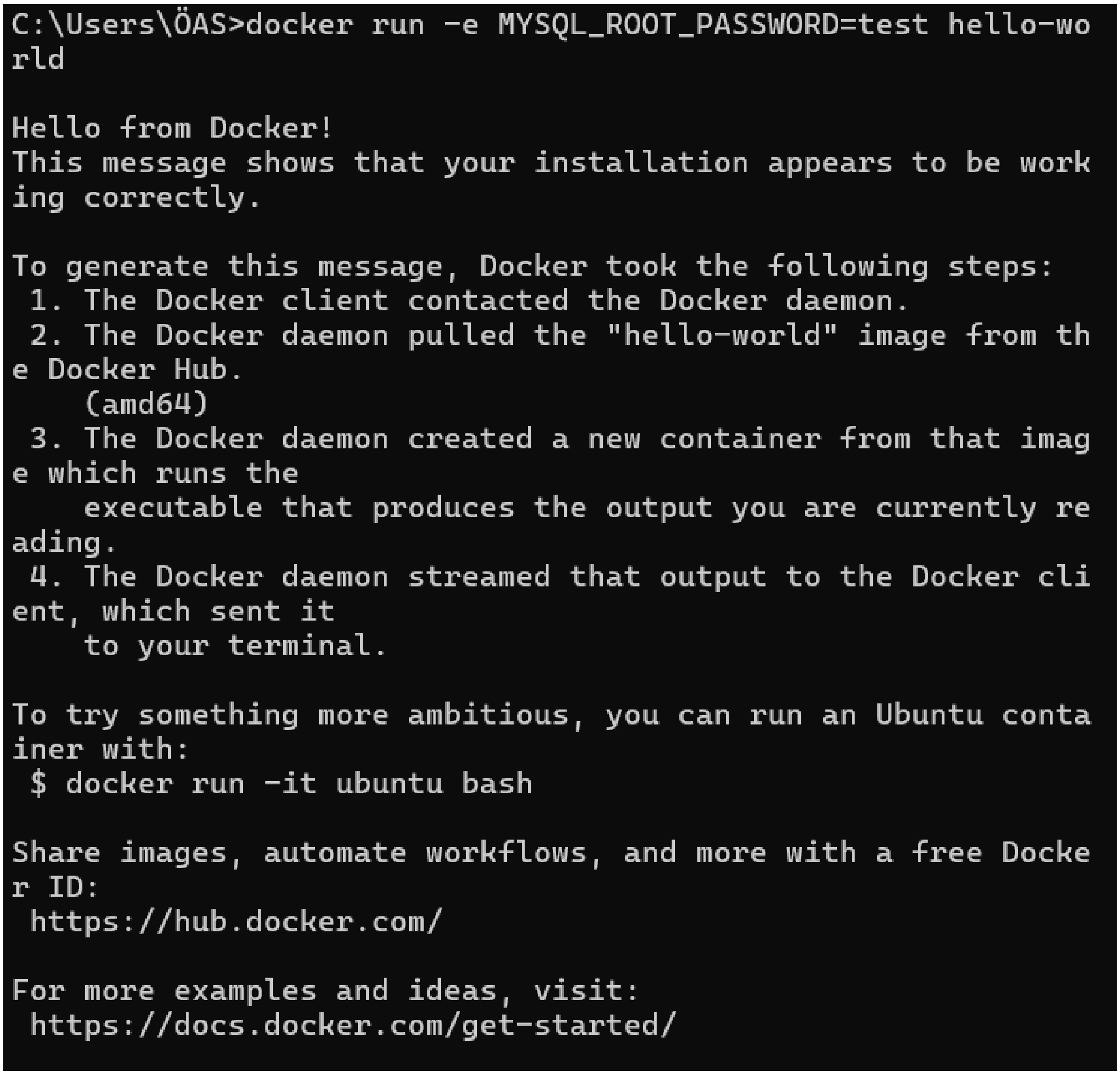
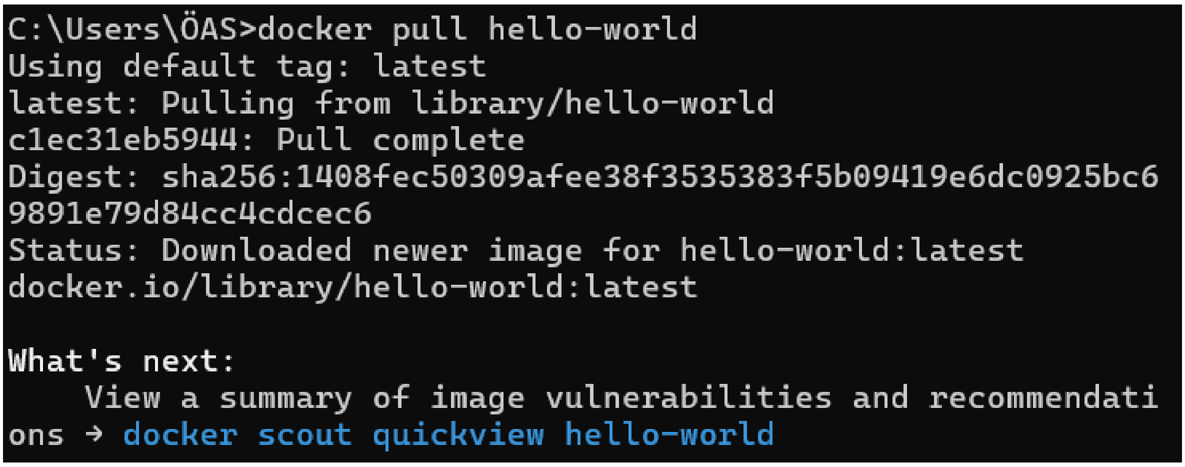


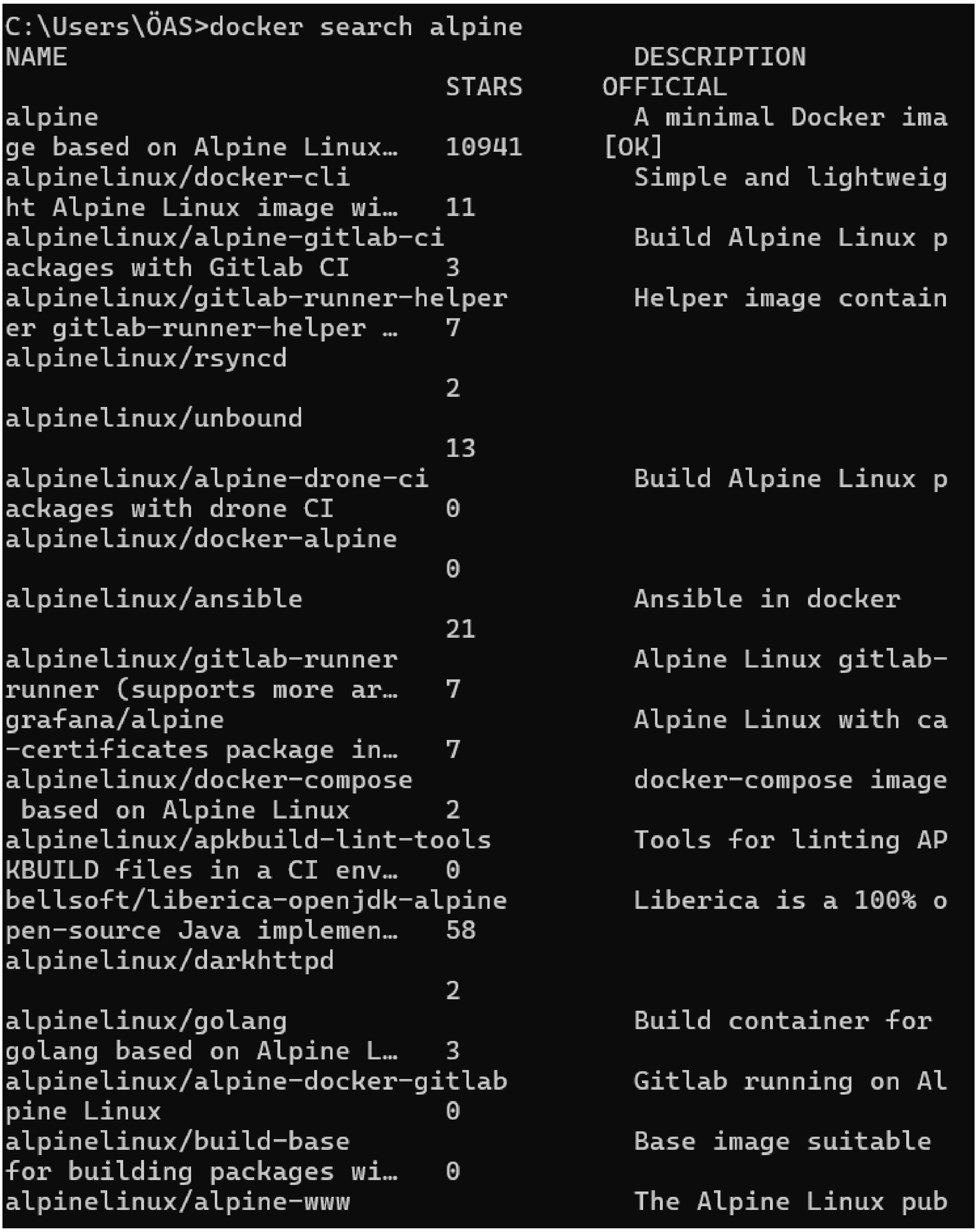
```
C:\Users\ÖAS>docker run -e MYSQL_ROOT_PASSWORD=test hello-wo
rld

Hello from Docker!
