Building an Apache Web Server through a Dockerfile and uploading it to Docker Hub

What is Apache Server?

The **Apache HTTP Server** is a free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation.

Source: Wikipedia

Need to Dockerize Apache WebServer

Setting up an Apache server on the workstation requires extensive configuration.
To reduce this effort, Docker introduced the concept of Dockerfile to easily create and set up configurations.

Steps to create an Apache Server through a Dockerfile

Step 1: First, we use the mkdir command to create a special directory for all Apache-related files.

```
$ mkdir apache2
```

Step 2:Having created a folder, now we go ahead and create a Dockerfile within that folder with the vi editor:

```
$ cd apache2/
~/apache2$ vi Dockerfile
```

Once we run the previous command, a vi editor opens. Paste the following content into the Dockerfile:

```
FROM ubuntu
RUN apt update
RUN apt-get install apache2 -y
RUN apt-get install apache2-utils -y
RUN apt clean
EXPOSE 80
CMD [“apache2ctl”, “-D”, “FOREGROUND”]
```

To exit the editor, press ESC then:wq then Enter.

Step 3:Tag and build the Docker image.

Now, we build the Dockerfile using the docker build command. Within this, we tag the image to be created as 1.0 and give a customized name to our image (i.e., apache_webserver).

```
docker build -t apache_webserver:1.0 .
```

The output of which will look like this:

```
mudasir@ubuntuserver:~/apache2$ docker build -t apache_webserver:1.0 .
[+] Building 20.1s (10/10) FINISHED
 => [internal] load build definition from Dockerfile
0.0s
 => => transferring dockerfile: 208B
0.0s
 => [internal] load .dockerignore
0.0s
 => => transferring context: 2B
0.0s
 => [internal] load metadata for docker.io/library/ubuntu:latest
1.2s
 => [auth] library/ubuntu:pull token for registry-1.docker.io
0.0s
 => [1/5] FROM
docker.io/library/ubuntu@sha256:67211c14fa74f070d27cc59d69a7fa9aeff8e28ea118
```

```
ef3babc295a0428a6d21        0.0s
 => CACHED [2/5] RUN apt-get update
0.0s
 => [3/5] RUN apt install apache2 -y
15.8s
 => [4/5] RUN apt install apache2-utils -y
1.8s
 => [5/5] RUN apt clean
0.5s
 => exporting to image
0.8s
 => => exporting layers
0.8s
 => => writing image
sha256:fbaffc7acf8c2d303bd0694d749610c1e29ae7c05f85e15a512bea9059d30841
0.0s
 => => naming to docker.io/library/apache_webserver:1.0
```

Once the image has been built, we should check for the presence of the image using docker images command.

```
mudasir@ubuntuserver:~/apache2$ docker images
REPOSITORY         TAG       IMAGE ID       CREATED              SIZE
apache_webserver   1.0       fbaffc7acf8c   About a minute ago   228MB
```

Step 4: Run the Docker image as a container

Once the image is created, run the image locally as a container:

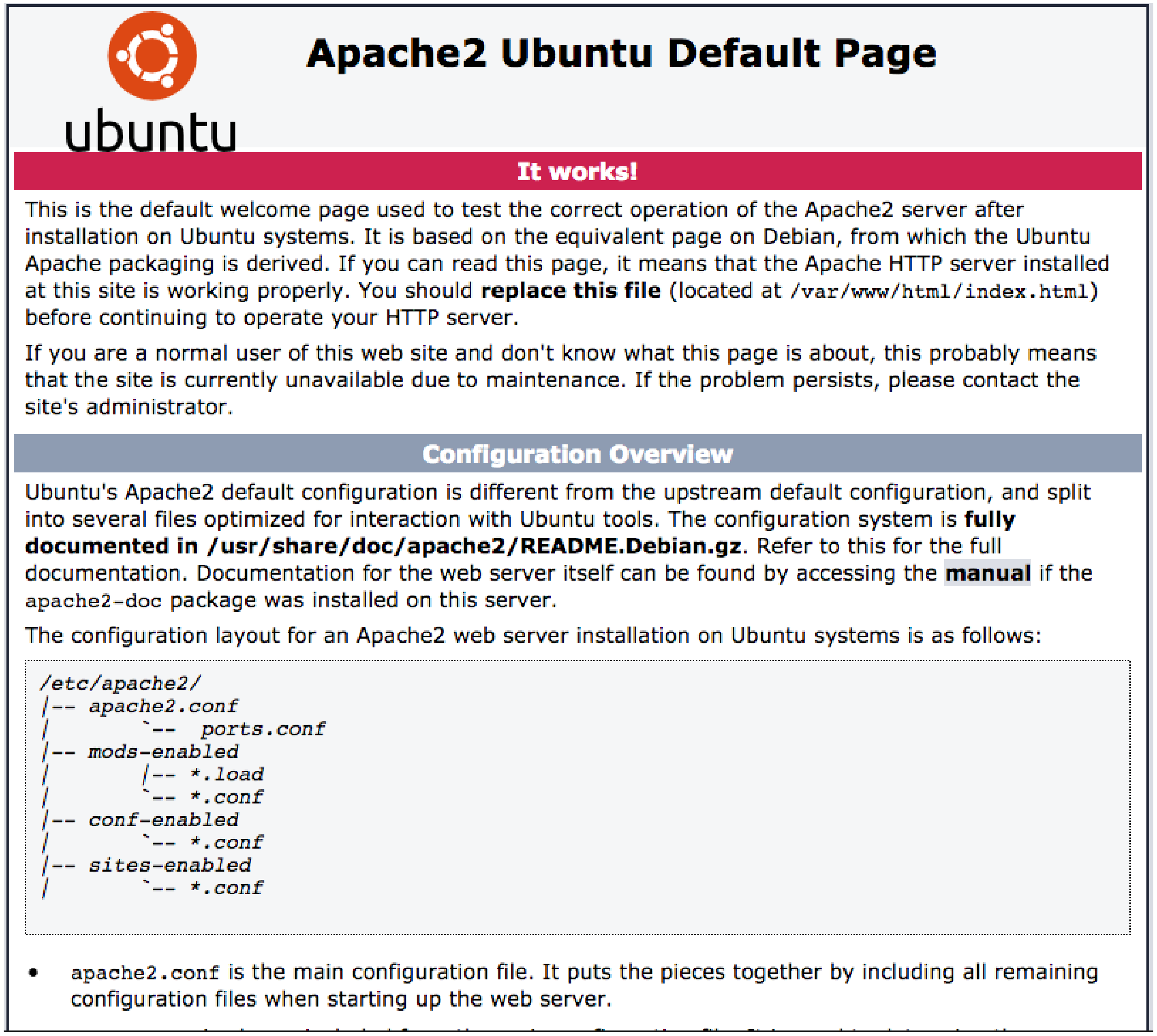
We run the container in the detached mode so that it runs continuously in the background. Add -d to the docker run command.
To host the Apache server, we provide port 80 (HTTP) for it. Use -p 8080:80 to run the server on localhost.
So, the docker run command also takes the image along with the associated tag as input to run it as a container.

```
docker run --name myapache -d -p 8080:80 apache_webserver:1.0
```

```
mudasir@ubuntuserver:~/apache2$ docker ps
CONTAINER ID   IMAGE                COMMAND                CREATED        STATUS
PORTS     NAMES
b6530e137e89   apache_webserver:1.0   "/bin/sh -c '[“apach…"   6 seconds ago
Created            myapache
```

