Create a container with Ubuntu image on the top of CentOS 9 or Rocky Stream Operating System.

It means we can create different OS containers.

Create Apache Web server

- Download Apache Web server package, vim, curl elinks on the Ubuntu container.
- Configure Apache Web server.
- Create our own Directory
- Create your HTML page.
- How to stop, start, restart web server.

Create an image from the running container.

Create a new container from newly created image.

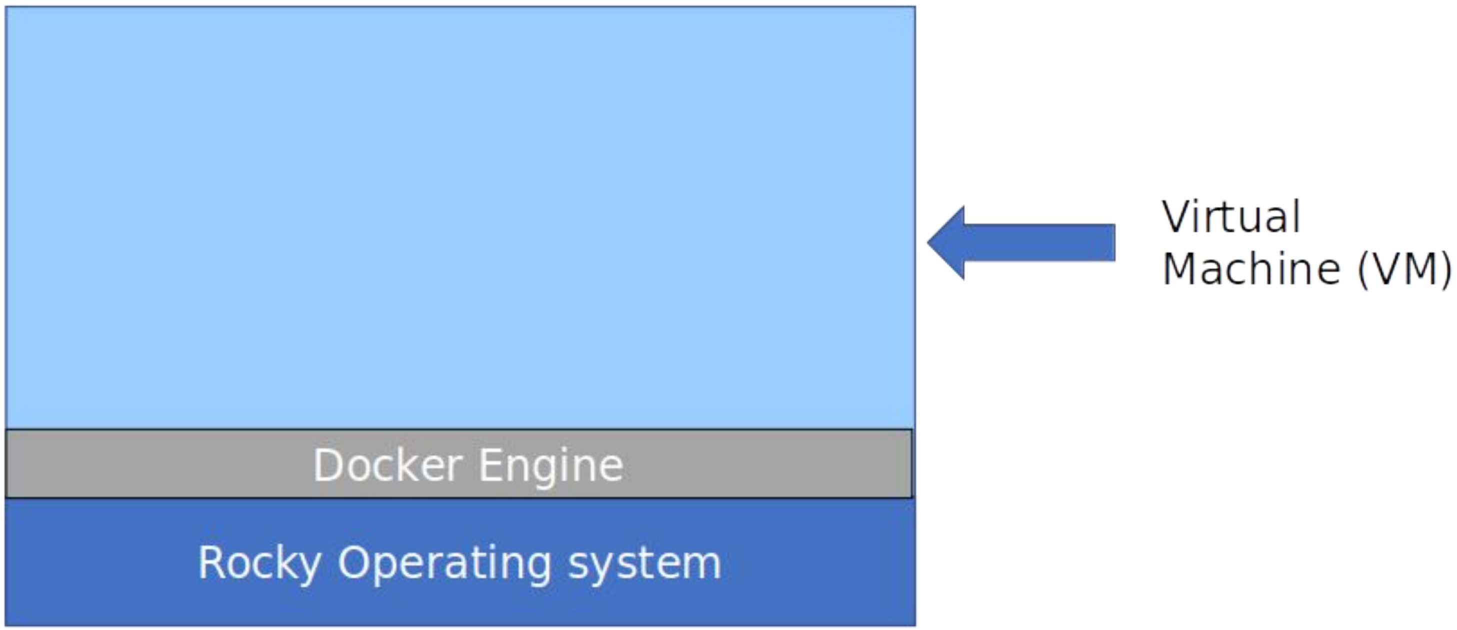
Create an Image from Dockerfile.

What is image tagging.

How to push the image to Docker Hub repository. So that other people can use it.

