**What are Dockerfile Directives**

In our previous tutorial, you learned how to build images with [Dockerfile](https://tecadmin.net/tutorial/docker/docker-dockerfile/" \t "_blank). This tutorial will help you to understand the basic Dockerfile directives and there uses.

**FROM**

The from directive is used to set base image for the subsequent instructions. A Dockerfile must have FROM directive with valid image name as the first instruction.

Examples:

FROM ubuntu

FROM tecadmin/ubuntu-ssh:16.04

**LABEL**

Using label you can organize images in a proper way. this is useful to set maintainer address, vender name, version of image, release date etc. The line must be begain with keywork “LABEL”.

LABEL maintainer="rahul@tecadmin.net"

LABEL vendor="TecAdmin"

LABEL com.example.version="0.0.1"

You can add multiple lable in single line with space seprated, or you can define in multiple lines as following.

LABEL maintainer="rahul@tecadmin.net" vendor="TecAdmin" \

com.example.version="0.0.1"

**RUN**

Using RUN directing ,you can run any command to image during build time. For example you can install required packages during the build of image.

RUN apt-get update

RUN apt-get install -y apache2 automake build-essential curl

As a more formatted syntax, you can use as following.

RUN apt-get update && apt-get install -y \

automake \

build-essential \

curl \

**COPY**

The COPY directive used for coping files and directories from host system to the image during build. For example the first commands will copy all the files from hosts html/ directory /var/www/html image directory. Second command will copy all files with extension .conf to /etc/apache2/sites-available/ directory.

COPY html/\* /var/www/html/

COPY \*.conf /etc/apache2/sites-available/

**WORKDIR**

The WORKDIR directive used to sets the working directory for any RUN, CMD, ENTRYPOINT, COPY and ADD commands during build.

WORKDIR /opt

**CMD**

The CMD directive is used to run the service or software contained by your image, along with any arguments during the launching the container. CMD uses follwoing basic syntax

CMD ["executable","param1","param2"]

CMD ["executable","param1","param2"]

For example, to start Apache service during launch of container, Use the following command.

CMD ["apachectl", "-D", "FOREGROUND"]

**EXPOSE**

The EXPOSE directive indicates the ports on which a container will listen for the connections. After that you can bind host system port with container and use them.

EXPOSE 80

EXPOSE 443

**ENV**

The ENV directive is used to set environment variable for specific service of container.

ENV PATH=$PATH:/usr/local/pgsql/bin/ \

PG\_MAJOR=9.6.0

**VOLUME**

The VOLUME directive creates a mount point with the specified name and marks it as holding externally mounted volumes from native host or other containers.

VOLUME ["/data"]

dockerfile

git clone https://github.com/tecrahul/dockerfile

cd dockerfile

docker build -t apacheimage .

docker run -it -p 8080:80 apacheimage

docker run -it -p 8080:80,8081:443 image\_name

docker run --name phpmyadmin -d -e PMA\_HOST=172.21.0.2 -p 8080:80 phpmyadmin/phpmyadmin