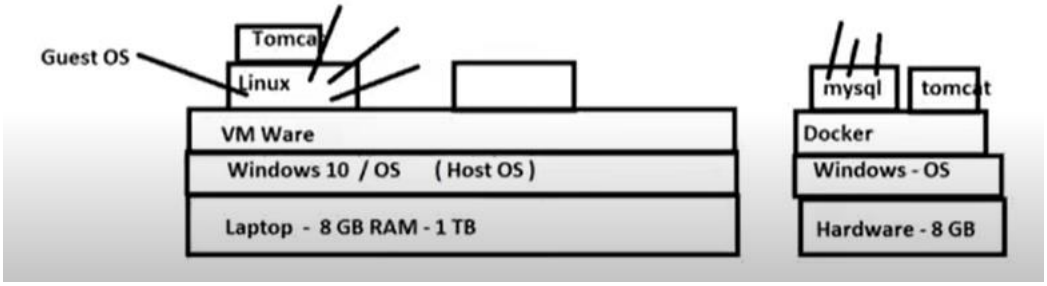
Docker : is a containerization tool



Docker Image: combination of binaries or libraries which are necessary for software application

Docker Container: when image is installed and comes into running condition is called container

Image-->run-->container

Docker Host: Machine on which docker is installed called as docker host

Docker Client: Terminal which is used to run docker commands (gitbash)

Docker commands:

Working on images:

Download image--> docker pull imagename

To see list of docker images -->docker images

To delete a docker image form docker host ->docker rmi imagename/imageid

To upload a docker image into docker hub-->docker push imagename

To build an image from customised container
Docker commit containername newimagename

To create an image from docker file
Docker build -t newimagename

Search for a docker image-->docker search imagename

Delete all images which are not attached to any container
Docker system prune -a

Working on containers:

To see list of all running containers
Docker container ls

To see list of running and stopped containers
Docker ps -a

To start a container
Docker start containername/id

To stop a container
Docker stop contatinername/id

To restart a conatiner
Docker restart containername/id

To delete stopped container
Docker rm conatinername/id

To delete a running container
Docker rm -f conatinername/id

To stop all running containers
Docker stop $(docker ps -aq)

To restart all containers
Docker restart $(docker ps -aq)

to remove all stopped containers
Docker rm $(docker ps -aq)

To remove all containers
docker rm $(docker ps -aq)

To execute any command in a container
Docker ex -it conatainername/id

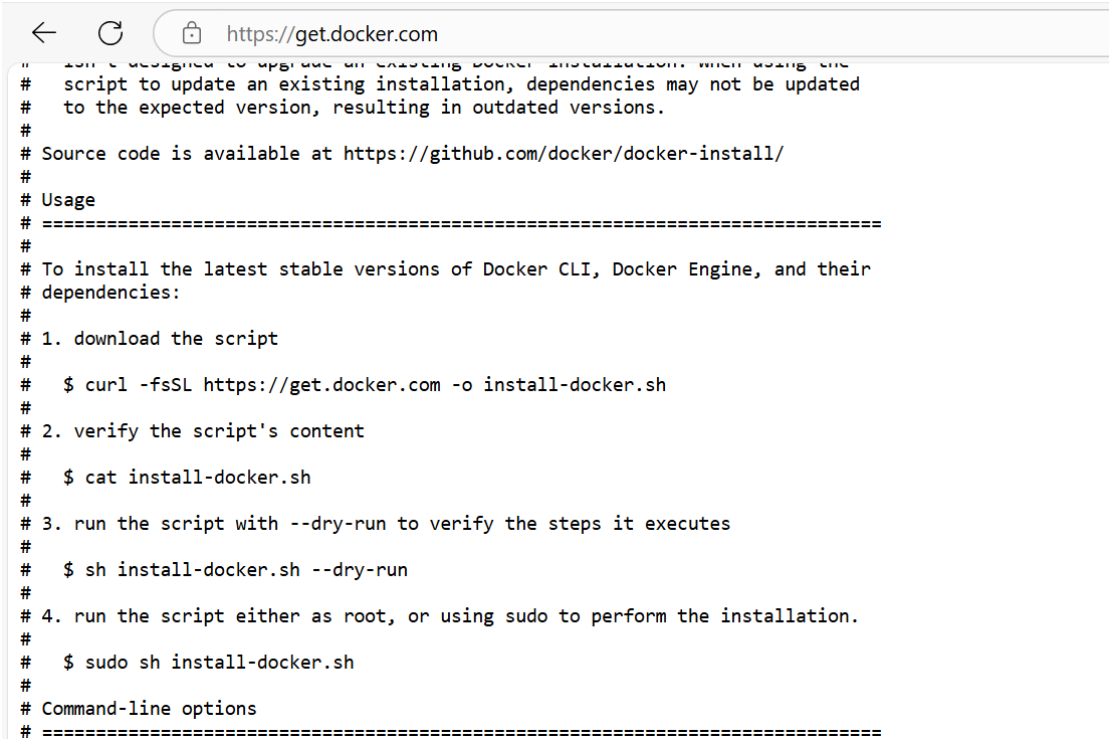
Run command options:

It-->opening interactive terminal
--name--> used for giving name to container
-d -->detached mode run background
p--> used for port mapping
P-->automatic port mapping
--link-->link the multiple containers

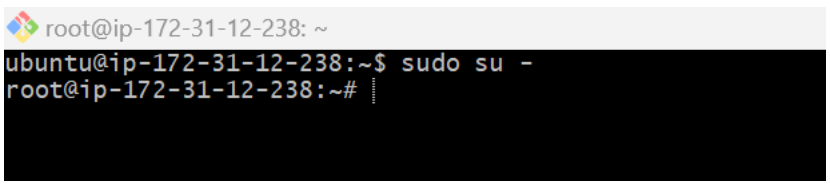
PRACTICAL:

Connect to AWS Create new machine

Open browser (get.docker.com)



Always docker run on root user (#)
$sudo su -



Download and install docker

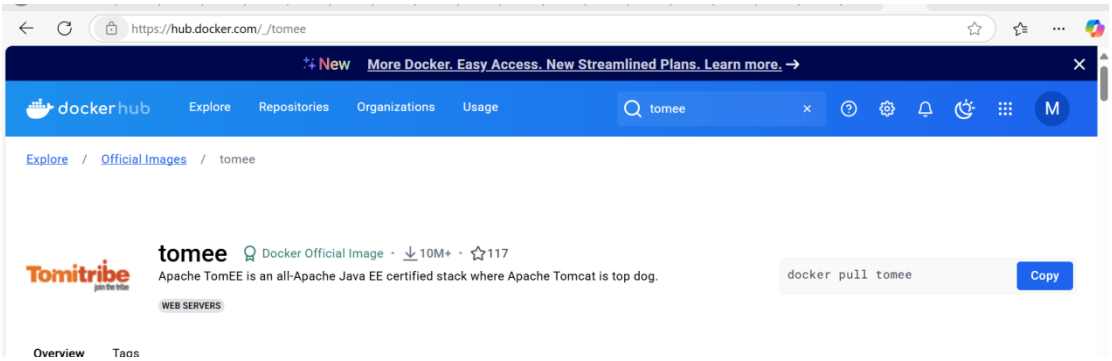
```
curl -fsSL https://get.docker.com -o install-docker.sh

sh install-docker.sh
```

```
ubuntu@ip-172-31-12-238:~$ sudo su -
root@ip-172-31-12-238:~# curl -fsSL https://get.docker.com -o install-docker.sh
root@ip-172-31-12-238:~# sh install-docker.sh
# Executing docker install script, commit: 4c94a56999e10efcf48c5b8e3f6afea464f9108e
+ sh -c apt-get -qq update >/dev/null
```

Install tomcat:

Go to the browser(hub.docker.com) search the image name



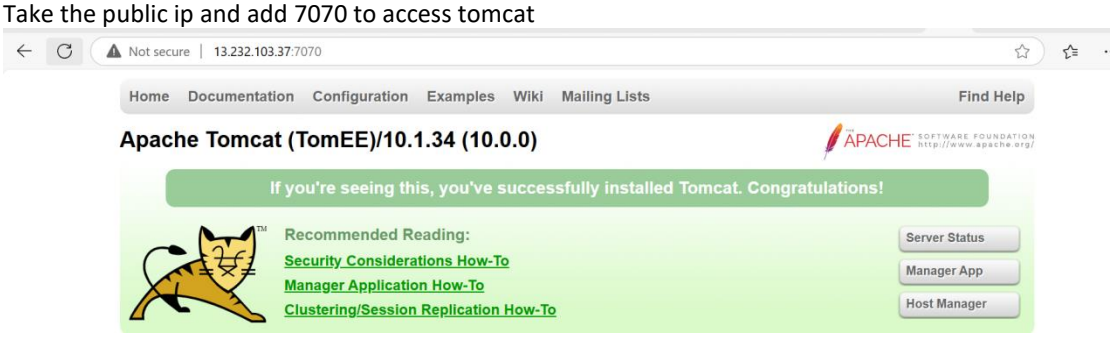

```
root@ip-172-31-12-238:~# docker pull tomee
Using default tag: latest
latest: Pulling from library/tomee
de44b265507a: Pull complete
455e8b7098ce: Pull complete
fde9c83ee8b8: Pull complete
f2fac756920d: Pull complete
3ba4d945c7b8: Pull complete
f82a34ab867d: Pull complete
4f4fb700ef54: Pull complete
15dac9c1f05b: Pull complete
5c0024050fe8: Pull complete
12214257e41d: Pull complete
Digest: sha256:7c809fcd6d99660e79ffdf64c6c6e9741ad897e41f748970baa58635936c00dc
Status: Downloaded newer image for tomee:latest
docker.io/library/tomee:latest
root@ip-172-31-12-238:~#
```

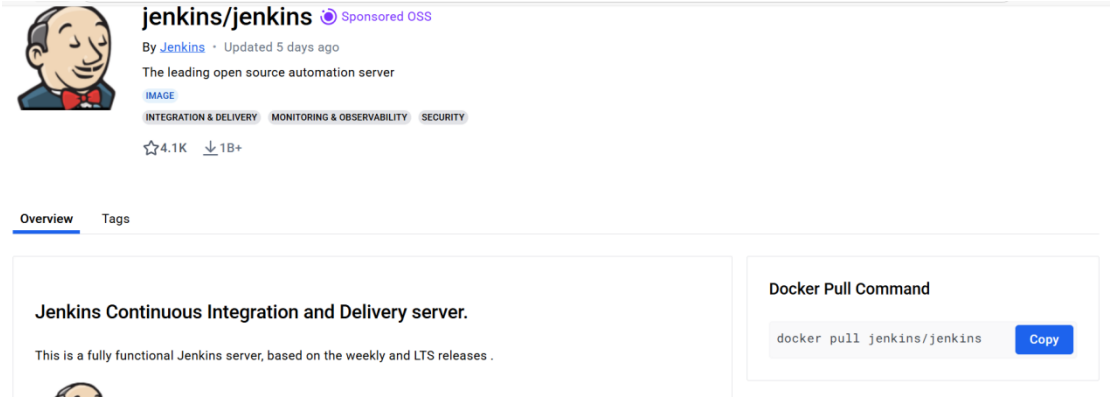
Docker run tomee

```
docker.io/library/tomee:latest
root@ip-172-31-12-238:~# docker run tomee
02-Feb-2025 14:05:18.395 INFO [main] java.lang.reflect.Method.invoke Server version name:   Apache Tomcat (TomEE)/10.1.34 (10.0.0)
02-Feb-2025 14:05:18.397 INFO [main] java.lang.reflect.Method.invoke Server built:          Dec 5 2024 16:01:16 UTC
02-Feb-2025 14:05:18.398 INFO [main] java.lang.reflect.Method.invoke Server version number: 10.1.34.0
02-Feb-2025 14:05:18.398 INFO [main] java.lang.reflect.Method.invoke OS Name:               Linux
02-Feb-2025 14:05:18.398 INFO [main] java.lang.reflect.Method.invoke OS Version:            6.8.0-1021-aws
02-Feb-2025 14:05:18.398 INFO [main] java.lang.reflect.Method.invoke Architecture:         amd64
02-Feb-2025 14:05:18.398 INFO [main] java.lang.reflect.Method.invoke Java Home:            /opt/java/openjdk
02-Feb-2025 14:05:18.399 INFO [main] java.lang.reflect.Method.invoke JVM Version:          21.0.6+7-LTS
02-Feb-2025 14:05:18.399 INFO [main] java.lang.reflect.Method.invoke JVM Vendor:           Eclipse Adoptium
02-Feb-2025 14:05:18.399 INFO [main] java.lang.reflect.Method.invoke CATALINA_BASE:        /usr/local/tomee
02-Feb-2025 14:05:18.399 INFO [main] java.lang.reflect.Method.invoke CATALINA_HOME:        /usr/local/tomee
02-Feb-2025 14:05:18.430 INFO [main] java.lang.reflect.Method.invoke Command line argument: -Djava.util.logging.config.file=/usr/loc
02-Feb-2025 14:05:18.433 INFO [main] java.lang.reflect.Method.invoke Command line argument: -Djava.util.logging.manager=org.apache.j
02-Feb-2025 14:05:18.433 INFO [main] java.lang.reflect.Method.invoke Command line argument: -javaagent:/usr/local/tomee/lib/openejb-
```

