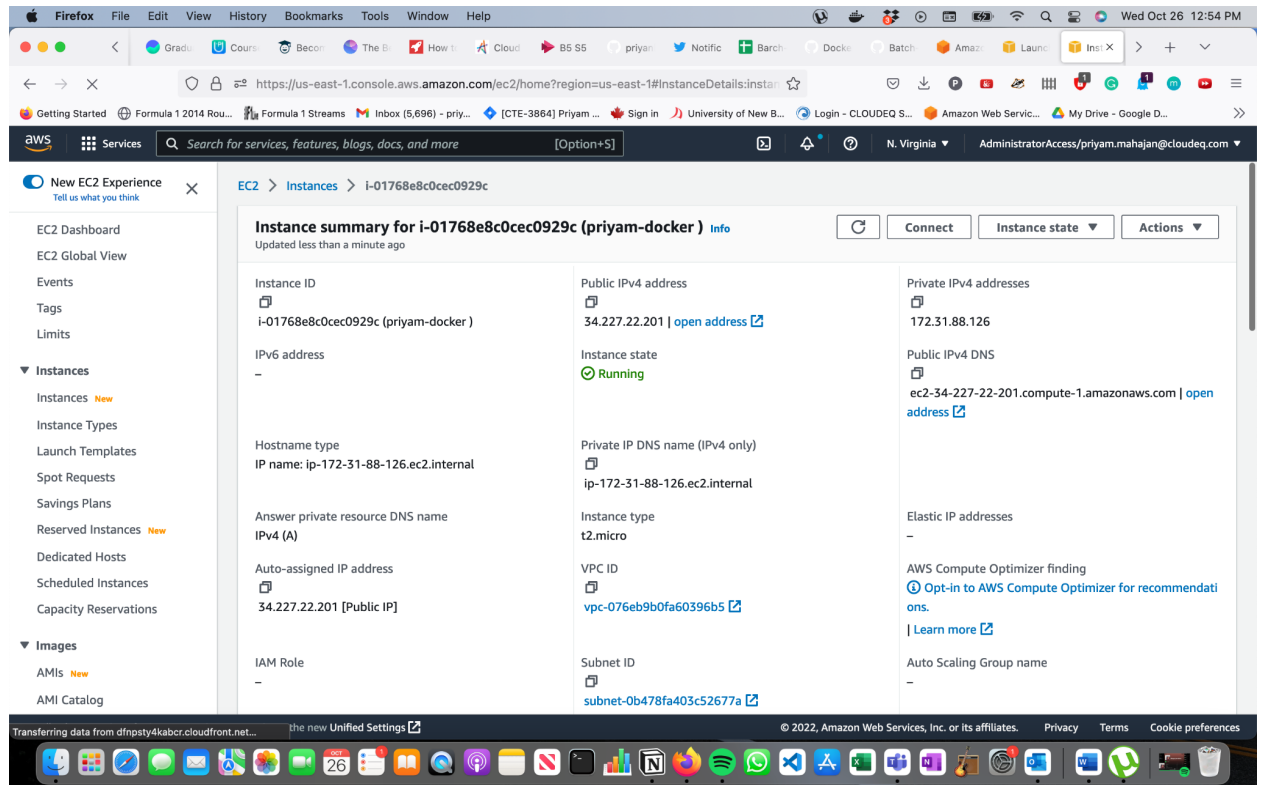


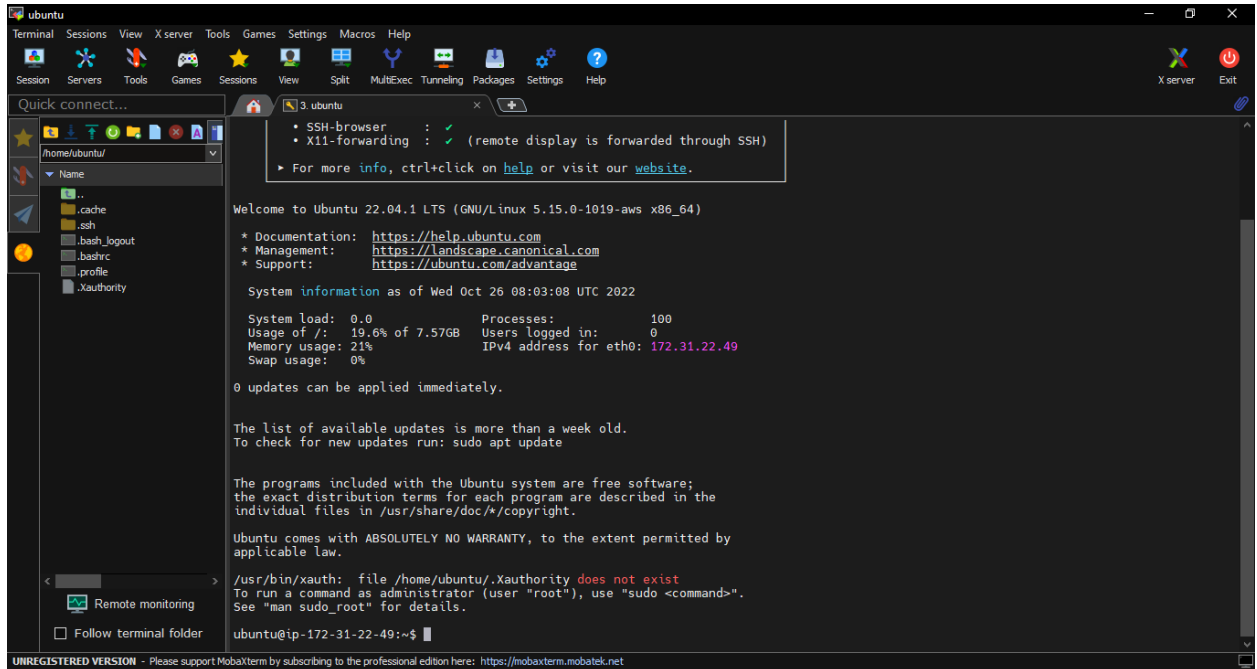
INSTALLING DOCKER USING MOBAXSTREAM

Steps :

- 1) Create an instance in AWS using Ubuntu AMI.



