2.0
    Path:     C:\Program Files\Docker\cli-plugins\docker-init.exe
```

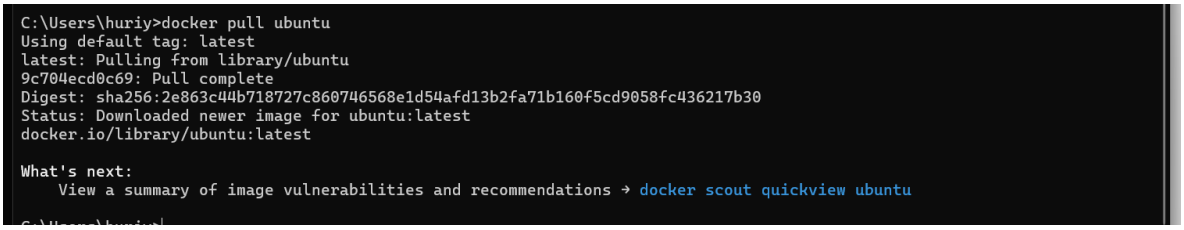
- docker search [anahtar kelime]

  Docker Hub'da belirli bir anahtar kelimeyle ilişkili resmi ve topluluk tarafından oluşturulmuş Docker görüntülerini aramak için kullanılır.
  Mesela, docker search alpine bu komut Bu komut, Alpine Linux tabanlı görüntüleri hızlıca bulmanıza ve ihtiyacınıza uygun olanı seçmenize yardımcı olur.

