

KTD-012

Docker Compose ile Çoklu Container Yönetimi

Kemal Tokmak Dokümantasyon Serisi

Okuma Süresi	30-40 dakika
Bölüm Sayısı	13 bölüm
Seviye	Başlangıç - Orta
Son Güncelleme	Temmuz 2026

Docker Compose ile Çoklu Container Yönetimi

Bu dokümanda Docker Compose kullanarak birden fazla container'ı tek dosya üzerinden nasıl yöneteceğimizi, compose.yaml yapısını, servis tanımlamayı, network ve volume kullanımını adım adım öğreneceğiz.

Gerçek Proje Notu

kemaltokmak.com projesinde web sitesi container'ı, Nginx Proxy Manager ve ileride eklenecek servisler Docker Compose ile daha düzenli yönetilebilir.

Bölüm 1

Docker Compose Nedir?

Docker Compose, birden fazla container'ı tek bir YAML dosyası üzerinden tanımlamamızı ve yönetmemizi sağlayan Docker aracıdır.

Tek tek uzun docker run komutları yazmak yerine servisleri, portları, volume bağlantılarını ve network ayarlarını compose.yaml dosyasında düzenli şekilde tutabiliriz.

Bölüm 2

Neden Docker Compose Kullanılır?

Docker Compose özellikle birden fazla container'ın birlikte çalıştığı projelerde yönetimi kolaylaştırır.

- Container ayarları tek dosyada tutulur.
- Servisler tek komutla başlatılır.
- Network ve volume tanımları düzenli olur.
- Projeyi başka sunucuya taşımak kolaylaşır.
- Tekrar kurulabilir bir yapı oluşturur.

Bölüm 3

Proje Klasörü Hazırlama

Her Docker Compose projesi için ayrı bir klasör kullanmak düzenli bir yapı sağlar.

```
cd /opt/docker
mkdir -p compose-ornek
cd compose-ornek
```

Bu klasör içinde compose.yaml dosyamız ve gerekli proje dosyaları bulunacaktır.

Bölüm 4

compose.yaml Dosyası Oluşturma

Docker Compose dosyası genellikle compose.yaml veya docker-compose.yml adıyla oluşturulur.

```
nano compose.yaml
```

Basit bir Nginx container örneği:

```
services:
  web:
    image: nginx:alpine
    container_name: compose-web
    restart: unless-stopped
    ports:
      - "8080:80"
```

Bölüm 5

Servis Tanımlama

Compose dosyasında her container bir servis olarak tanımlanır. Örneğin web, database veya proxy gibi servisler oluşturabiliriz.

```
services:
  web:
    image: nginx:alpine
    container_name: web-container

  db:
    image: mariadb:latest
    container_name: db-container
    environment:
      MYSQL_ROOT_PASSWORD: guclu_sifre
```

Dikkat

Gerçek projelerde şifreleri doğrudan compose.yaml içine yazmak yerine .env dosyası kullanmak daha güvenlidir.

Bölüm 6

Container'ları Başlatma

compose.yaml dosyasının bulunduğu klasörde şu komut çalıştırılır:

```
docker compose up -d
```

-d parametresi container'ların arka planda çalışmasını sağlar.

Bölüm 7

Container Durumunu Kontrol Etme

Çalışan container'ları görmek için:

```
docker compose ps
```

Genel Docker container listesini görmek için:

```
docker ps
```

Bölüm 8

Container Loglarını Görme

Servislerde hata olup olmadığını anlamak için log kontrolü yapılır.

```
docker compose logs
docker compose logs web
docker compose logs -f
```

Bölüm 9

Docker Compose ile Network Kullanımı

Servislerin aynı network üzerinde haberleşebilmesi için Compose içinde network tanımı yapılabilir.

```
services:
  web:
    image: nginx:alpine
    networks:
      - app-network

  db:
    image: mariadb:latest
    networks:
      - app-network

networks:
  app-network:
    driver: bridge
```

Bilgi

Aynı network içindeki container'lar birbirlerine servis isimleriyle erişebilir. Örneğin web container, db servisine db adıyla ulaşabilir.

Bölüm 10

Docker Compose ile Volume Kullanımı

Volume kullanarak container içindeki verileri kalıcı hale getirebiliriz.

```
services:
  web:
    image: nginx:alpine
    volumes:
      - ./html:/usr/share/nginx/html

  db:
    image: mariadb:latest
    volumes:
```

```
- db-data:/var/lib/mysql

volumes:
  db-data:
```

İpucu

Site dosyaları gibi düzenli değişen içeriklerde bind mount, veritabanı gibi kalıcı verilerde named volume kullanmak daha uygundur.

Bölüm 11

Compose Projesini Güncelleme

Compose dosyasında değişiklik yaptıktan sonra container'ları yeniden oluşturmak gerekir.

```
docker compose up -d
```

İmajları güncellemek için:

```
docker compose pull
docker compose up -d
```

Bölüm 12

Compose Projesini Durdurma ve Temizleme

Container'ları durdurmak, kaldırmak ve gerekiyorsa volume verilerini temizlemek için aşağıdaki komutlar kullanılır.

```
docker compose stop
docker compose down
docker compose down -v
```

Dikkat

docker compose down -v komutu volume verilerini de silebilir. Veritabanı gibi önemli veriler varsa dikkatli kullanılmalıdır.

Bölüm 13

Bu Dokümanda Ne Öğrendik?

Bu dokümanda Docker Compose ile çoklu container yönetimini, compose.yaml yapısını, servis tanımlamayı, network ve volume kullanımını öğrendik.

- Docker Compose'un ne işe yaradığını öğrendik.
- compose.yaml dosyası oluşturduk.
- Servis, port, network ve volume tanımları yaptık.
- Container'ları tek komutla başlattık.

- Log ve durum kontrollerini öğrendik.
- Compose projesini durdurma ve temizleme komutlarını gördük.

Sonraki Doküman

KTD-013 — Docker Compose ile Web Sitesi Yayınlama