

Docker File Creation

Lec-27 - Dockerfile Components & diff Command

- Login into AWS account and start your EC2 instance. Access it from putty
- Now we have to create container from our own image.
- Therefore, Create one container first
- `docker run -it --name bhupicontainer ubuntu /bin/bash`
- `cd tmp/`
- Now Create One file inside this tmp directory
- `touch myfile`
- Now if you want to see the difference between the base image & changes on it then
- `docker diff bhupicontainer updateimage`

O/p →

- C /root
- A /root/.bash-history
- C /tmp
- A /tmp/myfile

Now, Create image of this container

- `docker commit newcontainer updateimage`
- `docker images`
- Now Create Container from this image
- `docker run -it --name rajcontainer updateimage /bin/bash`

root@cid##

- ls
- ## `cd tmp/`
- tmp## `ls`

O/p → myfile { you will get all files back }

→ Our container

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
0f05d325d7e2	centos	"/bin/bash"	17 minutes ago	Exited (0) 17 minutes ago		Pandit
a036bf19efcb	72d5c909f166	"/sbin/tini -- /usr/..."	3 days ago	Exited (127) 3 days ago		youthful_newton
92e59d43bc56	ubuntu	"/bin/bash"	3 days ago	Exited (0) 3 days ago		infallible_euclid
21ef6206a3fe	ubuntu	"/bin/bash"	4 days ago	Exited (0) 3 days ago		goofy_meninsky

```
[root@ip-172-31-34-14 ec2-user]#
```

→ First, we enter a **pandit** container and then we added some file to the container and after adding the file to the container we exit a container.

```
[root@ip-172-31-34-14 ec2-user]# docker start Pandit
Pandit
[root@ip-172-31-34-14 ec2-user]# docker attach Pandit
[root@0f05d325d7e2 /]# ls
bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var
[root@0f05d325d7e2 /]# cd tmp
[root@0f05d325d7e2 tmp]# ls
ks-script-4luisyla ks-script-o23i7rc2 ks-script-x6ei4wuu
[root@0f05d325d7e2 tmp]# touch satyam
[root@0f05d325d7e2 tmp]# ls
ks-script-4luisyla ks-script-o23i7rc2 ks-script-x6ei4wuu satyam
[root@0f05d325d7e2 tmp]# exit
exit
[root@ip-172-31-34-14 ec2-user]#
```

→ If we want to see our changes in the container then we use this command to see what kind of changes have been done in the container.

C /root = it means changes in root directory.

A /root/ .bash history = it means changes save in .bash history

C /tmp = it means we change some things in this directory

A /tmp / satyam = it means we add satyam file in the tmp folder.

```
[root@ip-172-31-34-14 ec2-user]# docker diff Pandit
C /root
A /root/.bash_history
C /tmp
A /tmp/satyam
[root@ip-172-31-34-14 ec2-user]#
```

→ Docker images:-

```
[root@ip-172-31-34-14 ec2-user]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
ubuntu	latest	ba6acccedd29	10 days ago	72.8MB
centos	latest	5d0da3dc9764	5 weeks ago	231MB
darwinwilmut/react-docker	latest	6fafcfc54ae3	2 months ago	134MB

→ Convert container to image

- If we want to update our container we use this command
- If I have done some work in my container and then I want to convert the container into an image then we will use this command for that.

```
[root@ip-172-31-34-14 ec2-user]# docker commit Pandit updateimage  
sha256:063962a0f525ed6557fb0b2cae9804275cb5e0f69901cff53a5386cdbd860ee9
```

→ Now, we see our new image is created

```
[root@ip-172-31-34-14 ec2-user]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
updateimage	latest	063962a0f525	5 seconds ago	231MB
ubuntu	latest	ba6acccedd29	10 days ago	72.8MB
centos	latest	5d0da3dc9764	5 weeks ago	231MB
darwinwilmut/react-docker	latest	6fafcfc54ae3	2 months ago	134MB

```
[root@ip-172-31-34-14 ec2-user]#
```

→ Now we want to run our update image

```
[root@ip-172-31-34-14 ec2-user]# docker run -it --name Allahabad updateimage /bin/bash  
[root@45b11a7d39b3 /]# ls
```