Docker run --name mytomcat -p 7070:8080 tomee
Docker run --name mytomcat -p 7070:8080 -d tomee

Take the public ip and add 7070 to access tomcat






```
root@ip-172-31-12-238:~# docker run --name jenkins -p 9090:8080 jenkins/jenkins
Unable to find image 'jenkins/jenkins:latest' locally
latest: Pulling from jenkins/jenkins
fd0410a2d1ae: Extracting [==========================================>        ]  42.76MB/48.48MB
007307ec8e93: Download complete
1c6f0cfeaa6c: Download complete
d659880150a8: Download complete
8733ebe6e173: Download complete
32deaa29f9f6: Download complete
0dd998eb9cf5: Download complete
0110ea549d15: Download complete
d5cefcd910d4: Download complete
cc0276eb9274: Download complete
67c4d9d41045: Download complete
2810efbbbb3e: Download complete
```

Multi container: linking multiple conatiners

--link and docker compose

Link more than two containers use docker compose file

Docker run --name hub -d -p 4444:4444 selenium/hub
Docker run --name chrome -d -p 4445:5900 --link hub:selenium/hub selenium/node-chrome-debug

Docker volumes:

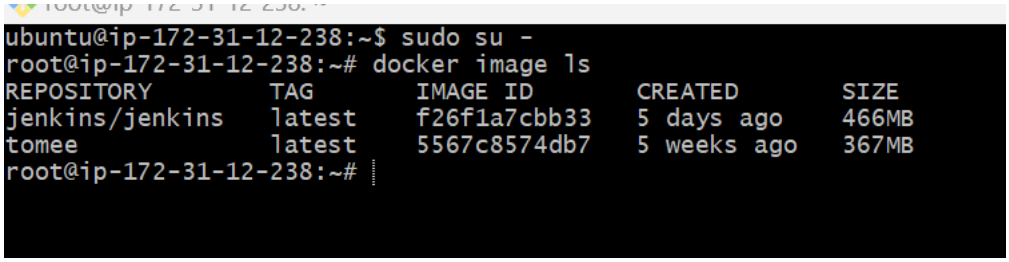
Docker containers are temporary. When a container is deleted all data will be lost. To preserve the data even after deleting the container use docker volumes

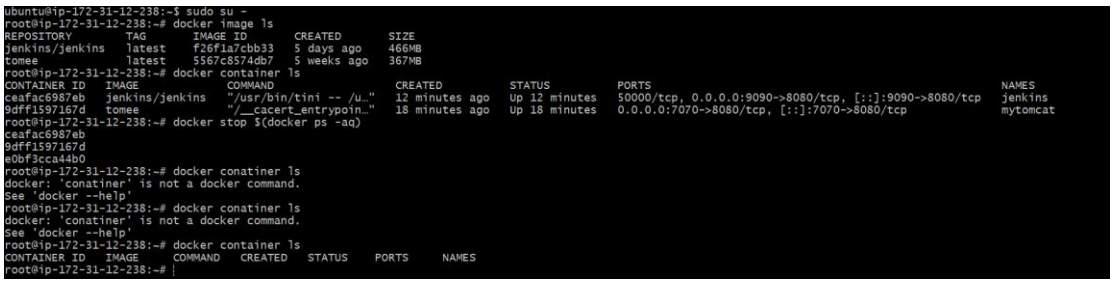
Simple docker volume and docker volume container

Customizing docker images:

Whenever docker container is deleted all the software that we have installed within the container will be deleted.
If we can save the container as an image then we can preserve the software

Using docker commit and using docker file





Docker image ls

Docker run --name c1 -it ubuntu

```
root@ip-172-31-12-238:~# docker image ls
REPOSITORY        TAG       IMAGE ID       CREATED        SIZE
jenkins/jenkins   latest    f26f1a7cbb33   5 days ago     466MB
tomee             latest    5567c8574db7   5 weeks ago    367MB
root@ip-172-31-12-238:~# docker run --name c1 -it ubuntu
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
de44b265507a: Already exists
Digest: sha256:80dd3c3b9c6cecb9f1667e9290b3bc61b78c2678c02cbdae5f0fea92cc6734ab
Status: Downloaded newer image for ubuntu:latest
root@cf966cb18ee0:/# git --version
bash: git: command not found
root@cf966cb18ee0:/#
```

Apt-get update
Apt-get install git
Exit

```
root@cf966cb18ee0:/# git --version
git version 2.43.0
root@cf966cb18ee0:/#
```

Docker commit c1 mycontainer

```
root@ip-172-31-12-238:~# docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-12-238:~# docker commit c1 mycontainer
sha256:d88c66989850a470d2cc518c541a6461c430f29e855bd729672d4ec51403e418
root@ip-172-31-12-238:~# docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                          PORTS     NAMES
cf966cb18ee0   ubuntu            "/bin/bash"              4 minutes ago    Exited (0) About a minute ago             c1
ceafac6987eb   jenkins/jenkins   "/usr/bin/tini -- /u…"   18 minutes ago   Exited (143) 5 minutes ago                jenkins
9dff1597167d   tomee             "/__cacert_entrypoin…"   25 minutes ago   Exited (143) 5 minutes ago                mytomcat
e0bf3cca44b0   tomee             "/__cacert_entrypoin…"   31 minutes ago   Exited (130) 25 minutes ago               busy_hypatia
root@ip-172-31-12-238:~#
```

Docker image ls
Docker run --name c2 -it mycontainer

```
root@ip-172-31-12-238:~# docker container ls
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-12-238:~# docker commit c1 mycontainer
sha256:d88c66989850a470d2cc518c541a6461c430f29e855bd729672d4ec51403e418
root@ip-172-31-12-238:~# docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                          PORTS     NAMES
cf966cb18ee0   ubuntu            "/bin/bash"              4 minutes ago    Exited (0) About a minute ago             c1
ceafac6987eb   jenkins/jenkins   "/usr/bin/tini -- /u…"   18 minutes ago   Exited (143) 5 minutes ago                jenkins
9dff1597167d   tomee             "/__cacert_entrypoin…"   25 minutes ago   Exited (143) 5 minutes ago                mytomcat
e0bf3cca44b0   tomee             "/__cacert_entrypoin…"   31 minutes ago   Exited (130) 25 minutes ago               busy_hypatia
root@ip-172-31-12-238:~# docker image ls
REPOSITORY        TAG       IMAGE ID       CREATED              SIZE
mycontainer       latest    d88c66989850   About a minute ago   214MB
jenkins/jenkins   latest    f26f1a7cbb33   5 days ago           466MB
tomee             latest    5567c8574db7   5 weeks ago          367MB
ubuntu            latest    b1d9df8ab815   2 months ago         78.1MB
root@ip-172-31-12-238:~# docker run --name c2 -it mycontainer
root@256271f3dbad:/# git --version
git version 2.43.0
root@256271f3dbad:/#
```

Using docker file:

```
Using docker file
-------------------
This is a simple text file, which uses  predefinied keywords for creating customized docker images.

Key words used in docker file  ( case sensitive )

1) FROM  --  used to specify the base image from which the docker file has to be created.

2) MAINTAINER -- This represents name of the organization or the
 author who created this docker file.

3) CMD   -- This is used to specify the initial command that should be executed when the container
starts.

4) ENTRYPOINT - used to specify the default process that should be executed when container starts.
It can also be used for accepting arguments from the CMD instruction.

5) RUN  -- Used for running linux commands within the container. It is generally helpful for
installing the software in the container.

6) USER  --      used to specify the default user who should login into the container.
```

```
3) CMD   -- This is used to specify the initial command that should be executed when the container
starts.

4) ENTRYPOINT - used to specify the default process that should be executed when container starts.
It can also be used for accepting arguments from the CMD instruction.

5) RUN  -- Used for running linux commands within the container. It is generally helpful for
installing the software in the container.

6) USER  --      used to specify the default user who should login into the container.

7) WORKDIR --
    Used to specify default working directory in the container

8) COPY  --  Copying the files from the host machine to the container.

9) ADD  -- Used for copying files  from host to container, it can also be used for downloading files
from remote servers.
```

Create dockerfile: vim dockerfile

```
FROM openjdk
WORKDIR /app
COPY . /app
RUN javac sample.java
CMD ["java","sample"]
```

Create sample.java(calculator program)

```
import java.util.Scanner;

class sample{
  public static void main(String[] args) {

    char operator;
    Double number1, number2, result;

    // create an object of Scanner class
    Scanner input = new Scanner(System.in);

    // ask users to enter operator
    System.out.println("Choose an operator: +, -, *, or /");
```

```
    operator = input.next().charAt(0);

    // ask users to enter numbers
    System.out.println("Enter first number");
    number1 = input.nextDouble();

    System.out.println("Enter second number");
    number2 = input.nextDouble();

    switch (operator) {

      // performs addition between numbers
      case '+':
        result = number1 + number2;
        System.out.println(number1 + " + " + number2 + " = " + result);
        break;

      // performs subtraction between numbers
      case '-':
        result = number1 - number2;
        System.out.println(number1 + " - " + number2 + " = " + result);
        break;

      // performs multiplication between numbers
      case '*':
        result = number1 * number2;
        System.out.println(number1 + " * " + number2 + " = " + result);
        break;

      // performs division between numbers
      case '/':
        result = number1 / number2;
        System.out.println(number1 + " / " + number2 + " = " + result);
        break;

      default:
        System.out.println("Invalid operator!");
        break;
    }

    input.close();
  }
}
```

Install java

sudo apt-get update

apt-get install openjdk-21-jdk -y

Compile and run the java program

```
root@ip-172-31-12-238:~# javac sample.java
root@ip-172-31-12-238:~# java sample
Choose an operator: +, -, *, or /
+
Enter first number
10
Enter second number
20
10.0 + 20.0 = 30.0
root@ip-172-31-12-238:~#
```