```
C:\Users\huriy>docker search alpine
NAME                                      DESCRIPTION                                     STARS     OFFICIAL
alpine                                    A minimal Docker image based on Alpine Linux…   10941     [OK]
alpinelinux/docker-cli                    Simple and lightweight Alpine Linux image wi…   11
alpinelinux/alpine-gitlab-ci              Build Alpine Linux packages with Gitlab CI      3
alpinelinux/gitlab-runner-helper          Helper image container gitlab-runner-helper …   7
alpinelinux/unbound                                                                       13
alpinelinux/rsyncd                                                                        2
alpinelinux/alpine-drone-ci               Build Alpine Linux packages with drone CI       0
alpinelinux/docker-alpine                                                                 0
alpinelinux/ansible                       Ansible in docker                               21
alpinelinux/gitlab-runner                 Alpine Linux gitlab-runner (supports more ar…   7
grafana/alpine                            Alpine Linux with ca-certificates package in…   7
alpinelinux/docker-compose                docker-compose image based on Alpine Linux      2
alpinelinux/apkbuild-lint-tools           Tools for linting APKBUILD files in a CI env…   0
bellsoft/liberica-openjdk-alpine          Liberica is a 100% open-source Java implemen…   58
alpinelinux/darkhttpd                                                                     2
alpinelinux/golang                        Build container for golang based on Alpine L…   3
alpinelinux/alpine-docker-gitlab          Gitlab running on Alpine Linux                  0
alpinelinux/build-base                    Base image suitable for building packages wi…   0
alpinelinux/alpine-www                    The Alpine Linux public website (www.alpinel…   0
bellsoft/liberica-openjre-alpine          Liberica is a 100% open-source Java implemen…   22
alpinelinux/docker-abuild                 Dockerised abuild                               0
bellsoft/liberica-openjdk-alpine-musl     Liberica is a 100% open-source Java implemen…   24
```

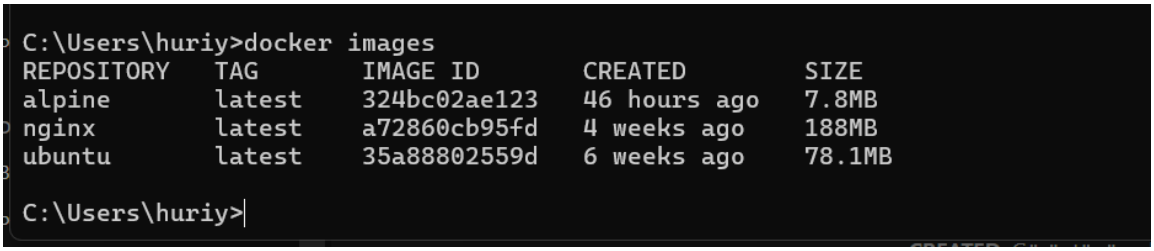
- docker pull <image_name>

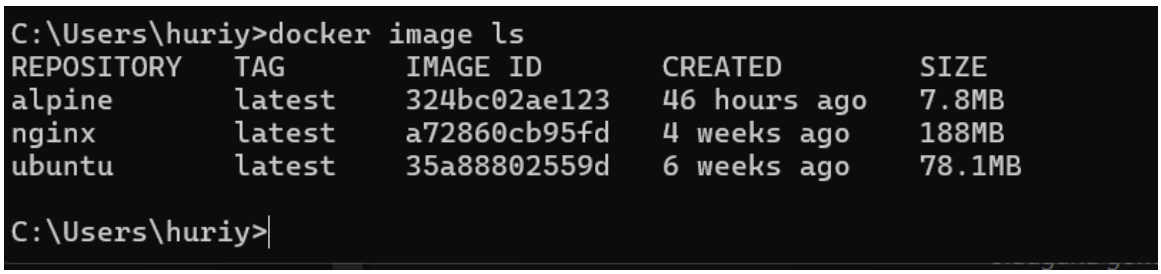
  Docker görüntüsünü Docker Hub veya başka bir Docker kayıt defterinden yerel sisteminize indirmek için kullanılır. Bu komut, belirtilen görüntünün en son sürümünü veya belirli bir sürümünü (tag) indirmenizi sağlar. İndirilen görüntü, daha sonra konteyner oluşturmak için kullanılabilir. Mesela, docker pull Ubuntu Docker Hub'dan Ubuntu'nun en son sürümünü indirir. Belirli bir sürümü indirmek için ise komutun yanına istediğim versiyonu yazarım.



- docker images

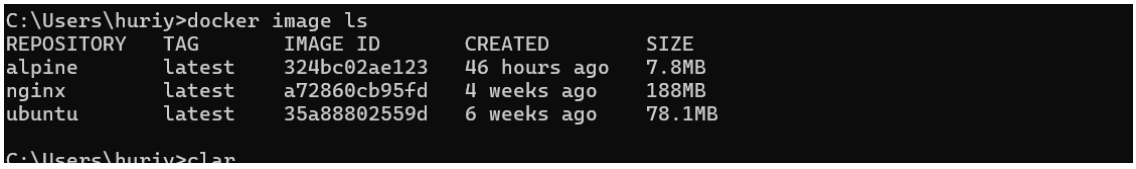
  Docker ortamınızdaki tüm Docker görüntülerini (images) listelemek için kullanılır. Bu komut, her görüntünün ismini, etiketini (tag), görüntü ID'sini, oluşturulma tarihini ve boyutunu gösterir bize. docker images ls komutu, docker images komutunun bir kısayolu olarak çalışır ve yerel Docker ortamınızdaki tüm Docker görüntülerini listelemek için kullanılır. Her iki komut da aynı işlevi yerine getirir ve aynı bilgileri sağlar.





- docker rmi <image_id>

  Belirli bir Docker görüntüsünü (image) yerel sisteminizden kaldırmak için kullanılır. Silmeyi göstermek için ilk önce image leri gösterdim. Sonrasın da alpineyi siliyorum ama silerken id sini kullanıyorum. Aşağıdaki fotoğrafta bulabilirsiniz.



- docker build -t [image_name]

  Docker görüntüsünü (image) oluştururken ona bir isim ve isteğe bağlı olarak bir etiket (tag) atamak için kullanılır. `-t` bayrağı, oluşturulan görüntüye bir etiket verir, bu da görüntüyü daha sonra kolayca tanımlamanızı ve yönetmenizi sağlar. Geçerli dizindeki Dockerfile'ı kullanarak bir görüntü oluşturur.
- docker ps -a

  Docker ortamınızdaki tüm konteynerleri listeler sadece çalışan konteynerleri değil, aynı zamanda sonlandırılmış ve durdurulmuş konteynerleri de gösterir.