→ We use ls command to view our files :-

```
[root@45b11a7d39b3 /]# ls  
bin dev etc home lib lib64 lost+found media mnt opt proc root run sbin srv sys tmp usr var  
[root@45b11a7d39b3 /]# cd tmp/
```

- we create previously in **tmp** folder we create file name is **satyam**, now we see in update image we can see file name satyam is visible.

```
[root@45b11a7d39b3 tmp]# ls  
ks-script-4luisyla ks-script-o23i7rc2 ks-script-x6ei4wuu satyam  
[root@45b11a7d39b3 tmp]#
```

Lec-27 - Dockerfile Components & diff Command

Dockerfile

- Dockerfile is basically a text file. It contains some set of instructions.
- Automation of Docker image creation

Docker Components

FROM → for base image. This command must be on top of the dockerfile.

RUN → To execute commands, it will create a layer in image.

MAINTAINER → Author/Owner/Description

COPY → Copy files from local system (docker vm). We need to provide source, destination.
(We can't download file from internet and only remote repo)

ADD → Similar to COPY but, it provides a feature to download files from internet, also we extract file at docker image side.

EXPOSE → To expose ports such as port 8080 for tomcat, port 80 for nginx etc.

WORKDIR → To set working directory for a container.

CMD → Execute commands but during container creation.

ENTRYPOINT → Similar to CMD, but has higher priority over CMD, first commands will be executed by ENTRYPOINT only.

ENV → Environment Variables

- Whenever we create a Docker file, we will keep its name only Docker file, can not name any other than Docker file
- Whenever we create a docker file, the first word of the docker file is always in capital D.
Ex = Dockerfile
- We will capitalize all the instructions inside the docker file.
Like = FROM, RUN, MAINTAINER, COPY, ADD, EXPOSE, WORKDIR, CMD, ENTRYPOINT, ENV

HOW TO WRITE DOCKER FILE :-

Dockerfile

- 1) → Create a file named Dockerfile
- 2) → Add instructions in Dockerfile
- 3) → Build dockerfile to create image
- 4) → Run image to create container

① **vi Dockerfile**

```
FROM ubuntu
RUN echo "Technical guftgu" > /tmp/testfile
```

To create image out of dockerfile

```
docker build -t myimg .
docker ps -a
docker images
```

Now, Create Container from the above image

```
docker run -it --name mycontainer myimg /bin/bash
```

→ Cat /tmp/testfile

→ Now we create a **Dockerfile**

```
[root@ip-172-31-34-14 ec2-user]# ls
[root@ip-172-31-34-14 ec2-user]# vi Dockerfile
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile
```

→ we write operating system name in **FROM** line.

→ hum jis bhi operating system ki image banana chahate hai us operating system ka naam hum **FROM** mein aage likhenge.

→ **RUN** command mene is liye apne docker file mein likha hai kyu ki hum jo bhi dependencies aur script apne Docker file mein likh rahe un sab ko run karane ka kaam karta hai

→ **RUN echo "hello satyam this is a temporary file " > /tmp/testfile** its means we create a testfile in tmp folder & inside a testfile we write some message .

```
root@ip-172-31-34-14:/home/ec2-user
FROM ubuntu
RUN echo "hello satyam this is a temporary file " > /tmp/testfile_
~
~
```

Convert Dockerfile to docker image

→ We use this command to convert **Dockerfile** to **docker image**. **docker build -t my-image .**

→ When we convert docker file to docker image, we use **.** always in last

```
[root@ip-172-31-34-14 ec2-user]# docker build -t myimage .
Sending build context to Docker daemon 7.68kB
Step 1/2 : FROM ubuntu
--> ba6accdd29
Step 2/2 : RUN echo "hello satyam this is a temporary file " > /tmp/testfile
--> Running in df858934c23d
Removing intermediate container df858934c23d
--> 26c5aed530cd
Successfully built 26c5aed530cd
Successfully tagged myimage:latest
[root@ip-172-31-34-14 ec2-user]#
```

→ Now we see our image is created .

