# Building and Running a Custom

# Apache Server Docker Image

Step-by-Step Guide with Advanced Features for Dynamic Web Hosting


Step-by-Step Guide to Build, Test, and Verify the Docker Image

## Step 1. Prepare the Workspace

1. Create the Dockerfile: Save the following content into a file named
    - Vim Dockerfile

```
controlplane $ vi Dockerfile
```

```
FROM docker.io/ubuntu
MAINTAINER vaibhavsurase@gmail.com
ARG PORT=8080
RUN apt update -y && apt install apache2 -y
RUN sed -i 's/Listen 80/Listen '$PORT'/g' /etc/apache2/ports.conf
COPY index.html /var/www/html/
RUN echo "Running on $PORT" >> /var/www/html/index.html
EXPOSE $PORT
CMD ["/usr/sbin/apache2ctl", "-D", "FOREGROUND"]
```

2. **Create the index.html file**: Create a simple index.html file:
    - Vim index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Docker Apache</title>
</head>
<body>
    <h1>Welcome to Docker Apache Server</h1>
</body>
</html>
```

## Step 2. Build the Docker Image

1. Run the following command to build the image:
    - docker build -t apache-docker --build-arg PORT=8080 .

```
controlplane $ docker build -t apache-docker --build-arg PORT=8080 .
[+] Building 52.2s (10/10) FINISHED                                                                                    docker:default
 => [internal] load build definition from Dockerfile                                                                             0.1s
 => => transferring dockerfile: 430B                                                                                             0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest                                                                 1.2s
 => [internal] load .dockerignore                                                                                                0.0s
 => => transferring context: 2B                                                                                                  0.0s
 => [1/5] FROM docker.io/library/ubuntu:latest@sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782           4.8s
 => => resolve docker.io/library/ubuntu:latest@sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782           0.1s
 => => sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782 6.69kB / 6.69kB                                   0.0s
 => => sha256:3afff29dffbc200d202546dc6c4f614edc3b109691e7ab4aa23d02b42ba86790 424B / 424B                                       0.0s
 => => sha256:a04dc4851cbcbb42b54d1f52a41f5f9eca6a5fd03748c3f6eb2cbeb238ca99bd 2.30kB / 2.30kB                                   0.0s
 => => sha256:5a7813e071bfadf18aaa6ca8318be4824a9b6297b3240f2cc84c1db6f4113040 29.75MB / 29.75MB                                 1.9s
 => => extracting sha256:5a7813e071bfadf18aaa6ca8318be4824a9b6297b3240f2cc84c1db6f4113040                                        2.6s
 => [internal] load build context                                                                                                0.1s
 => => transferring context: 186B                                                                                                0.0s
 => [2/5] RUN apt update -y && apt install apache2 -y                                                                           40.3s
 => [3/5] RUN sed -i 's/Listen 80/Listen '8080'/g' /etc/apache2/ports.conf                                                       0.9s
```

- -t apache-docker: Tags the image as apache-docker.
- --build-arg PORT=8080: Passes the PORT argument to replace the default value.

```
controlplane $ docker images
REPOSITORY      TAG       IMAGE ID       CREATED          SIZE
apache-docker   latest    209bdbfd6607   3 minutes ago    238MB
```

## Step 3. Run the Docker Container

1. Run a container from the image:
   - docker run -d -p 8080:8080 --name apache-test apache-docker

```
controlplane $ docker run -d -p 8080:8080 --name apache-test apache-docker
f26f7f0d473236945b1f4d72da9f57cff0900d1e4a30b2d04fcb65ef8841bcc3
```

   - -d: Runs the container in detached mode.
   - -p 8080:8080: Maps port 8080 of the container to port 8080 of the host.
   - --name apache-test: Names the container apache-test.

## Step 4. Test the Container 1. Check if the container is running:

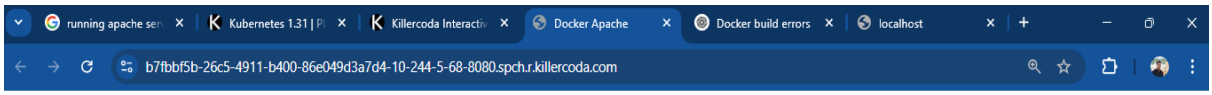
- docker ps

```
controlplane $ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS         PORTS                                       N
AMES
f26f7f0d4732   apache-docker  "/usr/sbin/apache2ct…"   3 minutes ago   Up 3 minutes   0.0.0.0:8080->8080/tcp, :::8080->8080/tcp   a
pache-test
```

2. **Access the website**: Open your browser and navigate to:
   - **Check on localhost**
     - Curl localhost:8080

```
controlplane $ curl localhost:8080
<!DOCTYPE html>
<html>
<head>
    <title>Docker Apache</title>
</head>
<body>
    <h1>Welcome to Docker Apache Server</h1>
</body>
</html>
Running on 8080
```

- http://localhost:8080




# Welcome to Docker Apache Server

Running on 8080

You should see the index.html content with an additional line, "Running on 8080".

Thank you ..