This message shows that your installation appears to be work
ing 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 th
e Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that imag
e which runs the
    executable that produces the output you are currently re
ading.
 4. The Docker daemon streamed that output to the Docker cli
ent, which sent it
    to your terminal.

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

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

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

run -e MYSQL_ROOT_PASSWORD=test komutu: Docker ile bir konteyner çalıştırır ve MYSQL_ROOT_PASSWORD ortam değişkenini test olarak ayarlar.



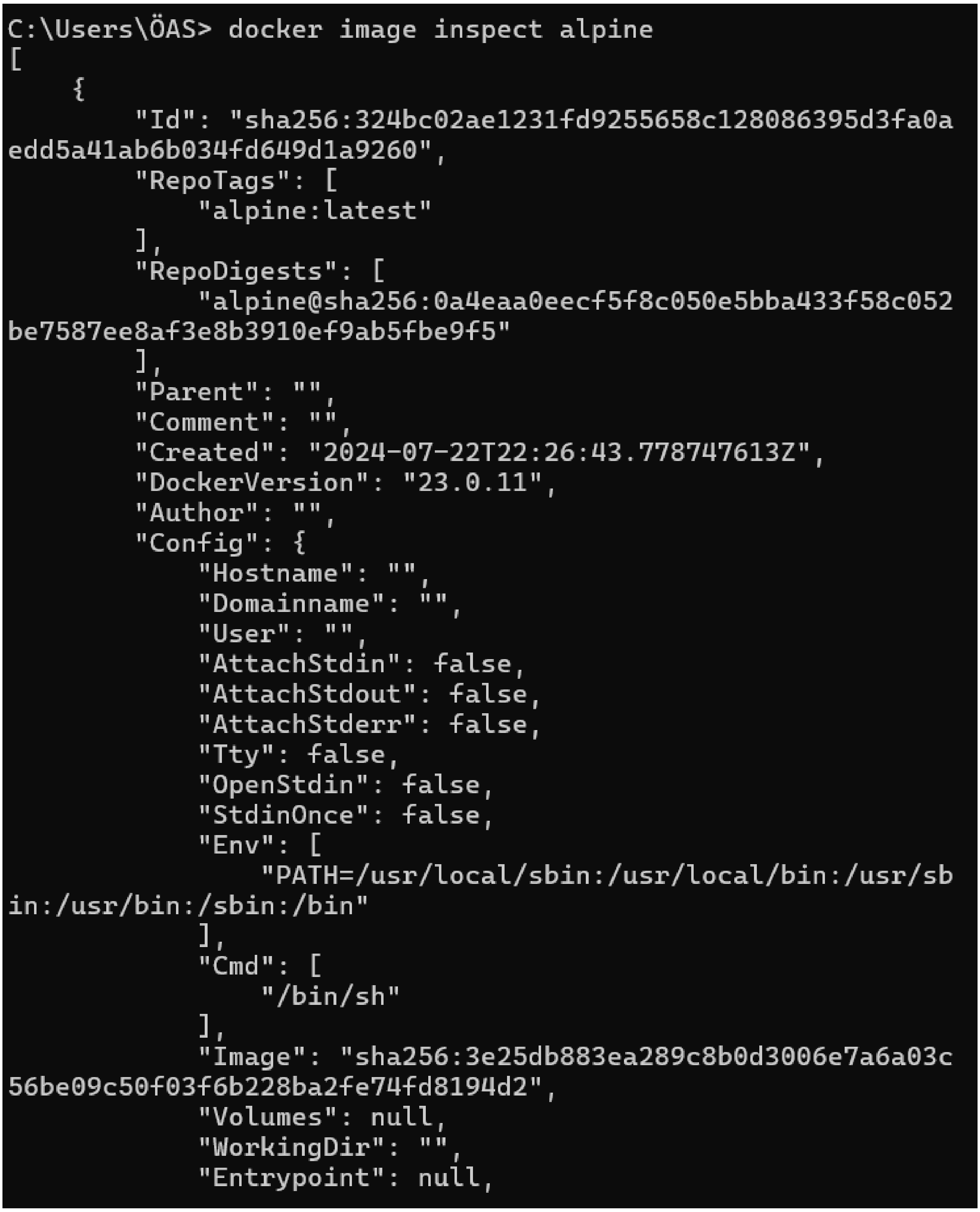
```
C:\Users\ÖAS>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc6
9891e79d84cc4cdcec6
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest

What's next:
    View a summary of image vulnerabilities and recommendati
ons → docker scout quickview hello-world
```

Pull komutu: İmage'ları indirir.



