Eren Ersoyluoğlu

```
keofla — docker run --name Yavuzlar --rm -p 8080:80 nginx — 132×44
Last login: Wed Jul 24 23:25:23 on console
keofla@keofla ~ % docker run --name Yavuzlar --rm -p 8080:80 nginx
/docker-entrypoint.sh: /docker-entrypoint.d/ is not empty, will attempt to perform configuration
/docker-entrypoint.sh: Looking for shell scripts in /docker-entrypoint.d/
/docker-entrypoint.sh: Launching /docker-entrypoint.d/10-listen-on-ipv6-by-default.sh
10-listen-on-ipv6-by-default.sh: info: Getting the checksum of /etc/nginx/conf.d/default.conf
10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
/docker-entrypoint.sh: Configuration complete; ready for start up
2024/07/24 20:45:23 [notice] 1#1: using the "epoll" event method
2024/07/24 20:45:23 [notice] 1#1: nginx/1.27.0
2024/07/24 20:45:23 [notice] 1#1: built by gcc 12.2.0 (Debian 12.2.0-14)
2024/07/24 20:45:23 [notice] 1#1: OS: Linux 6.6.31-linuxkit
2024/07/24 20:45:23 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2024/07/24 20:45:23 [notice] 1#1: start worker processes
2024/07/24 20:45:23 [notice] 1#1: start worker process 29
2024/07/24 20:45:23 [notice] 1#1: start worker process 30
2024/07/24 20:45:23 [notice] 1#1: start worker process 31
2024/07/24 20:45:23 [notice] 1#1: start worker process 32
2024/07/24 20:45:23 [notice] 1#1: start worker process 33
2024/07/24 20:45:23 [notice] 1#1: start worker process 34
2024/07/24 20:45:23 [notice] 1#1: start worker process 35
2024/07/24 20:45:23 [notice] 1#1: start worker process 36
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
2024/07/24 20:45:26 [notice] 1#1: signal 28 (SIGWINCH) received
```

Yavuzlar adında ngnix container’ ı oluşturur ve port 8080,80 arasında tünel oluşturur işlem bittiğinde container’ ı siler.

```
keofla@keofla ~ % docker logs nginx
Error response from daemon: No such container: nginx
keofla@keofla ~ % docker logs SiberVatan
Siber Vatan
keofla@keofla ~ %
```

Çalışan yada çalışmış container’ ların loglarını döndürür.

```
keofla@keofla ~ % docker network ls
NETWORK ID     NAME         DRIVER    SCOPE
949980605e92   SiberVatan   bridge    local
6b6b6e258e0a   bridge       bridge    local
b1eef99e5ca7   host         host      local
25bd0c1a7a16   none         null      local
keofla@keofla ~ %
```

Docker üzerindeki networkleri listeler.

```
keofla@keofla ~ % docker network create Keofla
0abc569a8ec743ab4bed17d007ba82a1281223f48d2fcd97afc79778940261b3
```

Docker da yeni network oluşturur.

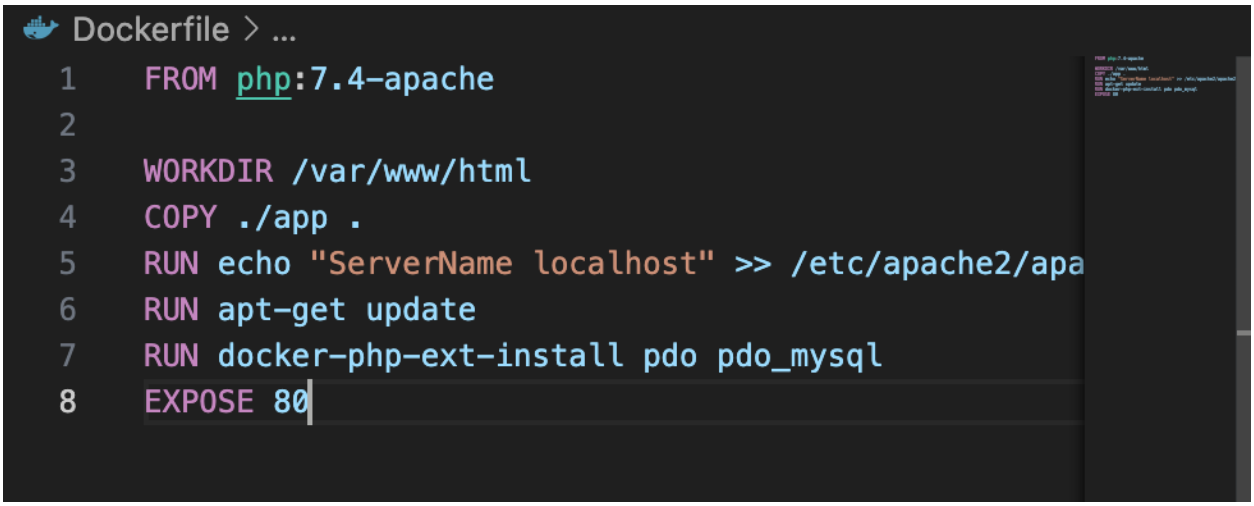
```
keofla@keofla ~ % docker run -d alpine sleep 100
b9ac3355e7371a747e1b633d37259c9e4856122209564c57a262b933054b7ae8
keofla@keofla ~ % ls
Desktop                 Library                 PHPEgitimi              Public
Documents               Movies                  Pictures                Virtual Machines.localized
Downloads               Music                   Postman
keofla@keofla ~ %
```

Container çalışsa bile terminalin kullanılmasını sağlar.

```
keofla@keofla ~ % docker run -e MYSQL_ROOT_PASSWORD=test hello-world

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
```

Image çalıştığında mysql ortamı için veri gönderir.