```
C:\Users\huriy>docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

  Gördüğümüz gibi hiç konteyner yok. İlerideki komutlarda oluşturacağız.
- docker ps

  Docker ortamınızdaki sadece aktif (çalışan) konteynerleri listeler. Bu komut, çalışmakta olan konteynerlerin temel bilgilerini gösterir. Benim şuan konteyner ım olmadığı için ileride tekrardn gösteririm.

```
C:\Users\huriy>docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

- docker inspect <image_id>

  Belirli bir Docker görüntüsü (image) hakkında ayrıntılı bilgi almak için kullanılır.

```
C:\Users\huriy>docker image inspect 35a88802559d
[
    {
        "Id": "sha256:35a88802559dd2077e584394471ddaa1a2c5bfd16893b829ea57619301eb3908",
        "RepoTags": [
            "ubuntu:latest"
        ],
        "RepoDigests": [
            "ubuntu@sha256:2e863c44b718727c860746568e1d54afd13b2fa71b160f5cd9058fc436217b30"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2024-06-07T12:00:09.099611108Z",
        "Container": "9e2f0a4a3e3829e7bf3c12134cfd03238aec6ca2ce5029cf0197627a3b37621e",
        "ContainerConfig": {
            "Hostname": "9e2f0a4a3e38",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
```

- docker run <image_name>

  Docker görüntüsünden (image) bir konteyner oluşturur ve çalıştırır. Bu komut, görüntüyü kullanarak yeni bir konteyner başlatır ve çalıştırır. Şimdi Ubuntu konteyner ını oluşturup konteyner ları görüntüleyeceğim.

```
C:\Users\huriy>docker run ubuntu

C:\Users\huriy>docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS                     PORTS     NAMES
1563fd9fa40e   ubuntu    "/bin/bash"   7 seconds ago   Exited (0) 6 seconds ago             funny_matsumoto
c6f52ce5c63e   ubuntu    "/bin/bash"   2 minutes ago   Exited (0) 2 minutes ago             beautiful_knuth

C:\Users\huriy>
```

- docker run -d <image_name>

  Belirtilen Docker görüntüsünden bir konteyner oluşturur ve bunu arka planda çalıştırır. `-d` bayrağı, konteynerin "detached" (arka plan) modda çalışmasını sağlar, yani terminali kapatmadan çalışmaya devam eder.

```
C:\Users\huriy>docker run -d ubuntu
02ad16bc71fa5121d5c07716e8b4656a2df14dd8c818b987c58381df714ff2dc
```

- docker run <image name> echo ...

  Docker görüntüsünden bir konteyner oluşturur ve içinde echo .... komutunu çalıştırır.

```
C:\Users\huriy>docker run alpine echo Merhaba Yavuzlar!!
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
c6a83fedfae6: Pull complete
Digest: sha256:0a4eaa0eecf5f8c050e5bba433f58c052be7587ee8af3e8b3910ef9ab5fbe9f5
Status: Downloaded newer image for alpine:latest
Merhaba Yavuzlar!!
```

- docker run --rm alpine echo selamlar

  Alpine görüntüsünden bir konteyner oluşturur, içinde echo selamlar komutunu çalıştırır ve işlem tamamlandıktan sonra konteyneri otomatik olarak siler.

```
C:\Users\huriy>docker run --rm alpine echo selamlar
selamlar
```

- docker run --name <container_name> <image_name>

  Ubuntu adlı Docker görüntüsünden bir konteyner oluşturur ve bunu arka planda çalıştırır. Konteynere my_ubuntu_container adını verir.

```
C:\Users\huriy>docker run -d --name my_ubuntu_container2 ubuntu
4633ed4e4906daf0c38bb3eecc75e456e8781a7e1d58c78453ac53fc53cfdaa2
```

- docker run -it --rm alpine

  Docker görüntüsünden bir konteyner oluşturur ve interaktif bir terminal açar. Konteyner çalışmayı tamamladıktan sonra otomatik olarak silinir. Bu komut, genellikle kısa süreli testler ve etkileşimli işlemler için kullanılır.

```
Microsoft Windows [Version 10.0.22631.3880]
(c) Microsoft Corporation. Tüm hakları saklıdır.