Delete the image from the server cache and then download the image from Docker hub and then create a container.

```
yum remove docker  docker-client  docker-client-latest   docker-common  docker-latest docker-latest-logrotate  docker-logrotate docker-engine podman buildah -y
```

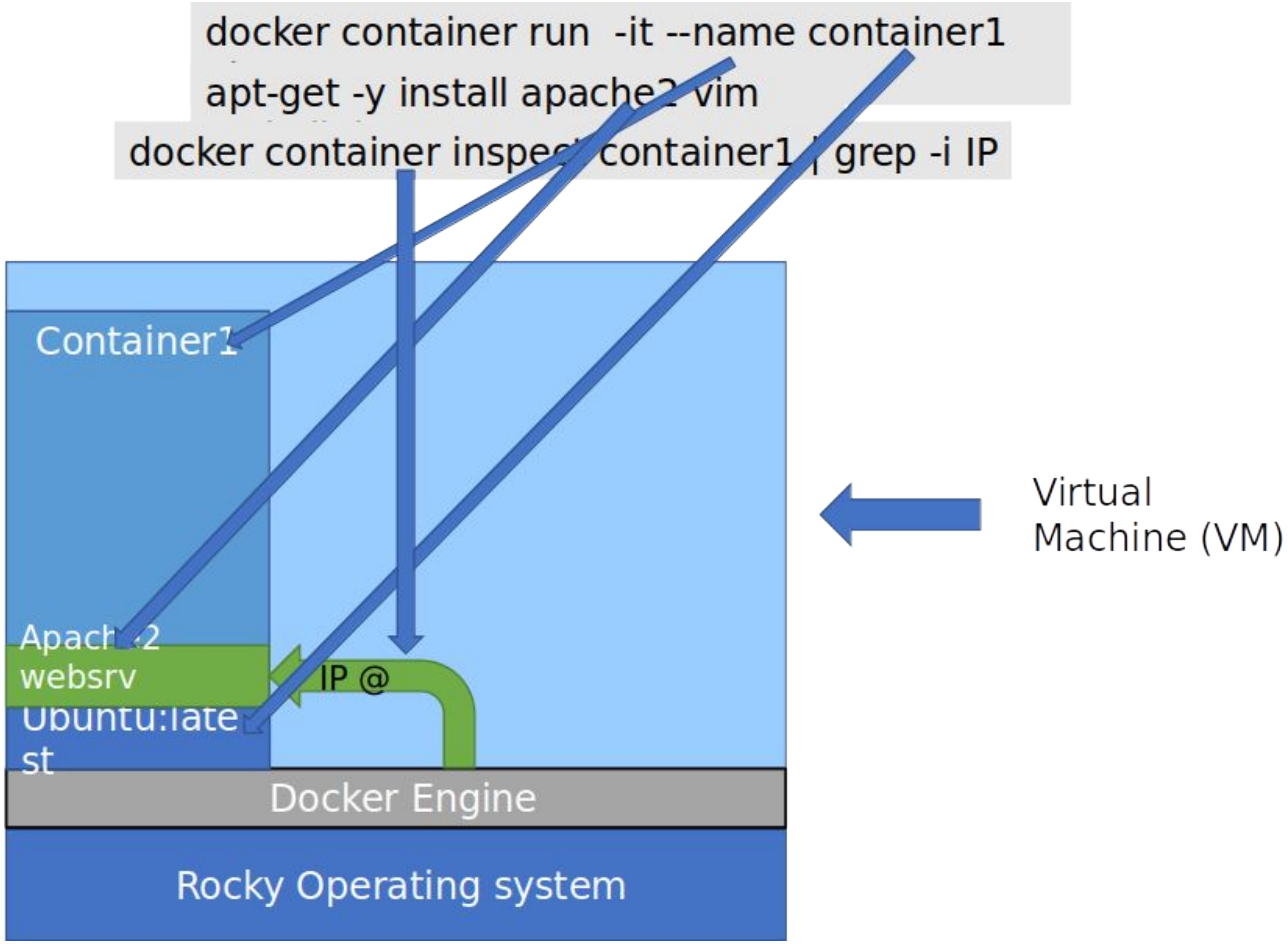


Virtual Machine (VM)

Docker Engine

Rocky Operating system

```
yum install docker-ce docker-ce-cli containerd.io -y
```



```
docker container run  -it --name container1
apt-get -y install apache2 vim
docker container inspect container1 | grep -i IP
```