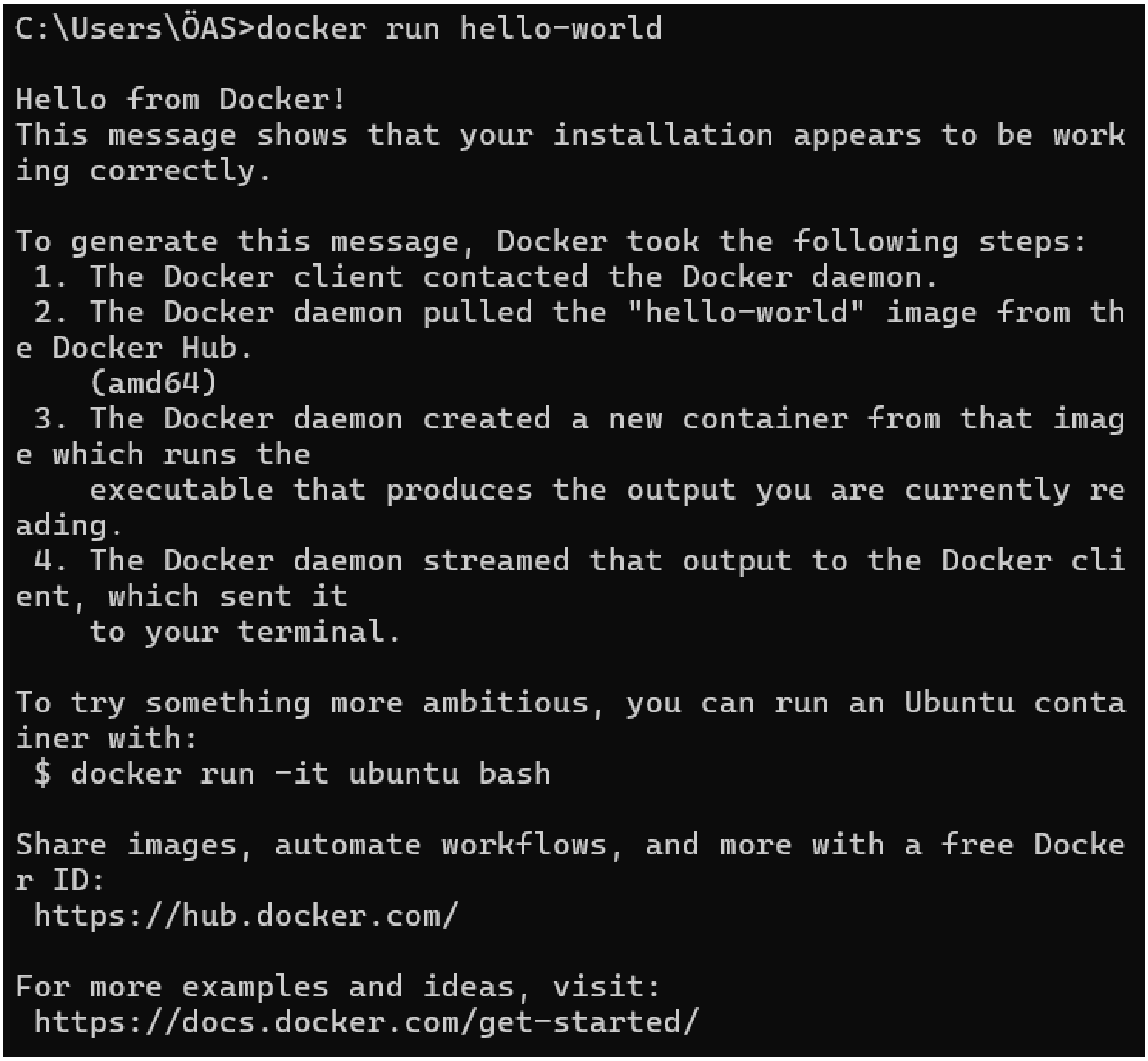
```
C:\Users\ÖAS>docker search alpine
NAME                                        DESCRIPTION
                            STARS     OFFICIAL
alpine                                      A minimal Docker ima
ge based on Alpine Linux…   10941     [OK]
alpinelinux/docker-cli                      Simple and lightweig
ht Alpine Linux image wi…   11
alpinelinux/alpine-gitlab-ci                Build Alpine Linux p
ackages with Gitlab CI      3
alpinelinux/gitlab-runner-helper            Helper image contain
er gitlab-runner-helper …   7
alpinelinux/rsyncd
                            2
alpinelinux/unbound
                            13
alpinelinux/alpine-drone-ci                 Build Alpine Linux p
ackages 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 AP
KBUILD files in a CI env…   0
bellsoft/liberica-openjdk-alpine            Liberica is a 100% o
pen-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 Al
pine Linux                  0
alpinelinux/build-base                      Base image suitable
for building packages wi…   0
