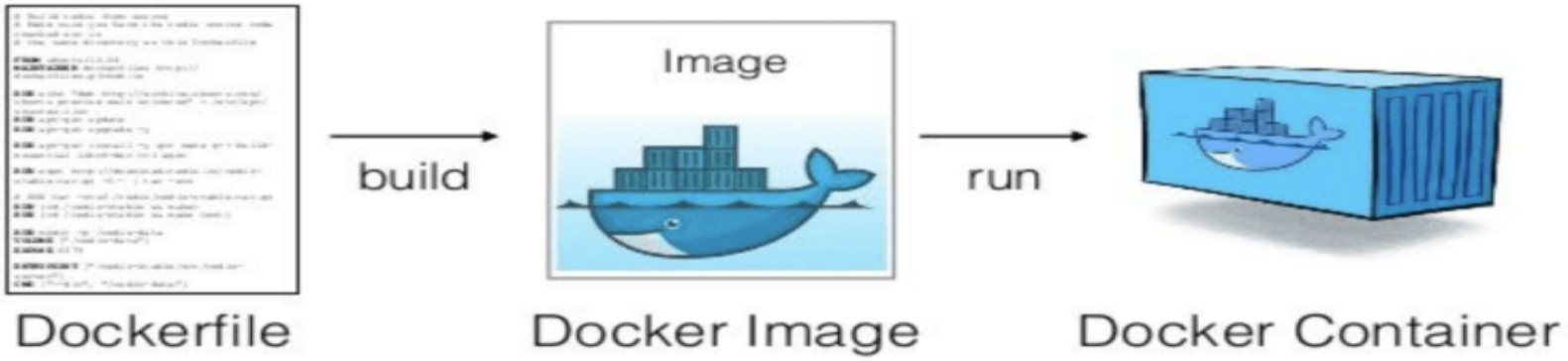
Create a Docker image using Dockerfile



Introduction of Dockerfile

A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

Docker can build images automatically by reading the instructions from a Dockerfile.

Using docker build users can create an automated build that executes several command-line instructions in succession

Docker file

A dockerfile is a text documents that contains all the commands a user could call on the command line to assemble an image.

1) FROM: Define the base image with FROM

2) RUN : Add the lines to install packages

3) ADD : The add keyword used to add files to the conatiner being built

4)CMD or ENTRYPOINT : used to run the commands on the start of the container .(startup command)

5) ENV : used to define environment variable in the container run time

Best practices for writing Dockerfiles--1

Use the official image as a parent image.

FROM ubuntu

Set the working directory.

WORKDIR /usr/src/app

Copy the file from your host to your current location.

COPY index.html . Or COPY content/ .

Run the command inside your image filesystem.

RUN apt install

Inform Docker that the container is listening on the specified port at runtime. **EXPOS**E 8080

Run the specified command within the container.

CMD [“apache2ctl", "start"]

Best practices for writing Dockerfiles---2

#Set one or more individual labels

LABEL com.example.version="0.0.1-beta“

LABEL vendor1="ACME Incorporated“

#To make new software easier to run, you can use ENV to update the PATH environment variable for the software your container installs

#The ENV instruction is also useful for providing required environment variables specific to services you wish to containerize

#ENV can also be used to set commonly used version numbers so that version bumps are easier to maintain

ENV PG_MAJOR 9.3

ENV PG_VERSION 9.3.4

RUN curl -SL http://example.com/postgres-$PG_VERSION.tar.xz | tar -xJC /usr/src/postgress && …

ENV PATH /usr/local/postgres-$PG_MAJOR/bin:$PATH

Best practices for writing Dockerfiles---3

#The VOLUME instruction should be used to expose any database storage area, configuration storage, or files/folders created by your docker container.

#WORKDIR: use absolute paths for your WORKDIR

#ONBUILD: An ONBUILD command executes after the current Dockerfile build completes. ONBUILD executes in any child image derived FROM the current image. Think of the ONBUILD command as an instruction the parent Dockerfile gives to the child Dockerfile

Maintainer: maintainer name, can give email id of the maintainer

Creating docker file for ubuntu to install apache web server

Example -1

```
# mkdir  dockerdata
# cd  dockerdata
# nano  dockerfile
FROM ubuntu
RUN apt-get update
RUN  apt-get  -y install apache2
ADD    .  /var/www/html
ENTRYPOINT apachectl  -D  FOREGROUND
EXPOSE 80
# nano mobile.html
<html>
<h1> Welcome to Mobile Zone </h1>
</html>
# ls
  dockerfile  mobile.html
```

Build Image and push to dockerhub

docker build . -t newimage

or

docker build . -t deepakkumarrts/ubuntu1

docker images

docker run -it -p 83:80 -d deepakkumarrts/ubuntu1

docker ps

container is running. Now we can check by opening it or directly use public IP in web browser

public-ip:83

docker login ----→ type dockerhub user id and password

docker push deepakkumarrts/ubuntu1

Now check in dockerhub.com

We can download it in any other system and run it and check the web page

Creating docker file for centos to install httpd web server

Example -2

```
# mkdir  dockerdata
# cd  dockerdata
# nano  dockerfile
FROM  centos
RUN  yum install  httpd  -y
ADD    .  /var/www/html
ENTRYPOINT /usr/sbin/httpd  -D  FOREGROUND
EXPOSE  80
# nano index.html
<html>
<h1> Welcome to Cloud Computing </h1>
</html>
# ls
  dockerfile  index.html
```

Creating docker file for centos to install httpd web server

Example 3
mkdir dockerdata
cd dockerdata
nano dockerfile
FROM centos
RUN yum install httpd -y
WORKDIR /var/www/html
COPY content/ .
ENTRYPOINT /usr/sbin/httpd -D FOREGROUND
EXPOSE 80
LABEL maintainer=deepak@gmail.com

mkdir content
cd content
Create some html files
cd ..
ls

Build Image and push to dockerhub

docker build . -t newimage1

or

docker build . -t deepakkumarrts/centos1

docker images

docker run -it -p 83:80 -d deepakkumarrts/centos1

docker ps

container is running. Now we can check by opening it or directly use public IP in web browser

public-ip:83

docker login ----→ type dockerhub user id and password

docker push deepakkumarrts/centos1

Now check in dockerhub.com

We can download it in any other system and run it and check the web page

Create Tomcat Server Image

docker pull tomcat:jdk8 or tomcat:8.0-alpine

docker images

nano dockerfile

```
FROM  tomcat:8.0-alpine
LABEL  maintainer=“deepakkumar@gmail.com"
ADD  sampleLogin.war /usr/local/tomcat/webapps/
ADD  sample.war /usr/local/tomcat/webapps/
EXPOSE  8080
CMD  ["catalina.sh", "run"]
```

Download from google – sampleLogin.war and sample.war and keep there

docker build . –t tomcatnew

#docker images

Running Tomcat Server container

docker run --name tomcatserver1 -p 82:8080 -d tomcatnew

Now go to web browser and check

Instance-IP:82

Now do some modification inside the container

docker exec –it tomcatcont-ID bash

pwd

#ls

cd webapps

#ls

cd sample

vi mobile.html or docker.jsp

<html>

<h1> Welcome from my first tomcat file </h1>

</html>

Now check in browser