Configuration of Apache2 web server

apt-get update

Main configuration files

apt-get -y install apache2 vim curl elinks

```
root@1e82edbe185d:/# cd /etc/apache2/
root@1e82edbe185d:/etc/apache2# ls -ltr
total 64
-rw-r--r-- 1 root root   320 Oct  1  2023 ports.conf
-rw-r--r-- 1 root root 31063 Oct  1  2023 magic
-rw-r--r-- 1 root root  1782 Oct  1  2023 envvars
drwxr-xr-x 2 root root   143 Apr 23 12:48 conf-available
drwxr-xr-x 2 root root  8192 Apr 23 12:48 mods-available
drwxr-xr-x 2 root root  4096 Apr 23 12:48 mods-enabled
drwxr-xr-x 2 root root   143 Apr 23 12:48 conf-enabled
-rw-r--r-- 1 root root  7245 Apr 23 12:50 apache2.conf
drwxr-xr-x 2 root root    77 Apr 23 12:53 sites-available
drwxr-xr-x 2 root root    29 Apr 23 12:53 sites-enabled
root@1e82edbe185d:/etc/apache2#
```

apache2.conf: This is the main Apache2 configuration file. Contains settings that are global to Apache2.

ports.conf: This directives determine which TCP ports Apache2 will listening on.

envvars: This file hold the environment variables are set in Apache2.

conf-available: This directory contains available configuration files.

conf-enabled: This directory contains symlinks to the files in /etc/apache2/conf-available. If configuration file is symlinked, it will be enabled only after next time apache2 is restarted.

mods-available: this directory contains configuration files to both load modules and configure them. Not all modules will have specific configuration files, however.

mods-enabled: It holds symlinks to the files in /etc/apache2/mods-available. If configuration file is symlinked, it will be enabled only after next time apache2 is restarted.

sites-available: In this directory, we will find the configuration files for Apache virtual host.

sites-enabled: like mods-enabled, sites-enabled contains symlinks to the /etc/apache2/sites-available directory. If configuration file is symlinked, it will be enabled only after next time apache2 is restarted.

httpd.conf: historically the main Apache2 conf file was httpd.conf. In Ubuntu, apache2.conf has all configuration options. It was named after the httpd daemon

```
apt-get update
apt-get -y install apache2 vim curl elinks
sed -i '$a\ServerName 127.0.0.1'  /etc/apache2/apache2.conf

mkdir /var/www/container1/
cd /var/www/container1/
cat > index.html <<EOF
<html><head>  <title> Docker Container1 of Ubuntu! </title>
</head><body>  <p> I'm running this website on Docker Container1 on Ubuntu OS
           </body>
</html>
EOF

service apache2 start

cd /etc/apache2/sites-available/
cp 000-default.conf  container1.conf
sed -i 's=DocumentRoot /var/www/html=DocumentRoot   /var/www/container1/=' container1.conf
sed -i '/container1/ a \\tServerName localhost' container1.conf
a2ensite container1.conf
a2dissite 000-default.conf
service apache2 reload
```

Let’s do the lab

https://github.com/cfitechops/Docker/blob/main/02-docker.md

Create a container with Ubuntu image on the top of Rocky Stream Operating System.

It means we can create different OS containers.

```
docker container run -it –name container1 ubuntu
```

Create Apache Web server

- Download Apache Web server package on the Ubuntu container.
- Configure Apache Web server.
- Create our own Directory.
- Create your HTML page.
- How to stop, start, restart web server.

```
apt-get -y install apache2 vim curl elink
```

```
sed -i '$a\ServerName 127.0.0.1'  /etc/apache2/apache2.conf
mkdir /var/www/container1/
cd /var/www/container1/
cd /etc/apache2/sites-available/
cp 000-default.conf  container1.conf
sed -i 's=DocumentRoot /var/www/html=DocumentRoot
/var/www/container1/=' container1.conf
sed -i '/container1/ a \\tServerName localhost' container1.conf
a2ensite container1.conf
a2dissite 000-default.conf
```

```
service apache2 start
service apache2 reload
service apache2 stop
service apache2 restart
```

```
cat > index.html <<EOF
<html><head>  <title> Docker Container1 of Ubuntu! </title>
</head><body>  <p> I'm running this website on Docker Container1 on Ubuntu
OS
            </body>
</html>
EOF
```

Download the package, such as apache2 vim curl elinks.

```
apt-get -y install apache2 vim curl elink
```

Create an image from the running container.

```
docker commit container1 apache1-ubuntu
```

Create a new container from newly created image.

```
docker run -it  --name container1 apache1-ubuntu
```

Dockerfile

- What is Dockerfile ?