alpinelinux/alpine-www                      The Alpine Linux pub
```

Search komutu: Ducker Hub’da public olan İmage’ları arar.



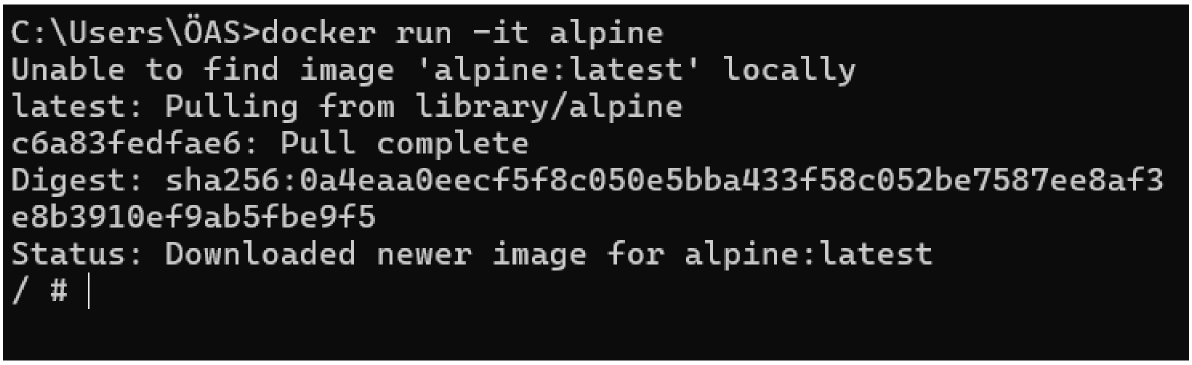
```
C:\Users\ÖAS> docker image inspect alpine
[
    {
        "Id": "sha256:324bc02ae1231fd9255658c128086395d3fa0a
edd5a41ab6b034fd649d1a9260",
        "RepoTags": [
            "alpine:latest"
        ],
        "RepoDigests": [
            "alpine@sha256:0a4eaa0eecf5f8c050e5bba433f58c052
be7587ee8af3e8b3910ef9ab5fbe9f5"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2024-07-22T22:26:43.778747613Z",
        "DockerVersion": "23.0.11",
        "Author": "",
        "Config": {
            "Hostname": "",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sb
in:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh"
            ],
            "Image": "sha256:3e25db883ea289c8b0d3006e7a6a03c
56be09c50f03f6b228ba2fe74fd8194d2",
            "Volumes": null,
            "WorkingDir": "",
            "Entrypoint": null,
```

Image inspect komutu: İmage’ların hakkında bilgi verir.



