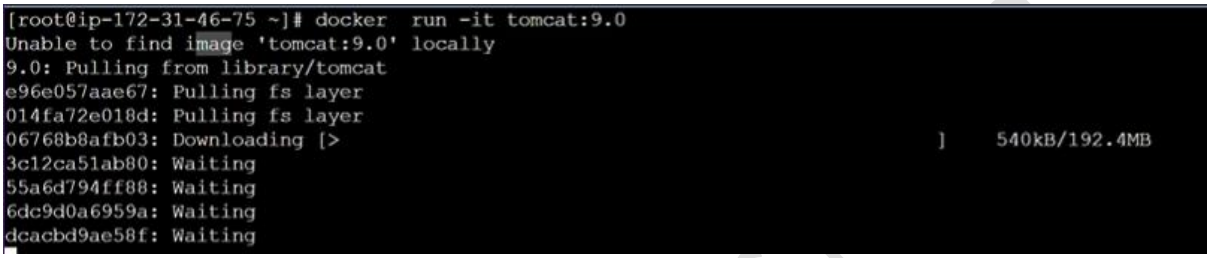
Summary

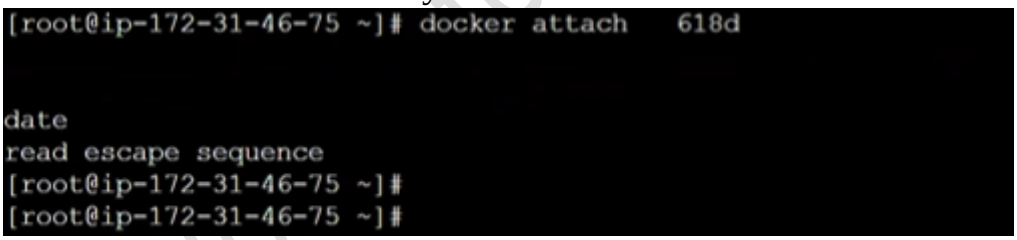
Session No – 12

- If the image is not available & we launch the container from the image the docker run command will first download it & launch the container



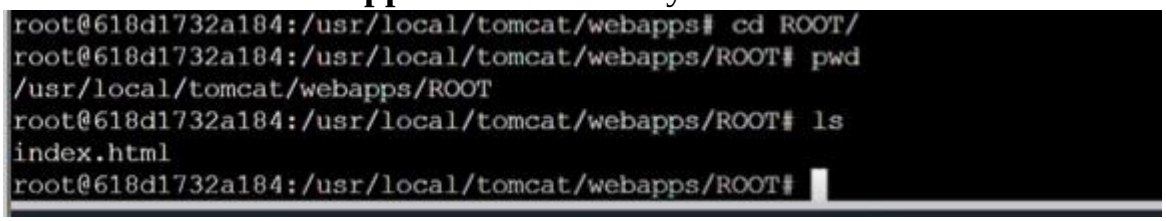
- We can curl the Ip address on port 8080 to see the website running in the container

- Docker attach will work only on that container that has a bash shell



- The only way to go inside the container which does not have a bash shell is with the exec command

- In the tomcat web server, we have to put our webpages in the **/usr/local/tomcat/webapps/ROOT** directory



- Accessing the web page with the curl command

```
[root@ip-172-31-46-75 ~]#
[root@ip-172-31-46-75 ~]# curl  http://172.17.0.2:8080
i m tomcat server for java web app
[root@ip-172-31-46-75 ~]#
[root@ip-172-31-46-75 ~]#
```

- Docker file for tomcat server

```
FROM tomcat:9.0

RUN mkdir  /usr/local/tomcat/webapps/ROOT

COPY index.html /usr/local/tomcat/webapps/ROOT/
~
~
~
~
```

- Building the image

```
[root@ip-172-31-46-75 tdocker]# docker build  -t mytomcat:v1  .
Sending build context to Docker daemon  3.072kB
Step 1/3 : FROM tomcat:9.0
 ---> d0f212a5a5f3
Step 2/3 : RUN mkdir  /usr/local/tomcat/webapps/ROOT
 ---> Running in 6a6fc4d57199
```