Dockerfile is a script and having instructions in the form of commands.

- Benefits / Uses

It is used to organize and helpful for deployment from start to finish.

- Syntax of Dockerfile
  Dockerfile may be begin with # use to add comments only

```
# Usage: FROM [image_name]
```

2. Command = FROM , It defines the base image to use to start the build process.

```
# Usage: FROM [image_name]
FROM ubuntu
```

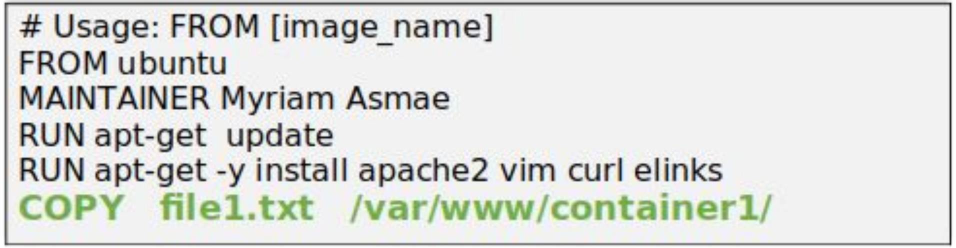
3. Command = MAINTAINER , this non-executing command declares the author, hence setting the author field of the images.

```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
```

4. Command = RUN, it is used to execute the command within the Docker image.
It executed RUN during the build phase.

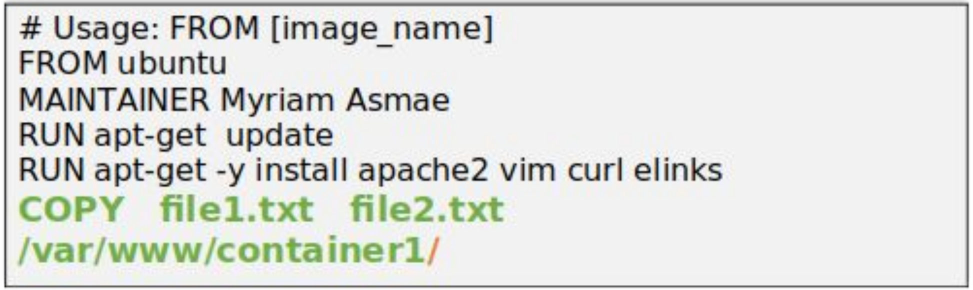
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
```

5. Command = COPY , this command is used to copy the file or directory from Host machine from where we are creating a image.



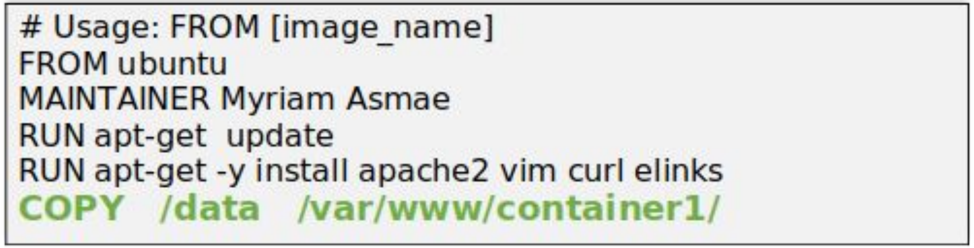
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
```

Copy multiple files



```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   file2.txt
/var/www/container1/
```

Copy directory



```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   /data   /var/www/container1/
```