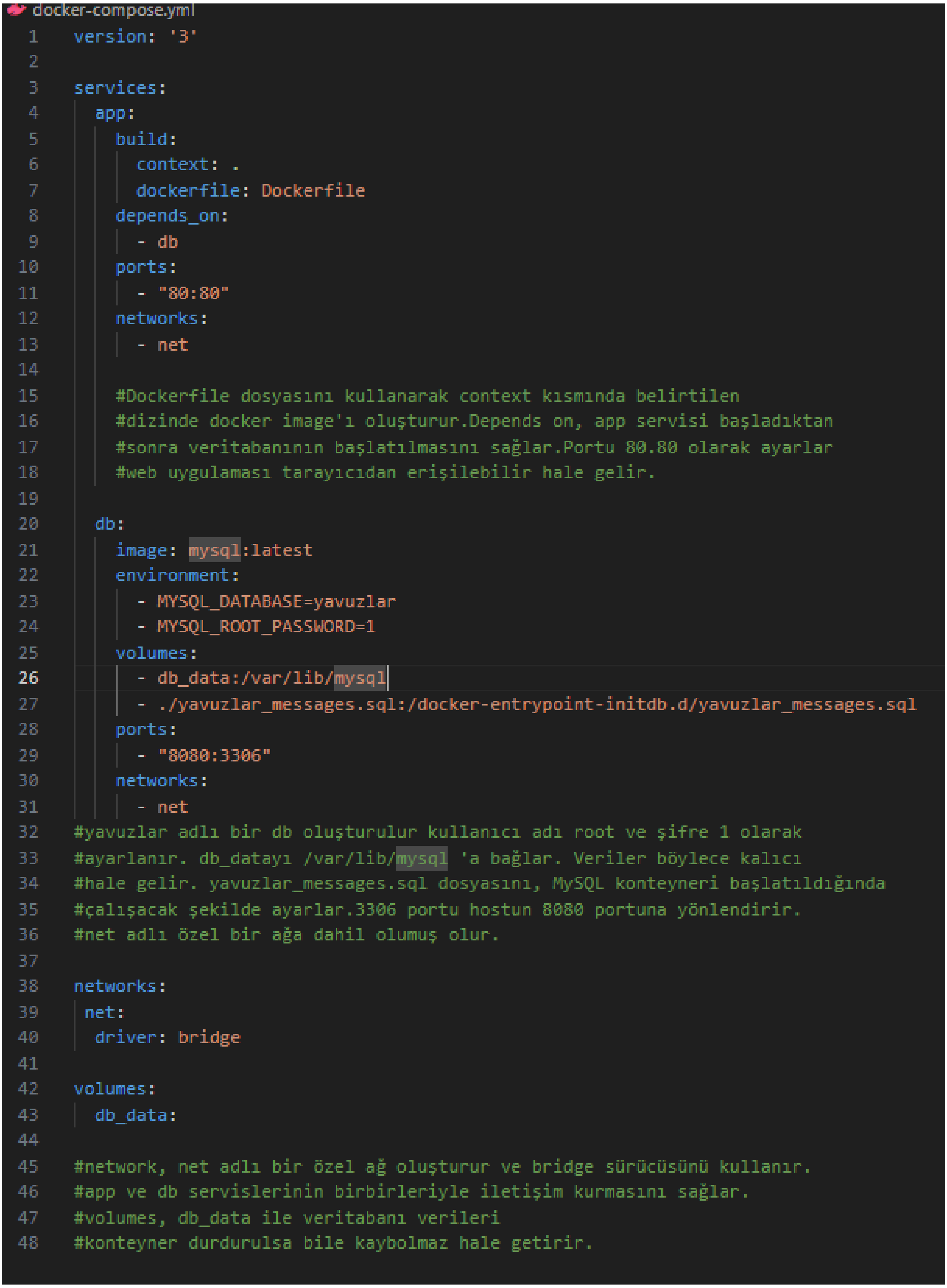
```
C:\Users\ÖAS>docker run hello-world

Hello from Docker!
This message shows that your installation appears to be work
ing 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 th
e Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that imag
e which runs the
    executable that produces the output you are currently re
ading.
 4. The Docker daemon streamed that output to the Docker cli
ent, which sent it
    to your terminal.

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

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

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

Run komutu: İmage’ları çalıştırır.