- Launching the container & accessing the webpage

```
[root@ip-172-31-46-75 tdocker]# docker run -dit mytomcat:v1
950a6e3a8611a571c4c6d327a240a88c6a62d809d43edbfd99985bc17c9eeb00
[root@ip-172-31-46-75 tdocker]#
[root@ip-172-31-46-75 tdocker]#
[root@ip-172-31-46-75 tdocker]# docker ps
CONTAINER ID   IMAGE         COMMAND             CREATED          STATUS          PORTS      NAMES
950a6e3a8611   mytomcat:v1   "catalina.sh run"   5 seconds ago    Up 4 seconds    8080/tcp   affectionate_leakey
e951e5b58476   centos:7      "/bin/bash"         12 minutes ago   Up 12 minutes              nervous_yonath
618d1732a184   tomcat:9.0    "catalina.sh run"   15 minutes ago   Up 15 minutes   8080/tcp   objective_shamir
[root@ip-172-31-46-75 tdocker]# curl 172.17.0.4:8080
i m tomcat ...
```

- Docker-compose has the capability to run the docker file, build the image and launch the container
- Docker compose file for launching the container

```
version: "3"
services:
  webapp:
    image: mytomcat:v1
~
~
```

- Launching the container with docker-compose

```
[root@ip-172-31-46-75 tdocker]# docker-compose   up  -d
[+] Running 1/1
 ⠿ Container tdocker-webapp-1  Started
[root@ip-172-31-46-75 tdocker]#
[root@ip-172-31-46-75 tdocker]#
[root@ip-172-31-46-75 tdocker]# docker-compose   ps
NAME                COMMAND             SERVICE             STATUS              PORTS
tdocker-webapp-1    "catalina.sh run"   webapp              running             8080/tcp
[root@ip-172-31-46-75 tdocker]#
```

- Docker file integration with Docker-compose
    - In docker-compose, we have a **build** keyword for building the image from the docker file

```
version: "3"
services:
  webapp:
    build: .
~
~
~
~
```

    - **docker-compose build** command is used for building the image in docker-compose

```
[root@ip-172-31-46-75 tdocker]# docker-compose  build
[+] Building 0.8s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                0.1s
 => => transferring dockerfile: 145B                                                                0.0s
 => [internal] load .dockerignore                                                                   0.1s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/tomcat:9.0                                       0.0s
 => [1/3] FROM docker.io/library/tomcat:9.0                                                         0.2s
 => [internal] load build context                                                                   0.1s
 => => transferring context: 57B                                                                    0.0s
 => [2/3] RUN mkdir  /usr/local/tomcat/webapps/ROOT                                                 0.4s
 => [3/3] COPY index.html /usr/local/tomcat/webapps/ROOT/                                           0.1s
 => exporting to image                                                                              0.1s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:87a9d6bb179ba81ca8c39975227c38fc47b8523afccf6743de8d674461eacef6        0.0s
```

    - The image has been built successfully

```
[root@ip-172-31-46-75 tdocker]# docker-compose   images
Container           Repository          Tag                 Image Id            Size
[root@ip-172-31-46-75 tdocker]# docker images
REPOSITORY                  TAG       IMAGE ID       CREATED          SIZE
tdocker-webapp              latest    87a9d6bb179b   36 seconds ago   476MB
mytomcat                    v1        4a65a16b8c67   6 minutes ago    476MB
```

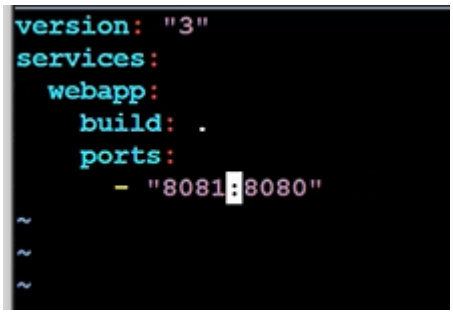
- The image name in docker-compose will be the workspace name then the service name
- Removing the image

```
[root@ip-172-31-46-75 tdocker]# docker rmi tdocker-webapp
Untagged: tdocker-webapp:latest
Deleted: sha256:87a9d6bb179ba81ca8c39975227c38fc47b8523afccf6743de8d674461eacef6
[root@ip-172-31-46-75 tdocker]#
```