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE
myimage              latest          26c5aed530cd    4 minutes ago   72.8MB
updateimage         latest          063962a0f525    22 hours ago    231MB
ubuntu              latest          ba6accdd29      11 days ago     72.8MB
centos               latest          5d0da3dc9764    5 weeks ago     231MB
darwinwilmut/react-docker latest          6fafcfc54ae3    2 months ago    134MB
[root@ip-172-31-34-14 ec2-user]#
```

→ Our Container

```
[root@ip-172-31-34-14 ec2-user]# ps -a
  PID TTY          TIME CMD
  8610 pts/0    00:00:00 sudo
  8612 pts/0    00:00:00 su
  8613 pts/0    00:00:00 bash
 11727 pts/0    00:00:00 ps
[root@ip-172-31-34-14 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
0f05d325d7e2   centos    "/bin/bash"             23 hours ago   Up 22 hours                   Pandit
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS   NAMES
45b11a7d39b3   updateimage "/bin/bash"             21 hours ago   Exited (0)    19 hours ago   Allahabad
0f05d325d7e2   centos    "/bin/bash"             23 hours ago   Up 22 hours                   Pandit
a036bf19efcb   72d5c909f166 "/sbin/tini -- /usr/..." 4 days ago     Exited (127)   4 days ago     youthful_newton
92e59d43bc56   ubuntu    "/bin/bash"             4 days ago     Exited (0)    4 days ago     infallible_euclid
21ef6206a3fe   ubuntu    "/bin/bash"             4 days ago     Exited (0)    4 days ago     goofy_meninsky
```


→ Now, before running container we create a Dockerfile and inside a Dockerfile we write a testfile now, testfile is visible when container is running.

```
root@96fef6268383:/# ls
bin  dev  home  lib32  libx32  mnt  proc  run  srv  tmp  var
boot  etc  lib  lib64  media  opt  root /sbin  sys  usr
root@96fef6268383:/# cd tmp/
root@96fef6268383:/tmp# ls
testfile
root@96fef6268383:/tmp# cat testfile
hello satyam this is a temporary file
root@96fef6268383:/tmp#
```

→ Now , we edit Dockerfile

vi Dockerfile

```
FROM ubuntu
WORKDIR /tmp
RUN echo "Subscribe Tech" > /tmp/testfile
ENV myname bhupinderrayput
COPY testfile1 /tmp
ADD test.tar.gz /tmp
```



→ Now, we again edit our Dockerfile

```
[root@ip-172-31-34-14 ec2-user]# vi Dockerfile
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile
```

```
[root@ip-172-31-34-14 ec2-user]# touch test test2
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile  test  test2
```

→ WORKDIR /tmp it means our current working directory is tmp folder

→ COPY testfile /tmp it means ki humare local machine mein sein testfile name sein jo file hai usko humare tmp folder mein daal do.

→ ADD test.tar.gz /tmp it means ki humare local machine mein eak test naam ki file padi hai us file ko test file ko pehale tar file banao then uske baad us file ko zip file mein convert kar do.

```
root@ip-172-31-34-14:/home/ec2-user
FROM ubuntu
WORKDIR /tmp
RUN echo "hello satyam this is a temporary file " > /tmp/testfile
ENV myname satyam tripathi
COPY test2 /tmp
ADD test.tar.gz /tmp
~
~
~
~
~
~
```

→ Now we convert our test file into tar file

```
[root@ip-172-31-34-14 ec2-user]# tar -cvf test.tar test
test
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile  test  test2  test.tar
```

→ Now we zipped our test.tar file

```
[root@ip-172-31-34-14 ec2-user]# gzip test.tar
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile  test  test2  test.tar.gz
```

→ Now, we remove our test file, kyuki mere bhai mene apane test file ko test.tar file mein bana diya aur usko zipped bhi kar diya haia toh humein ab zarurat nahi uski.