C:\Users\huriy>docker run -it --rm alpine
/ #
```

Komut çalıştırıldığında, terminal ekranında şu şekilde bir ekran çıkar.

Buradan, Alpine Linux içinde çeşitli komutları çalıştırabilirsiniz.

- docker exec -it my_container /bin/bash

  Docker konteyneri içinde interaktif bir bash shell başlatır. Bu komut, mevcut bir konteynerin içinde komutlar çalıştırmak için kullanılır. Önce, çalışmakta olan konteynerlerinizi listeleriz.

- docker stop <container_id_or_name>

  Belirtilen konteyneri durdurur.

```
C:\Users\huriy>docker stop 648640cfa73e
648640cfa73e
```

```
C:\Users\huriy>docker network rm host
Error response from daemon: host is a pre-defined network and cannot be removed

C:\Users\huriy>docker network rm none
Error response from daemon: none is a pre-defined network and cannot be removed

C:\Users\huriy>docker network rm my_network
my_network
C:\Users\huriy>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
8521dd1fbbfc   bridge    bridge    local
4e003e9ec949   host      host      local
0f5af28eb926   none      null      local
```

- docker start <container_id_or_name>
  Durdurulmuş bir Docker konteynerini yeniden başlatır. Bu komut, konteynerin daha önce durdurulduğu durumda olduğu gibi çalışmaya devam etmesini sağlar.

```
C:\Users\huriy>docker start 4633ed
4633ed

C:\Users\huriy>docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED          STATUS                      PORTS
4633ed4e4906   ubuntu    "/bin/bash"              46 minutes ago   Exited (0) 11 seconds ago
iner2
6abc16153d2e   ubuntu    "komutu, ubuntu adlı…"   57 minutes ago   Created
iner
f0b079294515   alpine    "/bin/sh"                2 hours ago      Exited (0) 2 hours ago
5c91ba532a6e   alpine    "echo Merhaba Yavuzl…"   2 hours ago      Exited (0) 2 hours ago
n
02ad16bc71fa   ubuntu    "/bin/bash"              2 hours ago      Exited (0) 2 hours ago
and
f21b2f460b14   nginx     "/docker-entrypoint.…"   2 hours ago      Exited (0) 2 hours ago
n
1563fd9fa40e   ubuntu    "/bin/bash"              2 hours ago      Exited (0) 2 hours ago
c6f52ce5c63e   ubuntu    "/bin/bash"              2 hours ago      Exited (0) 2 hours ago
```

- docker rm <container_id_or_name>
  Bir Docker konteynerini tamamen siler. Bu komut, durdurulmuş veya artık ihtiyaç duyulmayan konteynerleri sistemden kaldırmak için kullanılır.
- docker network ls
  Docker ağlarını listelemek için kullanılır. Bu komut, mevcut Docker ağlarını ve bunlarla ilgili temel bilgileri gösterir.

```
C:\Users\huriy>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
8521dd1fbbfc   bridge    bridge    local
4e003e9ec949   host      host      local
0f5af28eb926   none      null      local

C:\Users\huriy>
```

- docker network create <network_name>
  Docker'da yeni bir ağ oluşturur. Bu ağ, Docker konteynerlerinin birbirleriyle veya dış dünya ile iletişim kurması için kullanılır.

- docker network rm <network_name>
  Docker'da mevcut bir ağı silmek için kullanılır

```
C:\Users\huriy>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
8521dd1fbbfc   bridge    bridge    local
4e003e9ec949   host      host      local
0f5af28eb926   none      null      local

C:\Users\huriy>docker network create my_network
13a32fa6c5e50f6aa786a4a3c6b8ce55091a788faacb3315733b482341024d0e

C:\Users\huriy>
```


- docker volume ls

Docker'da mevcut olan tüm hacimleri (volumes) listelemek için kullanılır. Docker hacimleri, konteynerler arasında veri paylaşımını ve kalıcı veri depolamayı sağlamak için kullanılır.

```
C:\Users\huriy>docker volume ls
DRIVER    VOLUME NAME

C:\Users\huriy>
```

Şuan herhangi bir volüme yok.

- docker volume create <volume_name>

Docker'da yeni bir hacim (volume) oluşturur.

```
C:\Users\huriy>docker volume create dosya1
dosya1

C:\Users\huriy>docker volume ls
DRIVER    VOLUME NAME
local     dosya1
```

- docker volume rm <volume_name>

Belirli bir Docker hacmini (volume) silmek için kullanılır. Bu işlem hacmi tamamen kaldırır ve ilgili verileri temizler.

