

```
C:\Users\Eudor>docker search alpine
```

NAME	DESCRIPTION
alpine	A minimal Docker image
alpinelinux/docker-cli	Simple and lightweight
alpinelinux/alpine-gitlab-ci	Build Alpine Linux
alpinelinux/gitlab-runner-helper	Helper image conta
alpinelinux/rsyncd	
alpinelinux/unbound	
alpinelinux/alpine-drone-ci	Build Alpine Linux
alpinelinux/docker-alpine	
alpinelinux/ansible	Ansible in docker
alpinelinux/gitlab-runner	Alpine Linux gitla
grafana/alpine	Alpine Linux with
alpinelinux/docker-compose	docker-compose ima
alpinelinux/apkbuild-lint-tools	Tools for linting
bellsoft/liberica-openjdk-alpine	Liberica is a 100%
alpinelinux/darkhttpd	
alpinelinux/golang	Build container fo
alpinelinux/alpine-docker-gitlab	Gitlab running on
alpinelinux/build-base	Base image suitabl
alpinelinux/alpine-www	The Alpine Linux p
bellsoft/liberica-openjre-alpine	Liberica is a 100%
alpinelinux/docker-abuild	Dockerised abuild
bellsoft/liberica-openjdk-alpine-musl	Liberica is a 100%
balenalib/amd64-alpine-node	This image is part
alpinelinux/mqtt-exec	
alpinelinux/netbox	dockerized netbox

Docker search <image> = bu komut docker hubtan image arar.

```
C:\Users\Eudor>docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
Digest: sha256:1408fec50309afee38f3535383f5b09419e6dc0925bc
Status: Image is up to date for hello-world:latest
docker.io/library/hello-world:latest

What's Next?
  View a summary of image vulnerabilities and recommendations

C:\Users\Eudor>
```

Docker pull <image> = docker hub'tan image indirir.

```
C:\Users\Eudor>docker network ls
NETWORK ID          NAME                DRIVER              SCOPE
63862e63d120       bridge             bridge             local
904a7c20cc5f       host               host               local
8b1e62bb0110       none              null               local
```

Docker network ls = docker'daki ağları listeler.

```
C:\Users\Eudor>docker images
REPOSITORY          TAG                IMAGE ID            CREATED
alpine              latest            324bc02ae123       45 hours ago
ubuntu              latest            35a88802559d       6 weeks ago
diwa                latest            c45db2be76dc       2 months ago
yavuzlar/vulnlab    latest            92fac06d8285       3 months ago
hello-world         latest            d2c94e258dcb       14 months ago
```

Docker images / docker image ls = docker hubtaki indirilen image'leri listeler.

```
C:\Users\Eudor>docker image rm d2c94e258dcb --force
Untagged: hello-world:latest
Untagged: hello-world@sha256:1408fec50309afee38f3535383f5b0
Deleted: sha256:d2c94e258dcb3c5ac2798d32e1249e42ef01cba4841
```

Docker image rm <image id> = belirttiğimiz image'i siler.

```
C:\Users\Eudor>docker inspect 92fac06d8285
[
  {
    "Id": "sha256:92fac06d828556b14e0f357e351fe1064f80",
    "RepoTags": [
      "yavuzlar/vulnlab:latest"
    ],
    "RepoDigests": [
      "yavuzlar/vulnlab@sha256:6b8ba47c01b42346b0820"
    ]
  }
]
```

Docker inspect <image id> = image hakkında detaylı bilgi sağlar.

```
C:\Users\Eudor>docker run --name yavuzlar alpine

C:\Users\Eudor>docker ps -a
CONTAINER ID   IMAGE      COMMAND
NAMES
08d52bed50d5   alpine    "/bin/sh"
yavuzlar
```

Docker run --name <herhangi bir isim> <image> = image'e belirli bir isim veririz.

```
C:\Users\Eudor>docker run -it alpine
/ # ls
bin      dev      etc      home     lib      media    mnt      opt      pro
```

