## **DOCKER KULLANIMI**

**Docker login:** Docker çalışması için giriş yapıyoruz.

```
C:\Users\Hp>docker login
Authenticating with existing credentials...
Login did not succeed, error: error during connect: Post "http://%2F%2F.%2Fpipe%2FdockerDesktopLinuxEngine/v1.45/auth":
open //./pipe/dockerDesktopLinuxEngine: Sistem belirtilen dosyayı bulamıyor.
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username (muratozylmz):
Password:
Login Succeeded
C:\Users\Hp>_
```

Docker versiyon: Elimizdeki docker'ın sürümünü ve özelliklerini öğrenmek için kullanırız.

```
C:\Users\Hp>docker version
Client:

Version: 26.1.4
API version: 1.45
Go version: gol.21.11
Git commit: 5650f9b
Built: Wed Jun 5 11:29:54 2024
OS/Arch: windows/amd64
Context: desktop-linux

Server: Docker Desktop 4.31.1 (153621)
Engine:
Version: 26.1.4
API version: 1.45 (minimum version 1.24)
Go version: gol.21.11
Git commit: de5c9cf
Built: Wed Jun 5 11:29:22 2024
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.6.33
GitCommit: d258213f83a351ca8f528a95fbd145f5654e957
runc:
Version: 1.1.12
GitCommit: v1.1.12-0-g51d5e94
docker-init:
Version: 0.19.0
GitCommit: de40ad0

C:\Users\Hp>_
```

Docker pull alpine: Alpine isimli paketi indirmek için kullanıyoruz.

```
C:\Users\Hp>docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
Digest: sha256:0a4eaa0eecf5f8c050e5bba433f58c052be7587ee8af3e8b3910ef9ab5fbe9f5
Status: Image is up to date for alpine:latest
docker.io/library/alpine:latest
What's next:

View a summary of image vulnerabilities and recommendations → docker scout quickview alpine
```

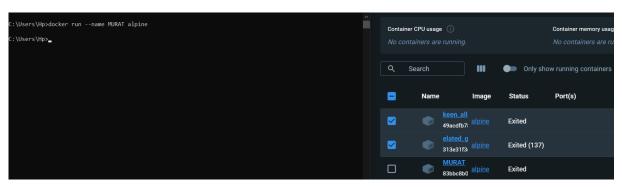
Docker images: Elimizde bulunan paketleri listeler.

```
C:\Users\Hp>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
```

Docker run –rm alpine: Yüklü paketi silmek için kullanırız.

```
:\Users\Hp>docker run --rm alpine
:\Users\Hp>
```

**Docker run –name MURAT alpine:** MURAT isimli bir konteynır oluşturmak için kullanırız.



Docker container ps: Çalışan paketleri listeler.

Docker container ps -a: Çalışmayan paketleri listeler.



**Docker container exec –it a6eb45 sh:** a6eb45 adlı id'ye sahip olan ve aktif çalışan konteynıra Shell atarak erişim sağlıyoruz. Daha sonra ls çekerek dosyaları ekliyoruz.

```
C:\Users\Hp>docker container exec -it a6eb45 sh
/ # ls
bin etc lib mnt proc run srv tmp var
dev home media opt root sbin sys usr
/ #
```

Hocam aklımda olanlar bunlardı. Gün çok uzun ve yorucu olunca...

## DOCKERFILE İÇERİĞİ

```
Dockerfile > ...

#MURAT ÖZYILMAZ

FROM php:7.4-apache

WORKDIR /var/www/html

COPY ./app .

RUN apt-get update -y

RUN docker-php-ext-install pdo pdo_mysql

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

EXPOSE 80

EXPOSE 80
```

**FROM php:7.4-apache:** *Kullandığımız imaj dosyasının ne olduğunu, neler kullanıldığını yazıyoruz.* 

**WORKDIR /var/www/html:** Çalışacağımız dosyanın yolunu yazarız. Daha sonradan yazdığımız komutlar burada çalıştırılır.

**COPY ./app .:** Ana makinemizde bulunan "./app" klasörünü docker'daki konteynırın içine kopyalar.

RUN apt-get update -y: Kodlar ubuntuda çalışacağı için sürümlerden dolayı sorun yaşamamak için makineyi update ederiz yani güncelleriz.(-y de çıkacak tüm sorulara evet denilmesi için yazılır.)

RUN docker-php-ext-install pdo pdo\_mysql: Mysql ile veri tabanı bağlantısı gerçekleştirilmesi için zorunlu olarak yazılan komut.

