

Dockerfile:

```
FROM php:7.4-apache                # referans imajı belirtiyoruz

WORKDIR /var/www/html              # çalışma dizinini belirtiyoruz

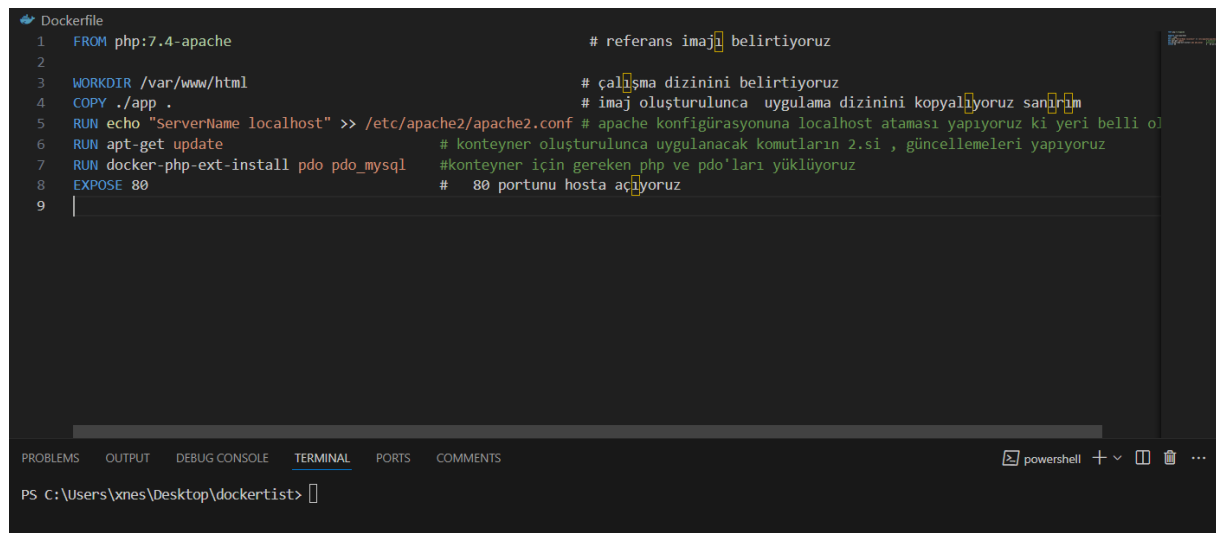
COPY ./app .                       # imaj oluşturulunca uygulama dizinini
kopyalıyoruz sanırım

RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf # apache
konfigürasyonuna localhost ataması yapıyoruz ki yeri belli olsun

RUN apt-get update                  # konteyner oluşturulunca uygulanacak
komutların 2.si , güncellemeleri yapıyoruz

RUN docker-php-ext-install pdo pdo_mysql #konteyner için gereken php ve
pdo'ları yüklüyoruz

EXPOSE 80                           # 80 portunu hosta açıyoruz
```



```
Dockerfile
1 FROM php:7.4-apache                # referans imajı belirtiyoruz
2
3 WORKDIR /var/www/html              # çalışma dizinini belirtiyoruz
4 COPY ./app .                       # imaj oluşturulunca uygulama dizinini kopyalıyoruz sanırım
5 RUN echo "ServerName localhost" >> /etc/apache2/apache2.conf # apache konfigürasyonuna localhost ataması yapıyoruz ki yeri belli ol
6 RUN apt-get update                  # konteyner oluşturulunca uygulanacak komutların 2.si , güncellemeleri yapıyoruz
7 RUN docker-php-ext-install pdo pdo_mysql #konteyner için gereken php ve pdo'ları yüklüyoruz
8 EXPOSE 80                           # 80 portunu hosta açıyoruz
9 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\xnes\Desktop\dockertist> |

Docker compose file:

version: '3' # 1-2 de yapabiliyorduk ama onlar sorunluydu sanırım composer sürümü bu

services: #dockerimiz için gerekli olan servisler

app:

build:

context: .

dockerfile: Dockerfile

depends\_on: # konteyner bağımlılıkları

- db

ports: #portlarımız-> host ile konteyner arası bağlantı için

- "80:80"

networks: #ağ ayarlarımız

- net

db: #veritabanı bağımlılığının detayları burada

image: mysql:latest #imaj dosyamız seçiliyor

environment: #enviroment değişkenleri ile veritabanı için gereken isim ve şifre gibi şeyleri ayarladık

- MYSQL\_DATABASE=yavuzlar

- MYSQL\_ROOT\_PASSWORD=1

volumes: #imajdan sonra volume ayarı burada yapılıyor

- db\_data:/var/lib/mysql

- ./yavuzlar\_messages.sql:/docker-entrypoint-initdb.d/yavuzlar\_messages.sql

ports: #veritabanı için gereken host konteyner arası port geçişi

- "8080:3306"

networks: # veritabanı bağımlılığının ağ ayarını seçiyoruz

- net

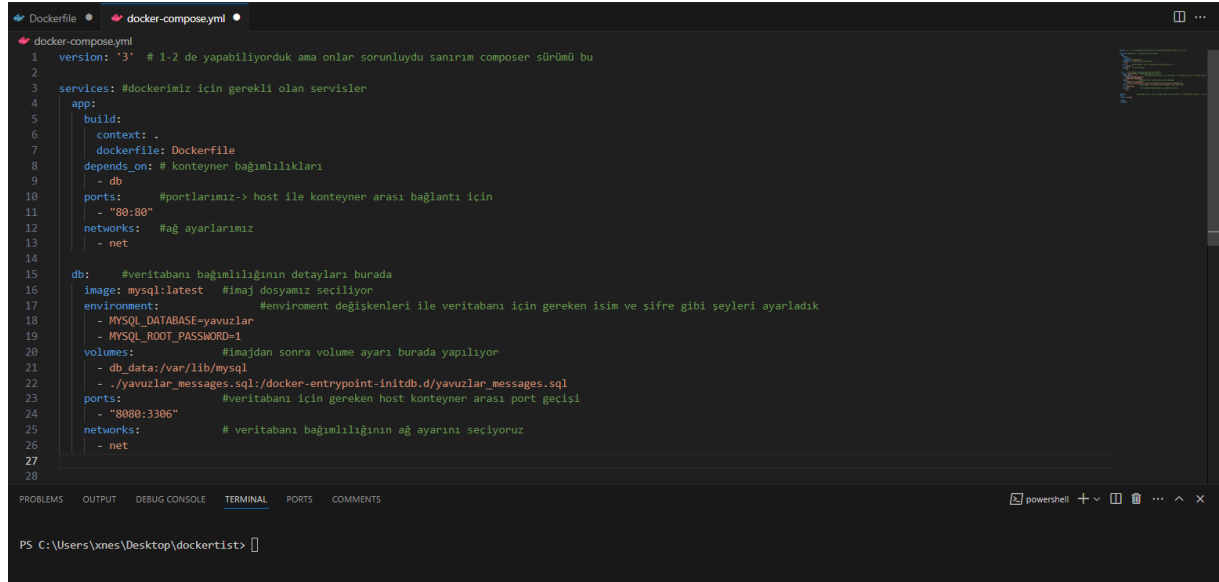
networks: #dockerdeki gibi ağ tipi, bridge olunca seçilen portlar seçilen şekilde açılıyor , en güvenlisi none ama iletişim gerekli.

net:

driver: bridge

volumes:

db\_data:



```
1 version: '3' # 1-2 de yapabiliyorduk ama onlar sorunluydu sanırım composer sürümü bu
2
3 services: #dockerimiz için gerekli olan servisler
4   app:
5     build:
6       context: .
7       dockerfile: Dockerfile
8     depends_on: # konteyner bağımlılıkları
9       - db
10    ports: #portlarımız-> host ile konteyner arası bağlantı için
11      - "80:80"
12    networks: #ağ ayarlarımız
13      - net
14
15   db: #veritabanı bağımlılığının detayları burada
16     image: mysql:latest #imaj dosyamız seçiliyor
17     environment: #enviroment değişkenleri ile veritabanı için gereken isim ve şifre gibi şeyleri ayarladık
18       - MYSQL_DATABASE=yavuzlar
19       - MYSQL_ROOT_PASSWORD=1
20     volumes: #imajdan sonra volume ayarı burada yapılıyor
21       - db_data:/var/lib/mysql
22       - ./yavuzlar_messages.sql:/docker-entrypoint-initdb.d/yavuzlar_messages.sql
23     ports: #veritabanı için gereken host konteyner arası port geçişi
24       - "8080:3306"
25     networks: # veritabanı bağımlılığının ağ ayarını seçiyoruz
26       - net
27
28
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS COMMENTS

PS C:\Users\xnes\Desktop\dockertist>