Docker run -it <image> = konteynırı interaktif modda başlatır ve komut çalıştırmamızı sağlar.

```
C:\Users\Eudor>docker run --rm alpine ls -l
total 56
drwxr-xr-x    2 root    root      4096 Jul 22 14:34 b
drwxr-xr-x    5 root    root      340 Jul 24 20:43 d
drwxr-xr-x    1 root    root     4096 Jul 24 20:43 e
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 h
drwxr-xr-x    6 root    root     4096 Jul 22 14:34 l
drwxr-xr-x    5 root    root     4096 Jul 22 14:34 m
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 m
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 o
dr-xr-xr-x   278 root    root      0 Jul 24 20:43 p
drwx-----    2 root    root     4096 Jul 22 14:34 r
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 r
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 s
drwxr-xr-x    2 root    root     4096 Jul 22 14:34 s
dr-xr-xr-x   11 root    root      0 Jul 24 20:43 s
drwxrwxrwt    2 root    root     4096 Jul 22 14:34 t
drwxr-xr-x    7 root    root     4096 Jul 22 14:34 u
drwxr-xr-x   12 root    root     4096 Jul 22 14:34 v

C:\Users\Eudor>
```

Docker run --rm <image> ls -l = --rm komutu konteynır işını bitirdikten sonra kendini silmesini sağlar.

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

Docker ps = çalışan konteynırları gösterir.(şuan bende çalışmadığı için göstermiyor.)

```

C:\Users\Eudor>docker ps -a
CONTAINER ID   IMAGE                                COMMAND
TS           NAMES
a6a42e432273   alpine                              "/bin/sh"
              agitated_ptolemy
08d52bed50d5   alpine                              "/bin/sh"
              yavuzlar
6753d78a9695   yavuzlar/vulnlab:latest            "/usr/sbin/run.sh"
              inspiring_allen
0c2dfdca9f6b   alpine                              "/bin/sh"
              admiring_kilby
5471028cf124   alpine                              "echo selamlar!"
              festive_tharp
0441bb91e012   d2c94e258dcb                       "/hello"
              docker101
b52caf0b5a46   ubuntu                             "bash"
              exciting_snyder
3e7385ce8fc5   d2c94e258dcb                       "/hello"
              vigilant_pike
31decd19c2e4   diwa:latest                         "docker-php-entryp
              diwa
9454cf708dd4   yavuzlar/vulnlab:latest            "/usr/sbin/run.sh"
              crazy_austin

```

Docker ps -a = tüm konteynırları listeler.

DockerFile

```

Dockerfile
1 FROM php:7.4-apache

```

FROM = konteynırın hangi image'ı kullanacağını belirtir.

```
WORKDIR /var/www/html
```

WORKDIR = çalışma dizinini belirtir.

```
COPY ./app .
```

COPY = konteynır içine dosya kopyalamaya yarar.

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

RUN = image üzerinde komutları çalıştırmaya yarar.

```
EXPOSE 80
```

EXPOSE = Konteynırın hangi portlarının dışarıya açıldığını belirler.

DOCKER COMPOSE

```
version: '3'
```

Version = Docker Compose'ın versiyonunu belirtir.

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

Services = Docker Compose'daki tanımlanan servislerin başladığı bölümü belirtir.

Bu kodda hangi servislerin çalışacağını bu komutun altına yazmış.

Build = DockerFile'in yerini belirtir.

Bu kodda DockerFile'in yerini belirtmiş.

Depends_on = neye bağımlı olduğunu belirtildiği yerdir.

Bu kodda db'ye bağımlı olduğunu söylemiş.

Ports = Konteynerin hangi portlarının host makinadaki portlara bağlanacağını belirtir.

Bu kodda 80 portunu bizim ana makinemizin 80 portumuza bağlamış.

Networks = Servisin hangi Docker ağlarında çalışacağını belirtir.

Bu kodda net adında bir ağ tanımlanmış.

```
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 = database ayarları

Image = hangi image olduğunu belirtmek için kullanılır.

Bu kodda mysql:latest adında bir image'ı kullanmışız.

Environment = Konteynırın çalıştırıldığı ortam değişikliklerini tanımlamak için kullanılır.

Bu kodda yavuzlar adında bir mysql database ismi belirtilmiş ve şifresini 1 olarak belirlemişiz.

Volumes = konteynırlar arasında veya host ve konteynır arasındaki veri paylaşımını kolaylaştırmak için kullanılır.

Bu kodda db_data kısmında database'in yolunu vermişiz ve yavuzlar_messages.sql dosyasını /docker-entrypoint-initdb.d/yavuzlar_messages.sql dizinine atmış.

Ports = bu kısımda 3306 portuna ana makinemizin 8080 portundan erişmemizi sağlar.