Step 5: Verify the online presence of Apache Server

In order to test the presence of Apache server on the system, visit any local browser and type localhost: or IP address of the Server



Apache2 Ubuntu Default Page

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in /usr/share/doc/apache2/README.Debian.gz**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|        `--  ports.conf
|-- mods-enabled
|        |-- *.load
|        `-- *.conf
|-- conf-enabled
|        `-- *.conf
|-- sites-enabled
|        `-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.

Step 5:How to push the image to Docker Hub

We need to log into our Docker Hub account in order to push the Image. The command to log in using the terminal is

```
docker login -u NAME
```

Where NAME is your Docker Hub username. You will be prompted for your Docker Hub password. You can also use your Access Token.

Finally, we’re going to tag our new image and then push it to Docker Hub. First tag the image with: 1.0 using the command:

```
docker image tag apache_webserver:1.0 bmudasir/myapache:1.0
```

Where bmudasir is my username.

Now that the image is tagged, we can push it to Docker Hub with

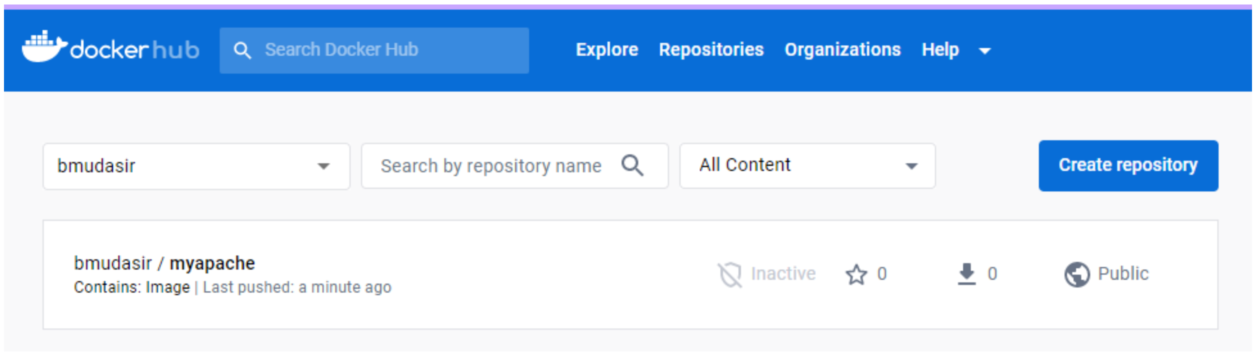
```
docker image push bmudasir/myapache:1.0
```

The Output of which will look as

```
mudasir@ubuntuserver:~$ docker image push bmudasir/myapache:1.0
The push refers to repository [docker.io/bmudasir/myapache]
a5248763301d: Pushed
ce27873e1018: Pushed
4129e7eff7e8: Pushed
534649edda20: Pushed
b93c1bd012ab: Mounted from library/ubuntu
1.0: digest:
sha256:f4089d4369e1e3d8217be194bac2965825ffd8e81c8670b6c3b3bd70
d99644b7 size: 1367
```

When the push completes, you should find the myapache:1.0 image in your Docker Hub repository.

The Screenshot of Docker Hub will look as



bmudasir / myapache
Contains: Image | Last pushed: a minute ago
Inactive 0 0 Public

And that’s all there is to build a Docker image and push it to your Docker Hub repository.

Thanks