

## Docker swarm:

- `docker swarm init --advertise-addr 192.168.0.13`

Bu komut , bir Docker Swarm kümesi başlatarak ve bu kümenin ilk yönetici (manager) düğümünü oluşturarak Swarm modunu etkinleştirir. Komutun genel yapısı şu şekildedir:

- **docker swarm init** : Swarm modunu başlatmak için kullanılan komuttur. Bu komut, küme yöneticisi olarak kullanılabilecek bir Docker düğümünü başlatır.
- **--advertise-addr <IP\_ADRESİ>** : Bu seçenek, kümenin dış dünyadan erişilebilecek IP adresini belirtir. **192.168.0.13** burada örnek bir IP adresidir; gerçek IP adresinizi bu değerle değiştirmelisiniz. Bu adres, diğer düğümlerin bu kümenin yönetici düğümüne katılmasında kullanılacak.

Komutun çıktısı, küme yöneticisi (manager) olacak düğümün bilgilerini içerir ve aynı zamanda yeni düğümlerin bu kümeye katılmasında kullanılabilecek olan bir **docker swarm join** komutu üretir. Örneğin:

Swarm initialized: current node (abc123) is now a manager.

To add a worker to this swarm, run the following command:

```
docker swarm join --token <TOKEN> <IP_ADRESİ>:<PORT>
```

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

Bu çıktıda **<TOKEN>** ve **<IP\_ADRESİ>:<PORT>** gibi bilgiler yer alır. Bu bilgiler, diğer düğümlerin Swarm kümesine katılmasında kullanılacaktır. Yönetici düğümü başarıyla oluşturulduktan sonra, Swarm kümeniz hazır durumda olacaktır.

- `docker swarm join-token manager`

Bu komut , bir Docker Swarm kümesine yönetici (manager) düğüm eklemek için kullanılan bir Docker komutudur.

Bu komutun çıktısı, yeni bir yönetici düğüm eklemek isteyen bir kullanıcıya bu işlemi gerçekleştirmek için gereken join token'ı sağlar. Bu join token'ı, diğer düğümlerin bu kümenin yönetici düğümü olmasına izin verir.

Örneğin:

```
docker swarm join --token
SWMTKN-1-3v7gikc1x8x0zq5llg08lf8rtcsyl2t5rzz2kl83wvf8vxxyy9-3gsqfys0c
qy52s2a8ynhd2ef2 192.168.1.10:2377
```

- **docker swarm join-token worker**

Bu komut ise , bir Docker Swarm kümesine çalışan (worker) düğüm eklemek için kullanılan bir Docker komutudur.

Bu komutun çıktısı, yeni bir çalışan düğüm eklemek isteyen bir kullanıcıya bu işlemi gerçekleştirmek için gereken join token'ı sağlar. Bu join token'ı, diğer düğümlerin bu kümenin çalışan düğümü olmasına izin verir.

Örnek bir çıktı:

```
docker swarm join --token
SWMTKN-1-3v7gikc1x8x0zq5llg08lf8rtcsyl2t5rzz2kl83wvf8vxxyy9-3gsqfys0c
qy52s2a8ynhd2ef2 192.168.1.10:2377
```

- **docker node ls**

Bu komut , bir Docker Swarm kümesindeki düğümleri listelemek için kullanılan bir Docker komutudur.

Ayrıca bu komut , kümedeki düğümlerin durumunu, rollerini ve diğer bilgilerini görüntülemek için kullanılır.

Bu komutun çıktısı, kümedeki her bir düğüm için aşağıdaki türden bilgileri içerir:

- **ID:** Düğümün benzersiz kimliği.
- **Hostname:** Düğümün ana bilgisayar adı.
- **STATUS:** Düğümün durumu (ready, down, pending, vb.).
- **AVAILABILITY:** Düğümün kullanılabilirlik durumu (active, pause, drain).
- **MANAGER STATUS:** Eğer düğüm bir yönetici (manager) düğüm ise, yönetici durumu.
- **ENGINE VERSION:** Docker Engine sürümü.

Örnek bir çıktı:

ID	HOSTNAME	STATUS	AVAILABILITY
----	----------	--------	--------------

MANAGER STATUS	ENGINE VERSION		
abc1234567890	node1	Ready	Active
Leader	20.10.3		
xyz0987654321	node2	Ready	Active
20.10.3			

Bu çıktıda, iki düğüm (**node1** ve **node2**) listelenmiştir. **MANAGER STATUS** sütunu, kümenin yönetici düğümünün durumunu gösterir. Eğer bir düğüm yönetici ise, "Leader" olarak işaretlenir. **STATUS** sütunu, düğümün genel durumunu gösterir.

- `docker service create --name test --replicas=5 -p 8080:80 nginx`

Bu komut , Docker Swarm'da bir hizmet (service) oluşturarak belirtilen özelliklere sahip bir Nginx konteyner kümesi başlatmayı amaçlar. İşte bu komutun detaylı açıklaması:

- **docker service create:** Bu komut, Docker Swarm'da bir hizmet oluşturmayı sağlar. Hizmet, birden çok konteyneri içeren ve bu konteynerleri belirli bir durumda tutmaya çalışan bir yapıdır.
- **--name test:** Oluşturulan hizmete "test" adını verir. Bu, hizmetin ismini belirler.
- **--replicas=5:** Oluşturulan hizmetin beş adet replikasını başlatır. Bu, kümede çalışacak toplamda beş adet Nginx konteyneri anlamına gelir.
- **-p 8080:80:** Host makine üzerinde 8080 portunu, hizmetin içindeki Nginx konteynerinin 80 portuna yönlendirir. Yani, dış dünyadan 8080 portu üzerinden erişim sağlanabilir.
- **nginx:** Oluşturulan hizmetin temel imaj olarak kullanılacak olan Docker imajını belirtir. Bu durumda, resmi Nginx imajı kullanılır.

Sonuç olarak, bu komut, adı "test" olan bir Docker Swarm hizmeti oluşturur. Bu hizmet, beş adet Nginx konteynerini barındırır ve dış dünyadan 8080 portu üzerinden erişilebilir. Bu hizmet, küme içinde belirtilen sayıda çalışan konteynerleri sürdürmeye çalışır ve gerektiğinde yeniden başlatır. Bu tür hizmetler, ölçeklenebilir ve yüksek uyumluluklu uygulamalar oluşturmak için kullanılır.

- `docker service ps test`

Bu komut , az önce kullandığımız Docker komutundaki '**test**' servisinin altında çalışan konteynerlerin durumunu ve ayrıntılarını görebiliriz.

Örnek bir çıktı şu şekilde olabilir:

ID	NAME	IMAGE	NODE	DESIRED STATE	CURRENT STATE	ERROR	PORTS
abc123456789	test.1	nginx:latest	node1	Running	Running 5 minutes ago		
xyz098765432	test.2	nginx:latest	node2	Running	Running 5 minutes ago		
lmn456789012	test.3	nginx:latest	node3	Running	Running 5 minutes ago		

Bu çıktıda her bir satır, bir konteyneri temsil eder. **DESIRED STATE** ve **CURRENT STATE** sütunları, konteynerin başlangıçta ne durumda olması gerektiğini ve şu anda ne durumda olduğunu gösterir. **NODE** sütunu, konteynerin hangi düğümde çalıştığını belirtir. Bu bilgiler, hizmetinizi izlemede ve sorunları tespit etmede size yardımcı olabilir.