```
C:\Users\huriy>docker volume rm dosya1
dosya1

C:\Users\huriy>docker volume ls
DRIVER    VOLUME NAME

C:\Users\huriy>
```

Dosya silinmiş olarak gözüküyor.

- docker-compose up

Docker Compose dosyasındaki tanımlamalara göre konteynerleri başlatır ve çalıştırır.

- docker-compose down

Docker Compose uygulamasını durdurur ve tüm oluşturulmuş olan konteynerleri, ağları ve varsayılan olarak volümleri siler. Bu, Docker Compose tarafından tanımlanan tüm kaynakları temizlemek için kullanılır.

- docker-compose down -v

compase uygulamasını durduruken -v ile geçmemiz

- docker volume inspect <volume_name>

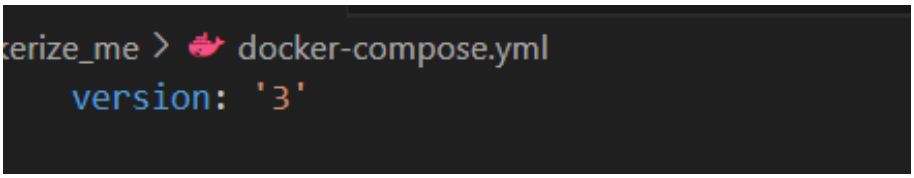
Bir volume'ı incelemek için bu komutunu kullanabilirim.

- docker run -e ENV_VAR=value my_image

Docker komutları genellikle kısa form seçeneklerle kullanılır ve bu seçenekler bazı komutların kısa yollarını içerebilir.

DOCKERFİLE VE DOCKER COMPOSE DOSYALARI AÇIKLAMA

- Bu satır, Docker Compose dosyasının 3. versiyonunu kullandığını belirtir.



- Services bölümü, uygulamanızı oluşturan farklı servisleri tanımlar.

```
services:
  app:
    build:
      context: .
      dockerfile: Dockerfile
```

build: Bu bölüm, app servisi için imajı, mevcut dizinde (.) bulunan Dockerfile'dan oluşturması gerektiğini Docker'a söyler. **context:** İmajın oluşturulacağı bağlamın yolu. **dockerfile:** İmajı oluşturmak için kullanılacak Dockerfile'ın adı.

- **depends_on:** Bu, app servisinin db servisinden sonra başlatılmasını sağlar.

```
    depends_on:
      - db
```

- **ports:** Bu, ana makinadaki 80 numaralı portu konteynerdaki 80 numaralı porta bağlar, böylece uygulama `http://localhost:80` üzerinden erişilebilir hale gelir.

```
    ports:
      - "80:80"
```

- **networks:** Bu, app servisini net ağına bağlar.

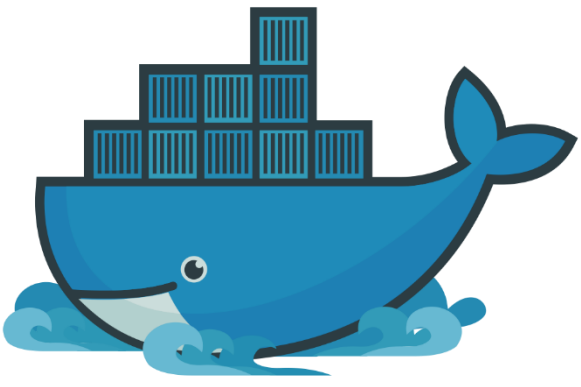
```
    networks:
      - net
```

- **image:** Bu, `db` servisi için kullanılacak Docker imajını belirtir. Burada, MySQL'in en son sürümü kullanılıyor.

```
  db:
    image: mysql:latest
```

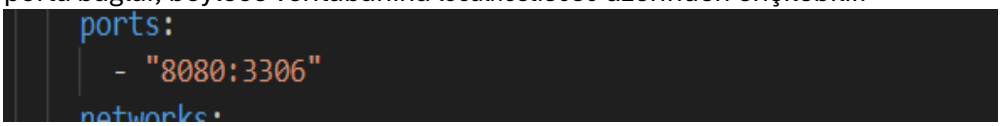
- **environment:** Bunlar, MySQL veritabanını yapılandırmak için kullanılan ortam değişkenleridir.
MYSQL_DATABASE=yavuzlar: yavuzlar adında bir veritabanı oluşturur.MYSQL_ROOT_PASSWORD=1: MySQL için root şifresini `1` olarak ayarlar.