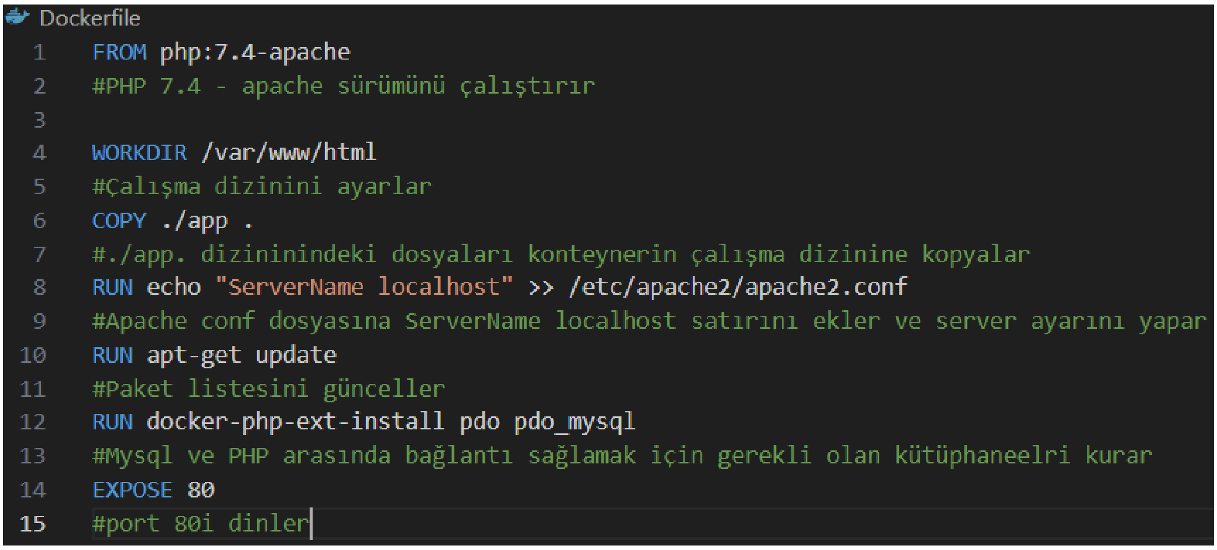
```
C:\Users\ÖAS>docker run -it alpine
Unable to find image 'alpine:latest' locally
latest: Pulling from library/alpine
c6a83fedfae6: Pull complete
Digest: sha256:0a4eaa0eecf5f8c050e5bba433f58c052be7587ee8af3
e8b3910ef9ab5fbe9f5
Status: Downloaded newer image for alpine:latest
/ #
```

Run -it komutu: İmage’ı shelled çalıştırır.



docker-compose.yml

```yaml
version: '3'

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

    #Dockerfile dosyasını kullanarak context kısmında belirtilen
    #dizinde docker image'ı oluşturur.Depends on, app servisi başladıktan
    #sonra veritabanının başlatılmasını sağlar.Portu 80.80 olarak ayarlar
    #web uygulaması tarayıcıdan erişilebilir hale gelir.

  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
#yavuzlar adlı bir db oluşturulur kullanıcı adı root ve şifre 1 olarak
#ayarlanır. db_datayı /var/lib/mysql 'a bağlar. Veriler böylece kalıcı
#hale gelir. yavuzlar_messages.sql dosyasını, MySQL konteyneri başlatıldığında
#çalışacak şekilde ayarlar.3306 portu hostun 8080 portuna yönlendirir.
#net adlı özel bir ağa dahil olumuş olur.

networks:
 net:
  driver: bridge

volumes:
  db_data:

#network, net adlı bir özel ağ oluşturur ve bridge sürücüsünü kullanır.
#app ve db servislerinin birbirleriyle iletişim kurmasını sağlar.
#volumes, db_data ile veritabanı verileri
#konteyner durdurulsa bile kaybolmaz hale getirir.
```



```dockerfile
FROM php:7.4-apache
#PHP 7.4 - apache sürümünü çalıştırır

WORKDIR /var/www/html
#Çalışma dizinini ayarlar
COPY ./app .
#./app. dizinindeki dosyaları konteynerin çalışma dizinine kopyalar
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
#Apache conf dosyasına ServerName localhost satırını ekler ve server ayarını yapar
RUN apt-get update
#Paket listesini günceller
RUN docker-php-ext-install pdo pdo_mysql
#Mysql ve PHP arasında bağlantı sağlamak için gerekli olan kütüphaneelri kurar
EXPOSE 80
#port 80i dinler
```