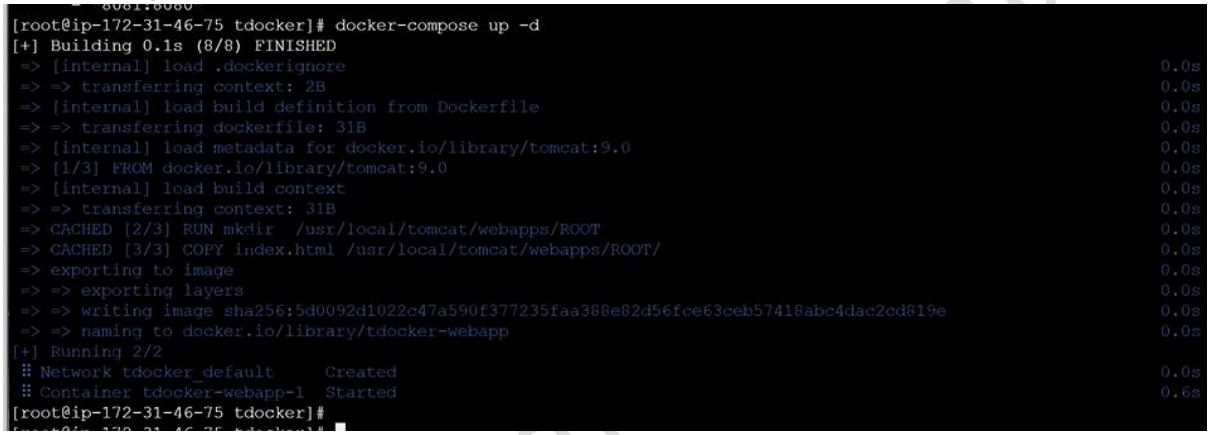
- Now as soon as we run the **docker-compose up** command it will build the image and launch the container

```
[root@ip-172-31-46-75 tdocker]# docker-compose  up -d
[+] Building 0.1s (8/8) FINISHED
 => [internal] load build definition from Dockerfile                                                0.0s
 => => transferring dockerfile: 31B                                                                 0.0s
 => [internal] load .dockerignore                                                                   0.0s
 => => transferring context: 2B                                                                     0.0s
 => [internal] load metadata for docker.io/library/tomcat:9.0                                       0.0s
 => [1/3] FROM docker.io/library/tomcat:9.0                                                         0.0s
 => [internal] load build context                                                                   0.0s
 => => transferring context: 31B                                                                    0.0s
 => CACHED [2/3] RUN mkdir  /usr/local/tomcat/webapps/ROOT                                          0.0s
 => CACHED [3/3] COPY index.html /usr/local/tomcat/webapps/ROOT/                                    0.0s
 => exporting to image                                                                              0.0s
 => => exporting layers                                                                             0.0s
 => => writing image sha256:87a9d6bb179ba81ca8c39975227c38fc47b8523afccf6743de8d674461eacef6        0.0s
 => => naming to docker.io/library/tdocker-webapp                                                   0.0s
[+] Running 2/2
 ⠿ Network tdocker_default     Created                                                              0.0s
 ⠿ Container tdocker-webapp-1  Started                                                              0.9s
[root@ip-172-31-46-75 tdocker]#
```

- If we do patting in docker-compose we don't need to know the IP address of the container we can connect from the base system IP

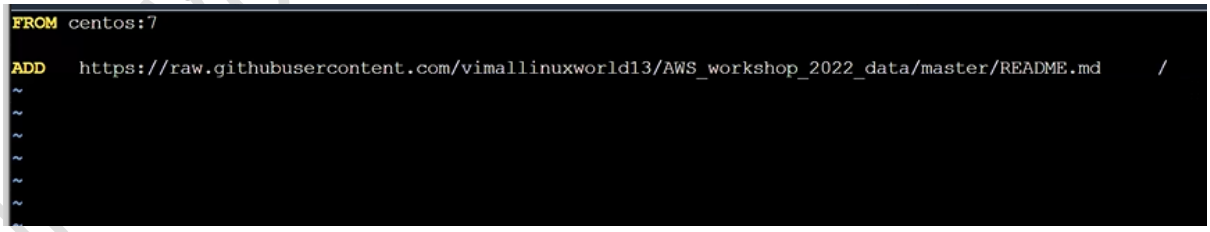


- Launching the container

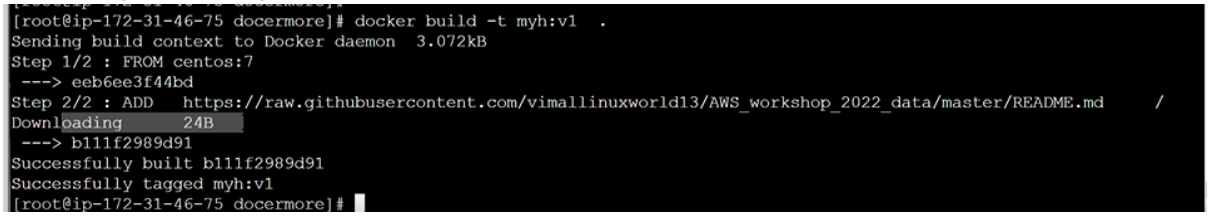


- Accessing the website from the base system IP address

- **ADD** keyword in the docker file is used for copying the files & downloading files from the internet



- The difference between **COPY** & **ADD** keywords is, copy does not have the capability to download files from the internet
- Building the image



- Launching the container & accessing the data

```
[root@ip-172-31-46-75 docermore]# docker run -it myh:v1
[root@534be57a8368 /]# cd /
[root@534be57a8368 /]# ls
README.md          bin  etc   lib    media  opt   root  sbin  sys  usr
anaconda-post.log  dev  home  lib64  mnt    proc  run   srv   tmp  var
[root@534be57a8368 /]# cat README.md
# AWS_workshop_2022_data[root@534be57a8368 /]#
[root@534be57a8368 /]#
```

- Creating achieve file
  - Command:- **tar -c -f (file name with extension .tar) (directory path)**

```
[root@ip-172-31-46-75 ~]# mkdir /website
[root@ip-172-31-46-75 ~]# cd /website
[root@ip-172-31-46-75 website]# ls
[root@ip-172-31-46-75 website]# touch file1.txt
[root@ip-172-31-46-75 website]# touch file2.txt
[root@ip-172-31-46-75 website]# touch pf1.txt
[root@ip-172-31-46-75 website]# touch pf2.txt
[root@ip-172-31-46-75 website]# touch image.png
[root@ip-172-31-46-75 website]# ls
file1.txt  file2.txt  image.png  pf1.txt  pf2.txt
[root@ip-172-31-46-75 website]# pwd
/website
[root@ip-172-31-46-75 website]# tar   -c -f myweb.tar    /website/
tar: Removing leading `/' from member names
tar: /website/myweb.tar: file is the archive; not dumped
[root@ip-172-31-46-75 website]#
[root@ip-172-31-46-75 website]# ls
file1.txt  file2.txt  image.png  myweb.tar  pf1.txt  pf2.txt
[root@ip-172-31-46-75 website]#
```

- Extracting from the tar file
  - Command: - tar -x -f (filename)

```
[root@ip-172-31-46-75 website]# tar -x -f myweb.tar
[root@ip-172-31-46-75 website]# ls
myweb.tar  website
[root@ip-172-31-46-75 website]#
```

- Docker file for copying achieves file

```
FROM centos:7

COPY myweb.tar  /
~
~
~
~
```

- Building the image

```
[root@ip-172-31-46-75 website]# docker build -t mya:v1 .
Sending build context to Docker daemon   12.8kB
Step 1/2 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/2 : COPY myweb.tar  /
 ---> e4880423e4a9
Successfully built e4880423e4a9
```

- Launching the container and extracting the tar file
  - Command:- **tar -x -f (file name)**

```
[root@ip-172-31-46-75 website]# docker run -it mya:v1
[root@c1b63dd45727 /]# cd /
[root@c1b63dd45727 /]# ls
anaconda-post.log  dev  home  lib64  mnt        opt   root  sbin  sys  usr
bin                etc  lib   media  myweb.tar  proc  run   srv   tmp  var
[root@c1b63dd45727 /]# tar -x -f myweb.tar
[root@c1b63dd45727 /]# ld
ld: no input files
[root@c1b63dd45727 /]# ls
anaconda-post.log  dev  home  lib64  mnt        opt   root  sbin  sys  usr  website
bin                etc  lib   media  myweb.tar  proc  run   srv   tmp  var
```