Create a docker image for the java application

docker build -t maheedhar45/javacalculator .

```
root@ip-172-31-12-238:~# docker build -t maheedhar45/javacalculator .
[+] Building 16.8s (9/9) FINISHED
 => [internal] load build definition from dockerfile
 => => transferring dockerfile: 119B
 => [internal] load metadata for docker.io/library/openjdk:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/4] FROM docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f
 => => resolve docker.io/library/openjdk:latest@sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f
 => => sha256:9b448de897d211c9e0ec635a485650aed6e28d4eca1efbc34940560a480b3f1f 1.04kB / 1.04kB
 => => sha256:fe05457a5e9b9403f8e72eeba507ae80a4237d2d2d3f219fa62ceb128482a9ee 954B / 954B
 => => sha256:71260f256d19f4ae5c762601e5301418d2516ca5911103b1376f063be0b7ba056 4.46kB / 4.46kB
 => => sha256:197c1adcd755131915cd019bdd58658d44445b3638f65449932c18ee39b6047c 44.56MB / 44.56MB
 => => sha256:57b698b7af4b18900b53c768746b1dfb603dfb9aec1eea328fdac86d37001e2a 12.26MB / 12.26MB
 => => sha256:95a27dbe0150755fca4304b4afd0d7d6dd6a40ede6fdb30da8568e9e8cdf23a9 188.74MB / 188.74MB
 => => extracting sha256:197c1adcd755131915cd019bdd58658d44445b3638f65449932c18ee39b6047c
 => => extracting sha256:57b698b7af4b18900b53c768746b1dfb603dfb9aec1eea328fdac86d37001e2a
 => => extracting sha256:95a27dbe0150755fca4304b4afd0d7d6dd6a40ede6fdb30da8568e9e8cdf23a9
 => [internal] load build context
 => => transferring context: 33.36kB
 => [2/4] WORKDIR /app
 => [3/4] COPY . /app
 => [4/4] RUN javac sample.java
 => exporting to image
 => => exporting layers
 => => writing image sha256:4c56e7c240a98376e68ab0d4657190e439631dda16cd5d995f708b51361a29f0
 => => naming to docker.io/maheedhar45/javacalculator
root@ip-172-31-12-238:~#
```

```
root@ip-172-31-12-238:~# docker image ls
REPOSITORY                    TAG       IMAGE ID       CREATED          SIZE
maheedhar45/javacalculator    latest    4c56e7c240a9   53 seconds ago   470MB
mycontainer                   latest    d88c66989850   18 minutes ago   214MB
jenkins/jenkins               latest    f26f1a7cbb33   5 days ago       466MB
tomee                         latest    5567c8574db7   5 weeks ago      367MB
ubuntu                        latest    b1d9df8ab815   2 months ago     78.1MB
root@ip-172-31-12-238:~#
```

Now u push the docker image into docker hub

So create dockerhub account (hub.docker.com)
Login

docker login -u ganga20
Password:

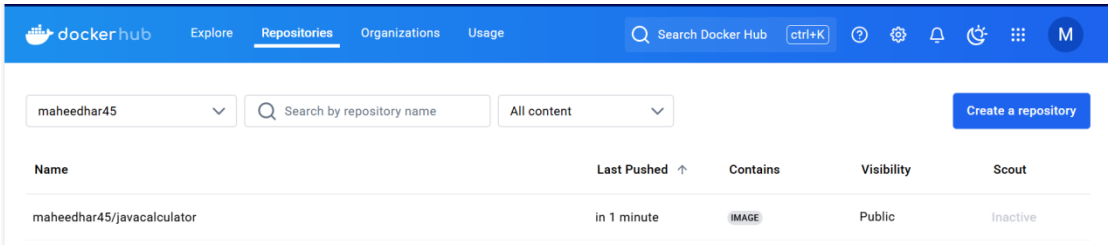
```
root@ip-172-31-12-238:~# docker login -u maheedhar45
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credential-stores

Login Succeeded
root@ip-172-31-12-238:~#
```

docker push ganga20/javacalculator2

```
root@ip-172-31-12-238:~# docker push maheedhar45/javacalculator
Using default tag: latest
The push refers to repository [docker.io/maheedhar45/javacalculator]
160ccfe884b2: Pushed
09f939878b71: Pushed
4fc40ef6f284: Pushed
56285d9a7760: Mounted from library/openjdk
077bff59ce57: Mounted from library/openjdk
9cd9df9ffc97: Mounted from library/openjdk
latest: digest: sha256:c2507d7ca91aa526ef92f9af23afc06ba896f8733269702e0b4af25fc074ca7c size: 1577
root@ip-172-31-12-238:~#
```

Now go and check in the docker hub



Accessing the image

```
docker run --name myjava -it ganga20/javacalculator2
```

```
root@ip-172-31-12-238:~# docker run --name myjava -it maheedhar45/javacalculator
Choose an operator: +, -, *, or /
+
Enter first number
10
Enter second number
20
10.0 + 20.0 = 30.0
root@ip-172-31-12-238:~#
```

Improvise docker image:

Docker file should be edited

```
FROM openjdk:alpine
WORKDIR /app
COPY . /app
RUN javac sample.java
CMD ["java","sample"]
```

```
root@ip-172-31-3-48:~# docker image ls
REPOSITORY              TAG       IMAGE ID       CREATED          SIZE
maheedhar45/calcimp     latest    f8b6ac8723d0   6 seconds ago    103MB
maheedhar45/javacalc    latest    f8c1f4f83c44   9 minutes ago    470MB
root@ip-172-31-3-48:~#
```