

Merhaba,

[A'dan Z'ye Docker](#) eğitimimdeki öğrencilerimin yararlanabilmesi adına Docker Cheat Sheet yani Türkçeleştirsek "Docker komutlarının nasıl kullanıldığının bir listesi" hazırladım. Yararlı bir derleme olduğunu düşünüyorum. Daha geniş bir kitleye ulaşabilmesi için de burada yayınlıyorum. Kolay gelsin.

## ***Docker Cheat Sheet (Türkçe)***

---

- ***Container***

### **Container çalıştırma:**

```
docker container run image:tag
```

Örnek: `docker container run nginx:latest`

---

### **Detach modda container çalıştırma (-d):**

```
docker container run -d image:tag
```

Örnek: `docker container run -d nginx:latest`

---

### **Varsayılan uygulama yerine başka uygulama ile container başlatma:**

```
docker container run image:tag uygulama
```

Örnek: `docker container run ubuntu:latest ping 127.0.0.1`

---

### **Container'a bir isim vererek çalıştırma**

```
docker container run --name isim image:tag
```

Örnek: `docker container run --name container1 -d nginx:latest`

---

### **Çalışan bir container içerisinde başka bir uygulama çalıştırma:**

```
docker container exec container_id|ya da|container_ismi uygulama
```

Örnek: `docker container exec 12a793b3fec0 ping 127.0.0.1`

---

## Çalışan bir container'a shell bağlantısı oluşturma:

```
docker container exec -it container_id|ya da|container_ismi sh
```

Örnek: `docker container exec -it 12a793b3fec0 sh`

---

## Container'ı detach modda ve shell bağlantısı ile oluşturma (dit):

```
docker container run -dit image:tag sh
```

Örnek: `docker container run -dit nginx:latest sh`

---

## Detach modda ve shell bağlantısı ile oluşturulmuş container'a bağlanma:

```
docker attach container_id|ya da|container_ismi
```

Örnek: `docker attach 12a793b3fec0`

---

## Container durdurma:

```
docker container stop container_id|ya da|container_ismi
```

Örnek: `docker container stop 12a793b3fec0`

---

## Container silme:

```
docker container rm container_id|ya da|container_ismi
```

Örnek: `docker container rm 12a793b3fec0`

---

## Çalışan containerı silme (-f):

```
docker container rm -f container_id|ya da|container_ismi
```

Örnek: `docker container rm -f 12a793b3fec0`

---

## Container kapatıldığı zaman aynı zamanda silinmesi (-rm):

```
docker container run -rm image:tag
```

Örnek: `docker container run -rm nginx:latest` (-rm ile container kapandığı zaman otomatik olarak silinmesini söylüyoruz)

---

### Container ile ilgili detayları inceleme:

```
docker container inspect container_id|ya da|container_ismi
```

Örnek: `docker container inspect 12a793b3fec0`

---

### Sistemdeki tüm containerları (çalışan ve durdurulmuş) silme

```
docker container rm -f $(docker ps -aq)
```

---

### Sistemdeki çalışan containerları listeleme:

```
docker container ls ya da
```

```
docker container ps
```

---

### Sistemdeki tüm containerları listeleme:

```
docker container ls -a ya da
```

```
docker container ps -a
```

---

### Çalışan container'daki processleri listeleme:

```
docker top container_id|ya da|container_ismi
```

Örnek: `docker top 12a793b3fec0`

---

### Çalışan container'ın Cpu, Ram, I/O kullanımını görme:

```
docker stats container_id|ya da|container_ismi
```

Örnek: `docker stats 12a793b3fec0`

---

### Container'ın memory kullanımını sınırlama (--memory, --memory-swap):

```
docker container run --memory=rakam(b,k,m,g) --memory-swap=rakam(b,k,m,g) image:tag
```

Örnek: `docker container run --memory=1g --memory-swap=2g nginx:latest` (memory-swap ile swap alanı da tanımlayabiliriz.b=byte k=kilobyte m=megabyte g=gigabyte)

---

### Container'ın cpu kullanımını sınırlama (--cpus, --cpuset-cpus):

```
docker container run --cpus="core_adeti" image:tag
```

Örnek: `docker container run --cpus="3" nginx:latest` (sistemden kaç core'a erişebileceğini belirledik)

```
docker container run --cpuset-cpus="core_numarası" image:tag
```

Örnek: `docker container run --cpuset-cpus="0,4" nginx:latest` (sistemdeki hangi corelara erişebileceğini belirledik)

---

### Container'a enviroment variable tanımlama:

```
docker container run --env enviroment_variable=değeri image:tag
```

Örnek: `docker container run --env VAR1=deneme1 --env VAR2=deneme2 nginx:latest`

---

### Containerdan hosta ya da tam tersi dosya kopyalama:

```
docker cp container_id|ya da|container_ismi:path host_path
```

Örnek: `docker cp 12a793b3fec0:/usr/src/uygulama/ .`

---

---

---

- *Image*

### Docker CLI aracılığıyla registry'de oturum açma:

```
docker login registry_url
```

Örnek: `docker login localhost:8080`

---

### Sisteme bir imaj çekme:

```
docker image pull image:tag
```

Örnek: `docker image pull nginx:latest`

---

### Docker hub'a (ya da başka bir repository) image gönderme:

```
docker image push repository/image:tag
```

Örnek: `docker image push ozgurozturknet/adanyedocker:latest`

---

### Mevcut bir imaja yeni tag ekleme

```
docker image tag image:tag yeniimage:tag
```

Örnek: `docker image tag nginx:latest ozgurozturknet\nginx:v1`

---

### Image ile ilgili detayları inceleme:

```
docker image inspect image:tag
```

Örnek: `docker image inspect nginx:latest`

---

### Image layerlarını listeleme:

```
docker image history image:tag
```

Örnek: `docker image history nginx:latest`

---

### Dockerfile kullanarak yeni bir imaj yaratma:

```
docker image build -t image:tag .
```

Örnek: `docker image build -t ozgurozturknet\merhaba-dunya:latest .` (Dockerfile dosyası komutun çalıştırıldığı folder'da bulunmalı)