Dockerfile



FROM php:7.4-apache = Php 7.4 sürümünü image' den çalıştırır.

WORKDIR /var/www/html = Çalışma dosyasını /var/www/html olarak değiştrir.

COPY ./app . = /app dosyasını şuanki çalışan dizine kopyalar.

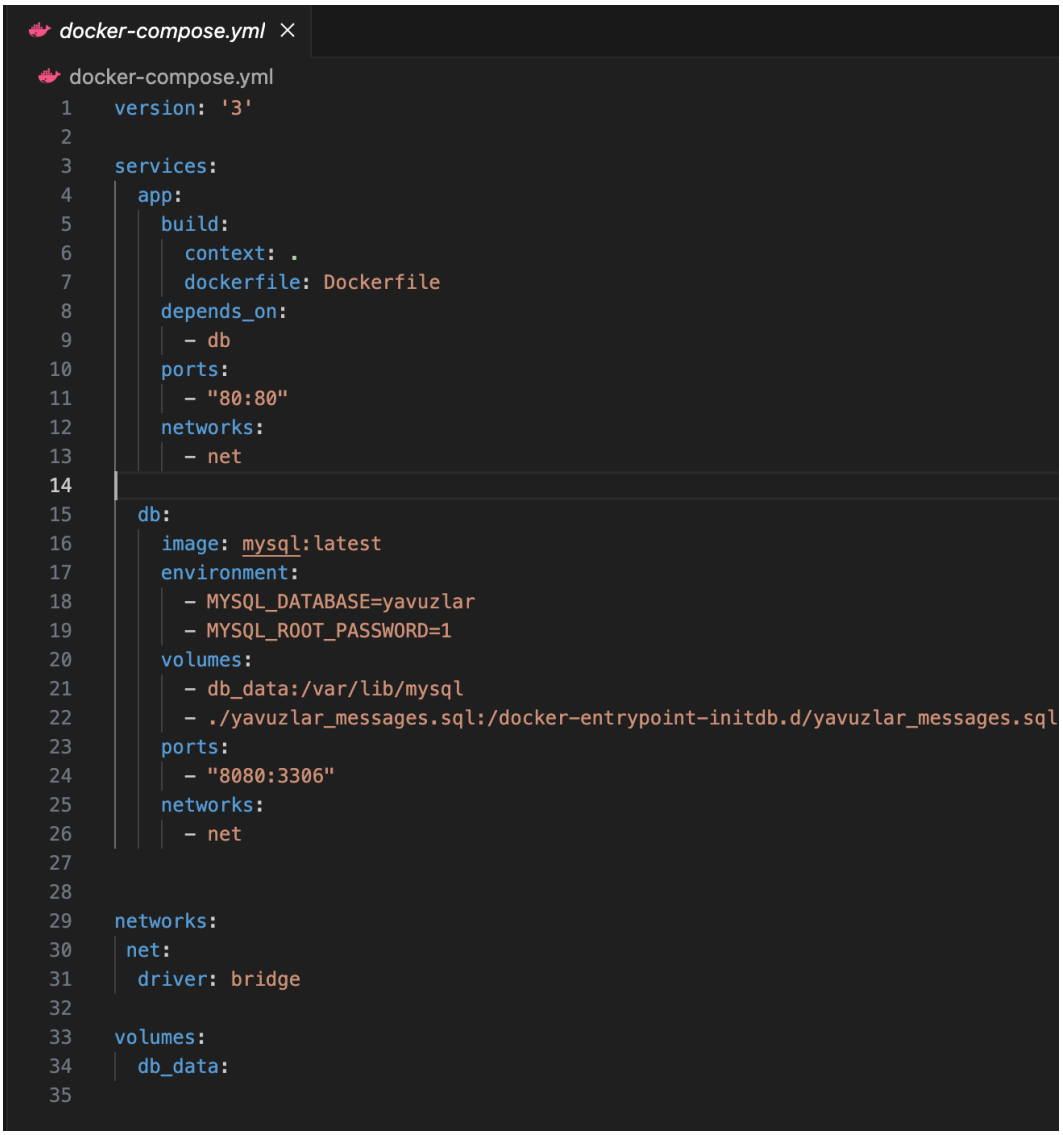
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf = Apache conf dosyasında ServerName' I localhost olarak değiştrir.

RUN apt-get update = işletim sistemi ortamını günceller.

RUN docker-php-ext-install pdo pdo_mysql = docker üzerinde php ve mysql arasındaki bağlantıyı kurmak için gerekli paketleri yükler.

EXPOSE 80 = port 80' I dinler.

Docker-compose.yml



docker-compose versiyon 3 syntax kullanılır.

Services:

Build context’ ini docker-compose dosyasının olduğu dizine ayarlar.

Kullanılacakk Dockerfile dosyası belirlenir.

App db, 80 portu ve net adlı network’ e olan bağımlılığı ayarlanır.

Db:

Mysql’ in son sürümü kullanılır

Mysql için gerekli olan ortam verileri tanımlanır.

Db_data docker’ a bağlanır.

Yavuzlar_messages.sql docker a bağlanır.

Por 8080 ve 3306 arasında tünel oluşturur

Network:

Net adında network oluşturur ve bridge driverını kullanır.

Volumes:

Db_data adında volume oluşturur

Github: