

# ÜMMÜ GÜLSÜM VARLI

Verilen Docker compose ve Dockerfile dosyalarının  
incelemesi yapılmıştır.

25.07.2024

**Ödev Tanımı:** İkinci kısımda dockerlıdockerize\_me örneğinin Dockerfile ve Docker compose dosyaları içerisindeki komutlar neden kullanılmış, görevleri nelerdir açıklayınız.

### DOCKER COMPOSE

```
version: '3'

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

  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

networks:
  net:
    driver: bridge

volumes:
  db_data:
```

Kod Kısmı:	Kodun İşlevi:
<pre>version: '3'</pre>	<p>Önemi yok, sadece versiyon belirtmek için kullanılır. Kullanılmaması durumunda sorun oluşturmaz. Kısaca Docker Compose dosyasının kullanılan versiyonu belirler.</p>
<pre>services:   app:     build:       context: .       dockerfile: Dockerfile     depends_on:       - db     ports:       - "80:80"     networks:       - net</pre>	<p>Bu “docker compose”, “app” ve “db” hizmetini sağlar. Burada “app” hizmetine dair olanları açıklayacak olursak;</p> <ul style="list-style-type: none"> <li>• <b>build:</b> Uygulamanın Docker imajının nasıl oluşturulacağını belirtir.</li> <li>• <b>context:</b> mevcut dizin</li> <li>• <b>dockerfile:</b> Docker dosyasının adı burada dosyanın adı “Dockerfile” olarak belirtilmiştir.</li> <li>• <b>depends_on:</b> Uygulamanın servisleri arasındaki bağımlılıkları ifade etmek ve bu servislerin başlatılıp durdurulacağı sırayı belirtmek için kullanılır . Bu örnekte, “db”ye bağlıdır.</li> <li>• <b>ports:</b> Uygulamanızın dış dünyaya hangi port üzerinden açılacağını belirtir. "80:80" hostun 80 numaralı portundan Docker container'ının 80 numaralı portuna yönlendirme yapar.</li> <li>• <b>networks:</b> Servisin hangi ağda çalışacağını belirtir. net adında bir ağda çalışacak. Bir kullanıcının bir Docker konteynerini ihtiyaç duyduğu kadar çok ağa bağlamasını sağlar .</li> </ul>
<pre>db:   image: mysql:latest   environment:     - MYSQL_DATABASE=yavuzlar     - MYSQL_ROOT_PASSWORD=1</pre>	<p>Bu “docker compose”, “app” ve “db” hizmetini sağlar. Burada “db” hizmetine dair olanları açıklayacak olursak;</p> <ul style="list-style-type: none"> <li>• <b>image:</b> Kullanılacak Docker imajının adı ve sürümü. Burada <i>mysql:latest</i> yani <i>en son MySQL imajı</i> kullanılıyor.</li> <li>• <b>environment:</b> MySQL için ortam değişkenlerini belirtir. <i>MYSQL_DATABASE</i> veritabanı adını, <i>MYSQL_ROOT_PASSWORD</i> ise root kullanıcısının şifresini belirtir. Docker Compose'daki ortam değişkenlerini ve enterpolasyonu kullanarak çok yönlü ve yeniden kullanılabilir yapılandırmalar oluşturabilir.</li> </ul>
<pre>volumes:   - db_data:/var/lib/mysql   - ./yavuzlar_messages.sql:/docker- entrypoint- initdb.d/yavuzlar_messages.sql</pre>	<ul style="list-style-type: none"> <li>• <b>volumes:</b> Amaç, verileri konteynerin dışında kalıcı hale getirmektir, böylece yedeklenebilir veya paylaşılabılır . Docker birimleri, Docker'ın dosya sistemine bağımlıdır.</li> </ul>