---

### Image oluştururken build arg kullanma:

```
docker image build --build-arg arg=deger -t image:tag .
```

Örnek: `docker image build --build-arg VERSION=3.7.1 -t nginx:latest .`

---

## Sistemdeki tüm imajeleri listeleme:

```
docker image ls
```

---

## Sistemden bir imajı silme:

```
docker image rm image:tag
```

Örnek: `docker image rm nginx:latest`

---

## Containerdan image yaratma:

```
docker commit container_id|ya da|container_ismi image:tag
```

Örnek: `docker commit 12a793b3fec0 ozgurozturknet/img:latest`

---

## Image'i bir dosyaya kaybetmek ve kaydedilmiş bir dosyadan image oluşturmak:

```
docker save image:tag -o dosyaadi.tar
```

Örnek: `docker save ozgurozturknet/img:latest -o image.tar`

```
docker load -i dosyaadi.tar
```

Örnek: `docker load -i imagecon1.tar`

---

---

---

- **Volume**

## Volume oluşturma:

```
docker volume create volume_ismi
```

Örnek: `docker volume create ilkvolume`

---

## Volume ile ilgili detayları inceleme:

```
docker volume inspect volume_id|ya da|volume_ismi
```

Örnek: `docker volume inspect ilkvolume`

---

### Sistemdeki tüm volumeleri listeleme:

```
docker volume ls
```

---

### Volume'u container'a bağlama (-v):

```
docker container run -v volume_ismi:container_icindeki_path image:tag
```

Örnek: `docker container run -v ilkvolume:/var/www/html image:tag`

---

### Volume'u container'a sadece okunur şekilde bağlama (:ro):

```
docker container run -v volume_ismi:container_icindeki_path:ro image:tag
```

Örnek: `docker container run -v ilkvolume:/var/www/html:ro image:tag`

---

### Host üstündeki bir klasör ya da dosyayı bind mount olarak bağlama:

```
docker container run -v host_klasör_path:container_icindeki_path image:tag
```

```
docker container run -v c:\websitesi:/usr/share/nginx/html nginx:latest
```

---

### Volume silme:

```
docker volume rm volume_ismi
```

Örnek: `docker volume rm ilkvolume`

---

---

---

- **Network**

### Kullanıcı tanımlı bridge network oluşturma (bridge):

```
docker network create --driver=bridge network_ismi
```

Örnek: `docker network create --driver=bridge kopru`

---

## Kullanıcı tanımlı bridge network oluşturma (ip bilgilerini belirleyerek):

```
docker network create --driver=bridge --subnet=cidr --ip-range=cidr --gateway=ip_adresi  
network_ismi
```

Örnek: `docker network create --driver=bridge --subnet=10.10.0.0/16 --ip-range=10.10.10.0/24 --gateway=10.10.10.10 kopru`

---

## Sistemdeki tüm volumeleri listeleme:

```
docker network ls
```

---

## Volume ile ilgili detayları inceleme:

```
docker network inspect network_ismi
```

Örnek: `docker network inspect kopru`

---

## Container'ı varsayılan dışında bir network'e bağlayarak çalıştırma:

```
docker container run --network network_ismi image:tag
```

Örnek: `docker container run --network kopru nginx:latest`

---

## Çalışan bir container'ı başka bir network'e bağlama:

```
docker network connect network_ismi container_id|ya da|container_ismi
```

Örnek: `docker network connect kopru 12a793b3fec0`

---

## Çalışan bir container'ın bağlı olduğu networkle bağlantısını kesme:

```
docker network disconnect network_ismi container_id|ya da|container_ismi
```

Örnek: `docker network disconnect kopru 12a793b3fec0`

---

## Port publish ederek bir container çalıştırma (-p):

```
docker container run -p host_portu:container_portu/tcp_yada_udp image:tag
```



Örnek: `docker container run -p 8080:80 -p 53:53/udp nginx:latest`

---

---

---

- **Logging**

### Container tarafından oluşturulan logları görmek:

```
docker logs container_id|ya da|container_ismi
```

Örnek: `docker logs 12a793b3fec0`

---

### Container tarafından oluşturulan logları uzun formatta detaylı görmek:

```
docker logs --details container_id|ya da|container_ismi
```

Örnek: `docker logs --details 12a793b3fec0`

---

### Container tarafından oluşturulan logları belirli tarih aralığında görmek:

```
docker logs --since tarih_saat --until tarih_saat container_id|ya da|container_ismi
```

Örnek: `docker logs --since 2020-01-13T11:34:43.154304300Z 12a793b3fec0` (since verilen andan itibaren olanları, until ise verilen ana kadar olanları listeler)

---

### Container tarafından oluşturulan logların belirli sayıda son oluşanlarını görmek:

```
docker logs --tail sayı container_id|ya da|container_ismi
```

Örnek: `docker logs --tail 10 12a793b3fec0` (son 10 log çıktısını listeler)

---

### Container tarafından oluşturulan logları anlık olarak izlemek:

```
docker logs -f container_id|ya da|container_ismi
```

Örnek: `docker logs -f 12a793b3fec0` (loglar oluştuğu anda ekranda gözükecektir. Ctrl-C ile bağlantı kesilebilir)