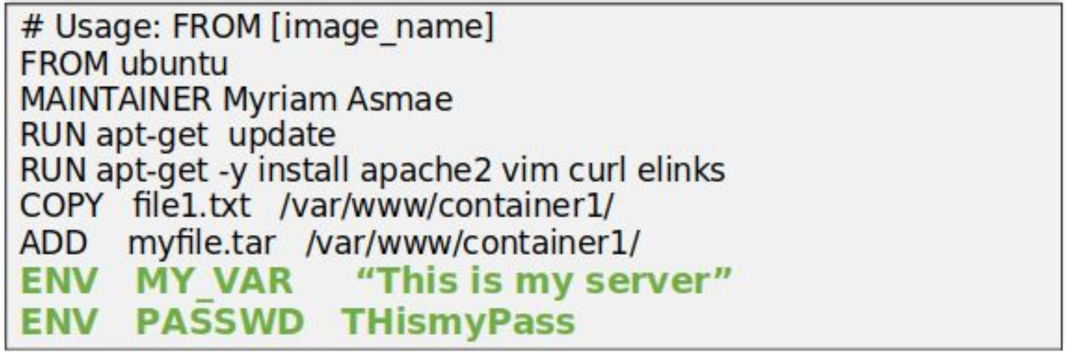
5. Command = ADD, this command is slightly more aggressive than COPY command.
ADD instruction can copy and extract the TAR file from the Docker host to the Docker image.

```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    myfile.tar   /var/www/container1/
```

ADD command can also download the file from HTTP and then copy into the Docker image.

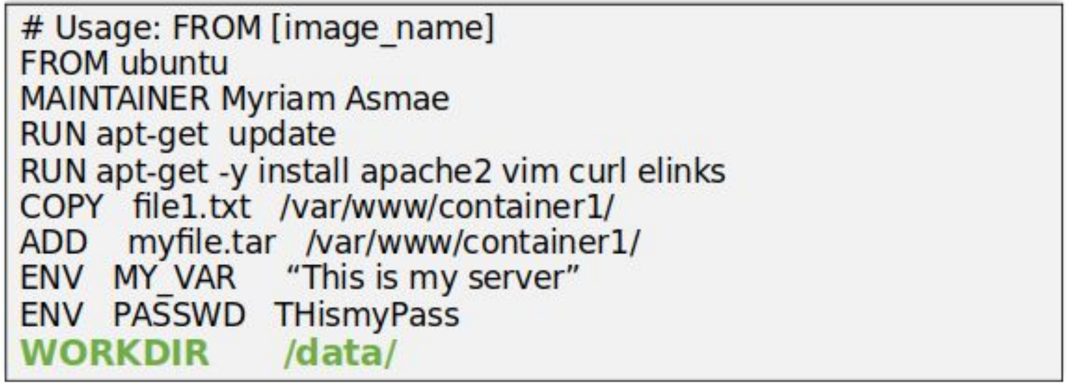
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    https://wallpaper-house.com/wallpaper-id-387707.php    /data
```

6. Command = **ENV,** this command can set an environment variable inside the Docker image.



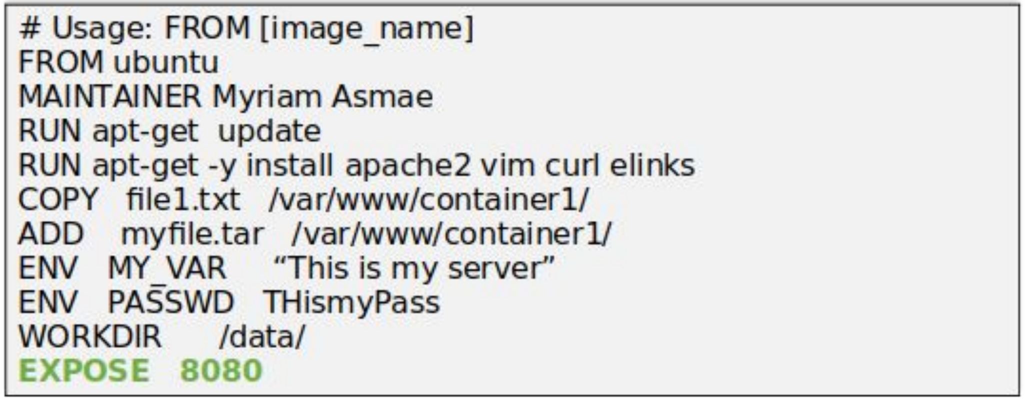
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    myfile.tar   /var/www/container1/
ENV   MY_VAR     "This is my server"
ENV   PASSWD   THismyPass
```