<pre> ports:   - "8080:3306" networks:   - net </pre>	<ul style="list-style-type: none"> <li>• Docker container'ının verilerini nereye saklayacağını belirtir. db_data adlı bir Docker volumü oluşturuluyor ve MySQL veritabanı verileri <i>/var/lib/mysql</i> dizininde saklanacak. Ayrıca <i>yavuzlar_messages.sql</i> dosyası container başlatılırken <i>docker-entrypoint-initdb.d</i> dizininde olacak, bu dosya MySQL veritabanına başlangıç verileri sağlar.</li> <li>• <b>ports:</b> MySQL servisinin dış dünyaya hangi port üzerinden erişileceğini belirtir. "8080:3306" hostun 8080 numaralı portundan Docker container'ının 3306 numaralı MySQL portuna yönlendirme yapar.</li> <li>• <b>networks:</b> Servisin hangi ağda çalışacağını belirtir. net adında bir ağda çalışacak.</li> </ul>
<pre> networks:   net:     driver: bridge volumes:   db_data: </pre>	<ul style="list-style-type: none"> <li>• <b>net:</b> bridge tipinde bir ağ oluşturur. Docker Compose dosyasındaki tüm servisler bu ağ üzerinde iletişim kurabilir.</li> <li>• <b>volumes:</b> Docker container'larının verilerini depolamak için kullanılan bir Docker volume'u. MySQL veritabanı verileri bu volume üzerinde saklanacak.</li> </ul>

## DOCKER FILE

```
FROM php:7.4-apache

WORKDIR /var/www/html
COPY ./app .
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf
RUN apt-get update
RUN docker-php-ext-install pdo pdo_mysql
EXPOSE 80
```

Kod Kısmı:	Kodun İşlevi:
FROM php:7.4-apache	<ul style="list-style-type: none"><li><b>FROM:</b> Yapının temeli haline gelen mevcut bir imaja atıfta bulunur . Sonraki tüm talimatlar başvurulmuş imajın dosya sisteminin üstüne uygulanır. <i>php:7.4-apache</i> imajını kullanarak, PHP 7.4 ve Apache web sunucusunu içeren bir Docker imajı oluşturulur.</li></ul>
WORKDIR /var/www/html	<ul style="list-style-type: none"><li><b>WORKDIR:</b> Bu komut, konteyner içindeki çalışma dizinini belirler. Bu durumda, <i>/var/www/html</i> dizini çalışma dizini olarak ayarlanır. Bu, Apache web sunucusunun varsayılan belge kök dizinidir.</li></ul>
COPY ./app .	<ul style="list-style-type: none"><li><b>COPY:</b> Bu komut, yerel dosya sisteminizdeki <i>./app</i> dizinindeki tüm dosyaları konteyner içindeki mevcut çalışma dizinine (<i>/var/www/html</i>) kopyalar. Bu şekilde, uygulamanızın dosyaları konteyner içine taşınmış olur.</li></ul>
RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf	<ul style="list-style-type: none"><li>Bu komut, Apache yapılandırma dosyasına (<i>/etc/apache2/apache2.conf</i>) <i>ServerName localhost</i> satırını ekler. Bu, "ServerName" direktifini ayarlayarak Apache'nin kendi hostname'ini belirlemesini sağlar ve bazı uyarıları önler.</li></ul>
RUN apt-get update	<ul style="list-style-type: none"><li>Bu komut, konteyner içindeki paket listesini günceller. <i>apt-get</i> komutu Debian tabanlı sistemlerde paket yönetimi için kullanılır ve <i>update</i> komutu mevcut paket listesini günceller.</li></ul>
RUN docker-php-ext-install pdo pdo_mysql	<ul style="list-style-type: none"><li>Bu komut, PHP için gerekli olan pdo ve pdo_mysql eklentilerini yükler. <i>docker-php-ext-install</i> komutu, PHP eklentilerini yüklemek ve yapılandırmak için kullanılır.</li></ul>
EXPOSE 80	<ul style="list-style-type: none"><li>Belirtilen konteynerin dış dünya ile iletişim kurabileceği portu belirtir.</li></ul>