RUN echo"ServerName localhost" >> /etc/apache/apache2.conf: Apache yapılandırma dosyasına "ServerName localhost" satırını ekler.

**EXPOSE:** Dinlenecek veya gittiğimiz sitede görüntülemek için kullanılacak portu buraya ekleriz.

## DOCKER COMPOSE İÇERİĞİ

```
build:
      dockerfile: Dockerfile
    networks:
    environment:
    - MYSQL_DATABASE=yavuzlar
      - MYSQL_ROOT_PASSWORD=1
      - db data:/var/lib/mysql
      - ./yavuzlar_messages.sql:/docker-entrypoint-initdb.d/yavuzlar_messages.sql
    ports:
     - "8080:3306"
     networks:
     - net
networks:
 driver: bridge
```

Version: Versiyonun kaç olduğunu belirtiriz.

**Services:** Servisleri açıklamaya başlarız.

**App:** Servisi tanımlarız.

Context: Dockerfile dosyasının bulunduğu dizini yazarız. (. Demek bu dizinde olduğunu

belirtir.)

Dockerfile: Dockerfile: Kullanılan Dockerfile dosya ismini belirtiriz.

Depends\_on: Veritabanı türü, hangi veritabanı kullanılacağı.

-db: Bağımlı olduğunu belirtiriz.

Ports: Hangi portla erişileceği yazılır.

80:80= Ana makinedeki 80 portunu konteynırdaki 80 portuna yönlendirme yaparız.

image: mysql:latest= Kullanılan imaj mysql' in en son sürümü olacağını belirtiriz.

**Environment:** Değişkenleri tanımlarız.

MYSQL\_DATABASE=yavuzlar: Veritabanı ismi yavuzlar olacak.

MYSQL\_ROOT\_PASSWORD=1: root şifresi 1 olacak.

Volumes: Hangi dosyaların bağlanacağını belirtiriz.

**Db\_data:/var/lib/mysql:** Bağlanan dosyamızı dosya yolu ile birlikte yazarız.

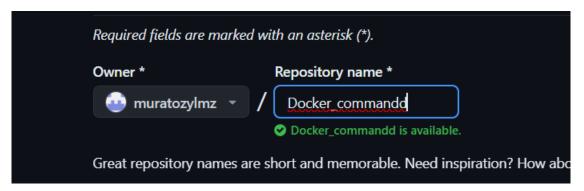
**8080:3306=** Ana makinemizdeki 8080 portu docker'daki konteynırda 3306 portuna yönlendirilir.

## GİTHUB'A PROJE YÜKLEME

İlk adım olarak repo oluşturuyoruz.



Daha sonra isim veriyoruz.



Bilgisayarımızda proje oluşturuyoruz.

```
MURAT ÖZYILMAZ@Murat MINGW64 ~ (main)

$ git init
Reinitialized existing Git repository in C:/Users/Hp/.git/
```

Burada dosyamızı github'a yüklüyoruz.

```
MURAT ÖZYILMAZ@Murat MINGW64 ~/Desktop (main)
$ git add Docker_Github.pdf
```

Burada da yüklenen dosyamıza yorum satırı yazıyoruz.

```
MURAT ÖZYILMAZ@Murat MINGW64 ~/Desktop (main)
$ git commit -m "Docker Ödevi"
[main a01b4b9] Docker Ödevi
1 file changed, 0 insertions(+), 0 deletions(-)
create mode 100644 Desktop/Docker_Github.pdf
```

Remote: Uzak repoyu linkte bulunan(kendi github linkim) gönderiyoruz.

Branch -m main: Mevcut branch'i main branchine bağlar.

git push: Dosyayı github'a yüklememizi sağlar.

```
MURAT ÖZYILMAZ@Murat MINGW64 ~/Desktop (main)

$ git remote add origin https://github.com/muratozylmz/Docker_command.git

MURAT ÖZYILMAZ@Murat MINGW64 ~/Desktop (main)

$ git branch -m main

MURAT ÖZYILMAZ@Murat MINGW64 ~/Desktop (main)

$ git push -u origin main
Enumerating objects: 12, done.
Counting objects: 100% (12/12), done.
Delta compression using up to 12 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (12/12), 1.14 MiB | 535.00 KiB/s, done.
Total 12 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), done.
To https://github.com/muratozylmz/Docker_command.git

* [new branch] main -> main
branch 'main' set up to track 'origin/main'.
```