7. Command = **WORKDIR**, it specify the working directory should be inside the Docker image.



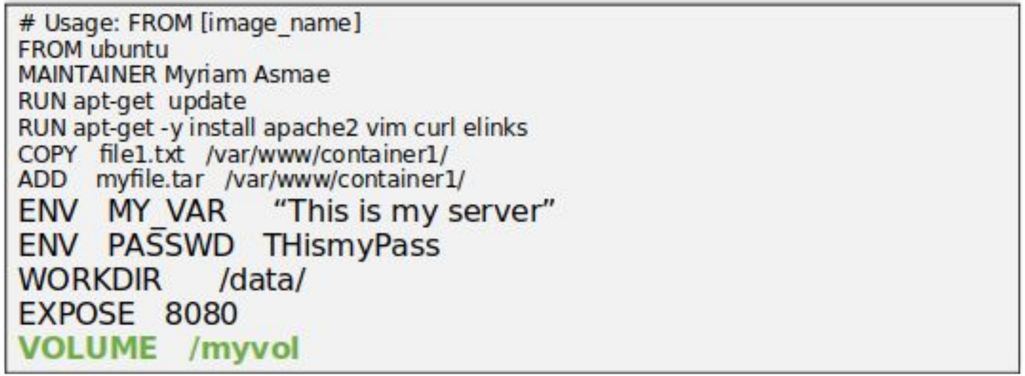
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    myfile.tar   /var/www/container1/
ENV   MY_VAR     "This is my server"
ENV   PASSWD   THismyPass
WORKDIR      /data/
```

8. Command = **EXPOSE,** it is used for port binding. Here, we mention only the host machine port number.



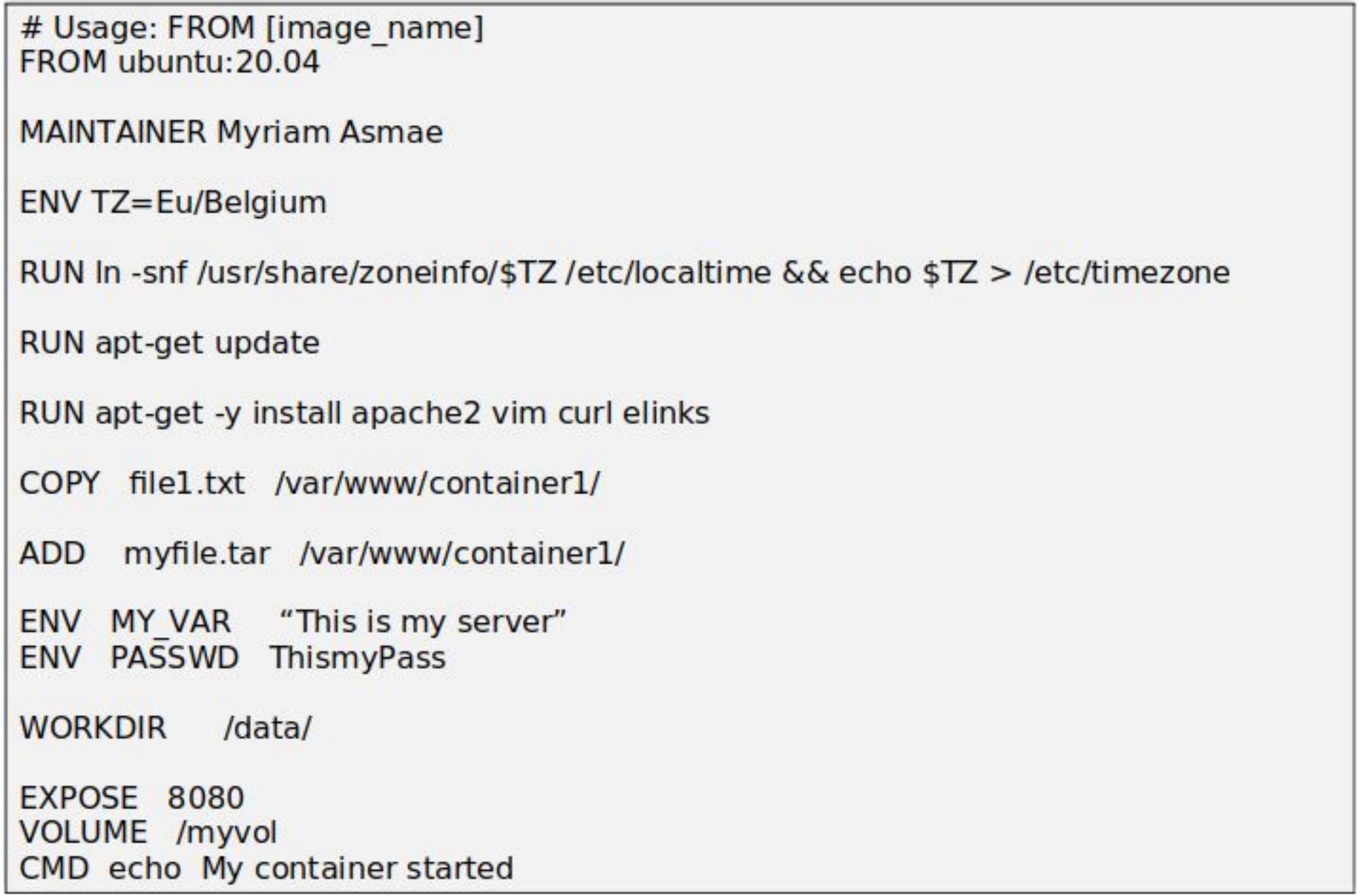
```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    myfile.tar   /var/www/container1/
ENV   MY_VAR     "This is my server"
ENV   PASSWD   THismyPass
WORKDIR      /data/
EXPOSE   8080
```

9. Command = **VOLUME**, it specify the working directory should be inside the Docker image.