- ADD keyword in the docker file gives us the capability to copy a tar file & extract it at the time of building the image
  - Docker file

```
ADD myweb.tar  /
~
~
```

  - Building the image & launching the container

```
[root@ip-172-31-46-75 website]# docker run -it mya:v1
[root@21fff4363f35 /]# cd /
[root@21fff4363f35 /]# ls
anaconda-post.log  dev  home  lib64  mnt  proc  run   srv   tmp  var
bin                etc  lib   media  opt  root  sbin  sys   usr  website
[root@21fff4363f35 /]# cd website/
[root@21fff4363f35 website]# ls
file1.txt  file2.txt  image.png  pf1.txt  pf2.txt
[root@21fff4363f35 website]#
```

  - Now at the time of building the image, the tar file was extracted
- Docker file for Apache webserver

```
FROM centos:7

RUN  yum install  httpd -y

COPY *.html  /var/www/html/

EXPOSE  80


ENTRYPOINT [ "httpd" ]
CMD [  "-DFOREGROUND" ]
```

- Building the image

```
[root@ip-172-31-46-75 website]# docker build -t myweb:v1  .
Sending build context to Docker daemon  14.85kB
Step 1/6 : FROM centos:7
 ---> eeb6ee3f44bd
Step 2/6 : RUN  yum install  httpd -y
 ---> Using cache
 ---> 82cbcd9e1fbf
Step 3/6 : COPY *.html  /var/www/html/
 ---> Using cache
 ---> 471a73897c6a
Step 4/6 : EXPOSE  80
 ---> Running in e8709f73200c
Removing intermediate container e8709f73200c
 ---> 838e0ae785df
Step 5/6 : ENTRYPOINT [ "httpd" ]
 ---> Running in 8e480bfe9327
Removing intermediate container 8e480bfe9327
 ---> 64413862ec84
Step 6/6 : CMD [  "-DFOREGROUND" ]
```

- Launching the container & exposing the port
  - Command:- **docker run -dit -P (image name)**
  - **-P** for publishing exposed ports to random ports

```
[root@ip-172-31-46-75 website]# docker run  -dit  -P   myweb:v1
0a7e3f9f7d0dc4702c2558ce198e912e21f22eeaaab9a3d8f5931c914f116b73
[root@ip-172-31-46-75 website]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS
    NAMES
0a7e3f9f7d0d   myweb:v1   "httpd -DFOREGROUND"   4 seconds ago   Up 3 seconds   0.0.0.0:49153->80/tcp, :::49153->80/tc
p   clever bardeen
```

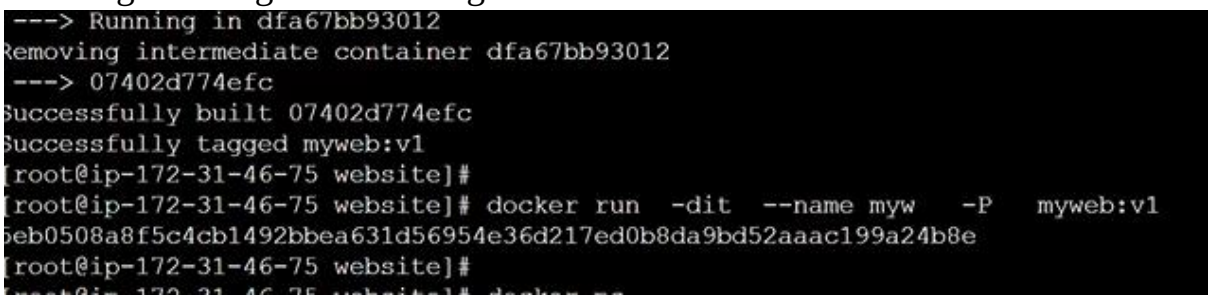
- Accessing the website with the curl command

```
[root@ip-172-31-46-75 website]# curl  http://127.0.0.1:49153
i m index
[root@ip-172-31-46-75 website]#
```

- **ENV** keyword in the docker file is used for setting environmental variable

```
FROM centos:7

RUN  yum install  httpd -y

COPY *.html  /var/www/html/

EXPOSE  80

ENV NAME=vimal

ENTRYPOINT [ "httpd" ]
CMD [  "-DFOREGROUND" ]


-- INSERT --
```

- Building the image & launching the container



- Checking environmental variables in the container