

# Dockerfile and Docker Compose

Apurwa Singh

January 21, 2025

# Contents

<b>1</b>	<b>Scenario 1: PHP</b>	<b>3</b>
1.1	1.1 index.php . . . . .	3
1.2	1.2 Dockerfile . . . . .	3
1.3	1.3 Build and Run . . . . .	3
<b>2</b>	<b>Scenario 2: Java</b>	<b>4</b>
2.1	2.1 sample.war . . . . .	4
2.2	2.2 Dockerfile . . . . .	4
2.3	2.3 Build and Run . . . . .	4
<b>3</b>	<b>Scenario 3: Python</b>	<b>5</b>
3.1	3.1 app.py . . . . .	5
3.2	3.2 pyreq.txt . . . . .	5
3.3	3.3 Dockerfile . . . . .	5
3.4	3.4 Build and Run . . . . .	5
<b>4</b>	<b>Pushing Docker Images to Docker Hub</b>	<b>6</b>
<b>5</b>	<b>Docker Compose: Wordpress Setup</b>	<b>7</b>
<b>6</b>	<b>Docker Compose: Python Flask App</b>	<b>8</b>

# 1 Scenario 1: PHP

## 1.1 1.1 index.php

```
1 <?php
2 echo '<h1>Welcome to the world of DevOps</h1>';
3 ?>
```

## 1.2 1.2 Dockerfile

```
1 FROM php:8.2-apache
2 COPY index.php /var/www/html
3 WORKDIR /var/www/html
4 EXPOSE 8008
5 CMD ["apache2-foreground"]
```

## 1.3 1.3 Build and Run

```
1 docker build -t papache .
2 docker run -dit -p 8008:80 --name papa papache
```

Access the web app at localhost:8008.

## 2 Scenario 2: Java

### 2.1 2.1 sample.war

sample.war

### 2.2 2.2 Dockerfile

```
1 FROM tomcat:9-jdk11
2 COPY sample.war /usr/local/tomcat/webapps/ROOT.war
3 EXPOSE 8080
4 CMD ["catalina.sh", "run"]
```

### 2.3 2.3 Build and Run

```
1 docker build -t javawar .
2 docker run -dit -p 8080:8080 --name javasrv javawar
```

Access the web app at localhost:8080.

## 3 Scenario 3: Python

### 3.1 3.1 app.py

```
1 from flask import Flask
2
3 app = Flask(__name__)
4
5 @app.route('/')
6 def webout():
7     return '<h1>DevOps is fun.</h1>'
8
9 app.run(host='0.0.0.0', port=7000)
```

### 3.2 3.2 pyreq.txt

```
1 Flask==2.0.3
2 Werkzeug==2.0.3
```

### 3.3 3.3 Dockerfile

```
1 FROM python:3.9-slim
2 COPY . /app
3 WORKDIR /app
4 RUN pip install --no-cache-dir -r pyreq.txt
5 EXPOSE 7000
6 CMD ["python", "app.py"]
```

### 3.4 3.4 Build and Run

```
1 docker build -t apurwasingh/flask .
2 docker run -dit -p 8000:7000 --name flask apurwasingh/flask
```

Access the web app at localhost:8000.

## 4 Pushing Docker Images to Docker Hub

```
1 docker login -u apurwasingh
2 # Input password
3 docker push apurwasingh/flask
4 docker logout
```

## 5 Docker Compose: Wordpress Setup

```
1 version: '3'
2 services:
3   wordpress:
4     image: wordpress
5     container_name: wordweb
6     ports:
7       - "8070:80"
8     volumes:
9       - wordwebvol:/var/www/html
10    depends_on:
11      - db
12    restart: unless-stopped
13  db:
14    image: mysql:8
15    container_name: worddb
16    environment:
17      MYSQL_ROOT_PASSWORD: nepal
18      MYSQL_DATABASE: devdb
19      MYSQL_USER: devuser
20      MYSQL_PASSWORD: devpwd
21    volumes:
22      - worddbvol:/var/lib/mysql
23    restart: unless-stopped
24
25 volumes:
26   wordwebvol:
27   worddbvol:
```

Run with `docker-compose up -d`. Access at `localhost:8070`.

## 6 Docker Compose: Python Flask App

```
1 services:  
2   flask:  
3     image: apurwasingh/flask  
4     container_name: flask  
5     ports:  
6       - "8000:7000"
```

Run with `docker-compose up -d`. Access at `localhost:8001`.