```
# Usage: FROM [image_name]
FROM ubuntu
MAINTAINER Myriam Asmae
RUN apt-get  update
RUN apt-get -y install apache2 vim curl elinks
COPY   file1.txt   /var/www/container1/
ADD    myfile.tar   /var/www/container1/
ENV   MY_VAR     "This is my server"
ENV   PASSWD   THismyPass
WORKDIR      /data/
EXPOSE   8080
VOLUME   /myvol
```

10. Command = CMD, it specifies the command line command to execute when a Docker container is started up which is based on the Docker image built from this Dockerfile.



```
# Usage: FROM [image_name]
FROM ubuntu:20.04

MAINTAINER Myriam Asmae

ENV TZ=Eu/Belgium

RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone

RUN apt-get update

RUN apt-get -y install apache2 vim curl elinks

COPY   file1.txt   /var/www/container1/

ADD    myfile.tar   /var/www/container1/

ENV   MY_VAR    “This is my server”
ENV   PASSWD   ThismyPass

WORKDIR     /data/

EXPOSE   8080
VOLUME   /myvol
CMD  echo  My container started
```

DockerFile

```
# Usage: FROM [image_name]
FROM ubuntu:20.04
MAINTAINER Myriam Asmae
ENV TZ=Eu/Belgium
RUN ln -snf /usr/share/zoneinfo/$TZ /etc/localtime && echo $TZ > /etc/timezone
RUN apt-get update
RUN apt-get -y install apache2 vim curl elinks

ADD    myfile.tar   /var/www/container1/
COPY   file1.txt   /var/www/container1/

RUN cd /etc/apache2/sites-available/
RUN cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/container1.conf
RUN sed -i 's=DocumentRoot /var/www/html=DocumentRoot   /var/www/container1/='
/etc/apache2/sites-available/container1.conf
RUN sed -i '/container1/ a \\tServerName localhost' /etc/apache2/sites-available/container1.conf
RUN a2ensite container1
RUN a2dissite 000-default.conf

ENV   MY_VAR     “This is my server”
ENV   PASSWD   ThismyPass

WORKDIR     /data/
EXPOSE   8080
VOLUME   /myvol
ENTRYPOINT service apache2 restart && bash
```

- How to build the image from DockerFile

```
[root@docker vagrant]# docker build -t mywebserver1 .
```

Let’s do the lab

https://github.com/cfitechops/Docker/blob/main/03-docker.md