```
[root@ip-172-31-34-14 ec2-user]# rm -rf test
[root@ip-172-31-34-14 ec2-user]# ls
Dockerfile  test2  test.tar.gz
```


Docker images :-

s

```
[root@ip-172-31-34-14 ec2-user]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
myimage	latest	26c5aed530cd	3 hours ago	72.8MB
updateimage	latest	063962a0f525	25 hours ago	231MB
ubuntu	latest	ba6accdd29	11 days ago	72.8MB
centos	latest	5d0da3dc9764	5 weeks ago	231MB
darwinwilmur/react-docker	latest	6fafcfc54ae3	2 months ago	134MB

- Now, again we convert our docker file to docker image. Humane apne docker file mein jo bhi changes kiya tha humne phir usi file se ek new image create kar diya . hum dekh sakte hai ki docker ek -eak karke humari sab dependences install kar raha jo humne docker file mein de rakha tha apne .

```
[root@ip-172-31-34-14 ec2-user]# docker build -t mynewimage .
Sending build context to Docker daemon 9.216kB
Step 1/6 : FROM ubuntu
--> ba6accdd29
Step 2/6 : WORKDIR /tmp
--> Running in b24881b20eef
Removing intermediate container b24881b20eef
--> 69799eea16b5
Step 3/6 : RUN echo "hello satyam this is a temporary file " > /tmp/testfile
--> Running in d63ec33ef603
Removing intermediate container d63ec33ef603
--> 1378276b4e43
Step 4/6 : ENV myname satyam tripathi
--> Running in faac8f36ee40
Removing intermediate container faac8f36ee40
--> cb66f5174ba8
Step 5/6 : COPY test2 /tmp
--> d3540222d1d2
Step 6/6 : ADD test.tar.gz /tmp
--> 6836c642ede9
Successfully built 6836c642ede9
Successfully tagged mynewimage:latest
[root@ip-172-31-34-14 ec2-user]#
```

→ Image created

```
[root@ip-172-31-34-14 ec2-user]# docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
mynewimage	latest	6836c642ede9	7 minutes ago	72.8MB
myimage	latest	26c5aed530cd	3 hours ago	72.8MB
updateimage	latest	063962a0f525	25 hours ago	231MB
ubuntu	latest	ba6accdd29	11 days ago	72.8MB
centos	latest	5d0da3dc9764	5 weeks ago	231MB
darwinwilmur/react-docker	latest	6fafcfc54ae3	2 months ago	134MB

```
[root@ip-172-31-34-14 ec2-user]#
```

Docker container

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
96fef6268383	myimage	"/bin/bash"	3 hours ago	Exited (0) About an hour ago		testcontainer
45b11a7d39b3	updateimage	"/bin/bash"	24 hours ago	Exited (0) 22 hours ago		Allahabad
0f05d325d7e2	centos	"/bin/bash"	25 hours ago	Up 24 hours		Pandit
a036bf19efcb	72d5c909f166	"/sbin/tini -- /usr/..."	4 days ago	Exited (127) 4 days ago		youthful_newton
92e59d43bc56	ubuntu	"/bin/bash"	5 days ago	Exited (0) 5 days ago		infallible_euclid
21ef6206a3fe	ubuntu	"/bin/bash"	5 days ago	Exited (0) 5 days ago		goofy_meninsky

➔ Mene yaha **mynewcontainer** kein naam sein eak naya container banaya hai aur uske andar mene apni jo **mynewimage** banayi thii us image ko hi use kiya hai

➔ Mein yaha dekh sakta hu ki mene jaise container ko run iya usane turant hi container kein andar tmp folder mein mujhe lekar chala gaya kyu ki mene apaane docker file kein andar apni WORKDIR mein tmp folder ko efine kar rakha hai

```
[root@ip-172-31-34-14 ec2-user]# docker run -it --name mynewcontainer mynewimage /bin/bash
root@ef1cf15debb3:/tmp#
root@ef1cf15debb3:/tmp# ls
test test2 testfile
```

➔ Now, we can see here in container, all details are come.

➔ humane jo bhi apane file mein de rakha thaa vo sab aagaya isme.

```
root@ef1cf15debb3:/tmp# cat testfile
hello satyam this is a temporary file
root@ef1cf15debb3:/tmp#
```

➔ If I want to know our environment we use this command

➔ Humane ENV apane docker file mein de rakha hai.

```
root@ef1cf15debb3:/tmp# echo $myname
satyam tripathi
```