```
      environment:
        - MYSQL_DATABASE=yavuzlar
        - MYSQL_ROOT_PASSWORD=1
```

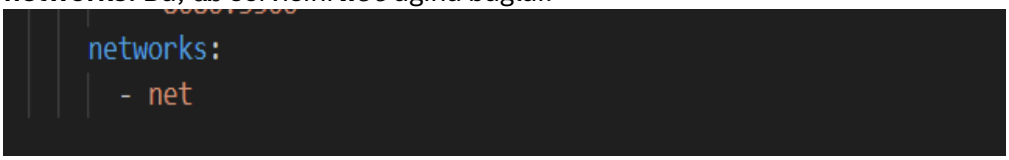


- **volumes:** Bu, veri kalıcılığı ve başlangıç verileri için hacimleri bağlar. db_data:/var/lib/mysql: Veritabanı verilerini db_data hacmine kalıcı olarak kaydeder../yavuzlar_messages.sql:/docker-entrypoint-initdb.d/yavuzlar_messages.sql: yavuzlar_messages.sql scriptini konteynıra kopyalar ve bu script, veritabanını başlatmak için çalıştırılır.

- **ports:** Bu, ana makinadaki 8080 numaralı portu konteynerdaki 3306 numaralı porta bağlar, böylece veritabanına localhost:8080 üzerinden erişilebilir.



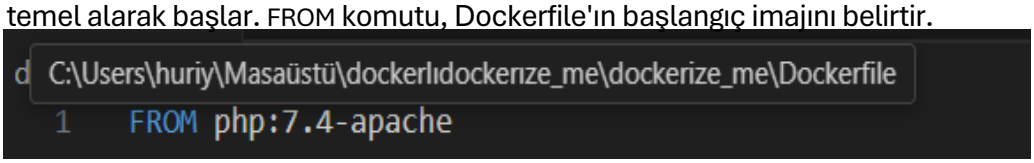
- **networks:** Bu, `db` servisini `net` ağına bağlar.



- Bu, servislerin birbirleriyle iletişim kurmasını sağlayan `net` adlı bir köprü ağı tanımlar.

- Bu, MySQL veritabanı verilerini kalıcı olarak saklamak için db_data adlı bir isimlendirilmiş hacim tanımlar.

- Bu satır, PHP 7.4 ve Apache web sunucusunu içeren resmi bir Docker imajını temel alarak başlar. FROM komutu, Dockerfile'ın başlangıç imajını belirtir.



- Bu satır, çalışma dizinini /var/www/html olarak ayarlar. Docker konteyneri bu dizinde çalışacaktır. Apache web sunucusu varsayılan olarak bu dizinde belgeleri sunar.

- Bu satır, yerel makinedeki ./app dizinindeki tüm dosyaları Docker konteynerindeki /var/www/html dizinine kopyalar. Bu, uygulama dosyalarınızı konteynerin web kök dizinine yerleştirir.

```
4   COPY ./app .
```

- Bu satır, Apache yapılandırma dosyasına ServerName localhost satırını ekler. Bu, Apache'ye hangi sunucu adını kullanacağını söyler ve bazı olası uyarıları önler.

- Bu satır, Debian tabanlı sistemler için paket listesini günceller. Bu, daha sonra paket yüklerken en güncel listeleri kullanmanızı sağlar.

```
5   RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
6   RUN apt-get update
```

- Bu satır, PHP'nin PDO (PHP Data Objects) ve PDO_MySQL uzantılarını yükler. Bu uzantılar, PHP'nin MySQL veritabanlarına erişmesine ve onlarla çalışmasına izin verir.

```
RUN docker-php-ext-install pdo pdo_mysql
```

- Bu satır, Docker konteynerinin 80 numaralı portunu açar. 80 numaralı port, HTTP web trafiği için standart porttur ve bu port üzerinden gelen trafiği kabul etmek için konteyneri yapılamdırdım.

```
EXPOSE 80
```