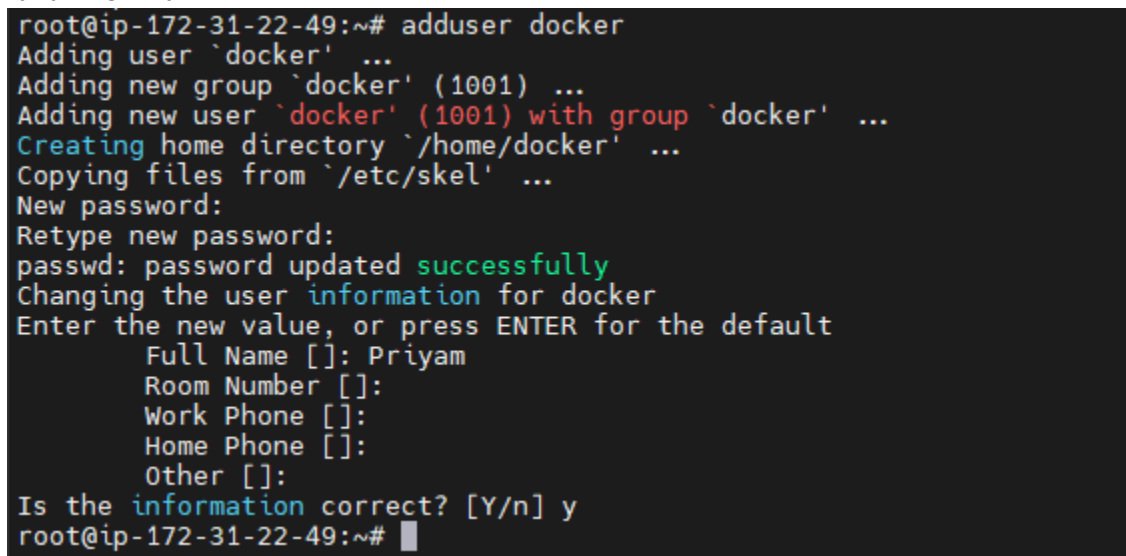
- 2) Connect to instance in MobaXstream using public IP address and setting session name and username = "ubuntu"
- 3)



- 3) Use “**sudo -i**” to go into root user and create root user



- 4) Add user to root using **# adduser docker**, add password and name and confirm details by typing “Y/y”



- 5) Now we give our root permissions so we can give all users the permissions of the root user using “**vi/etc/sudoers**”

```
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root ALL=(ALL:ALL) ALL
docker ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

-- INSERT --
```

45,25

94%

6) Use `#cat /etc/sudoers` to see all permissions.

```
root@ip-172-31-22-49:~# cat /etc/sudoers
#
# This file MUST be edited with the 'visudo' command as root.
#
# Please consider adding local content in /etc/sudoers.d/ instead of
# directly modifying this file.
#
# See the man page for details on how to write a sudoers file.
#
Defaults        env_reset
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults        use_pty

# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
#Defaults:%sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
#Defaults:%sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:%sudo env_keep += "GREP_COLOR"

# While you shouldn't normally run git as root, you need to with etckeeper
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*"

# Per-user preferences; root won't have sensible values for them.
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"
```

7) Move out of root to docker using `#su docker`

```
root@ip-172-31-22-49:~# su docker
docker@ip-172-31-22-49:/root$
```

8) Install docker using repository.

1. We will update the apt package index and install packages to allow apt to use repository over https:
2. Command : **\$sudo apt-get update:**

```
root@ip-172-31-22-49:~# su docker
docker@ip-172-31-22-49:/root$ sudo apt-get update
[sudo] password for docker:
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [114 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [99.8 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [457 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [100 kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [367 kB]
Get:10 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [56.5 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [602 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [76.2 kB]
Get:13 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [2408 B]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [4192 B]
Get:15 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [900 B]
Get:16 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [228 B]
Get:17 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:18 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:19 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:20 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:21 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [693 kB]
Get:22 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [157 kB]
Get:23 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [10.7 kB]
Get:24 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [410 kB]
Get:25 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [63.1 kB]
Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [540 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [743 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [122 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [4404 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [13.7 kB]
Get:31 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [4228 B]
Get:32 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [420 B]
Get:33 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [3008 B]
Get:34 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [1432 B]
Get:35 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [272 B]
```

3. Command : **\$sudo apt-get install**

```
docker@ip-172-31-22-49:/root$ sudo apt-get install
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
0 upgraded, 0 newly installed, 0 to remove and 64 not upgraded.
docker@ip-172-31-22-49:/root$
```

4. Commands :

```
docker@ip-172-31-22-49:/root$ sudo apt-get install \
> ca-certificates \
> curl \
> gnupg \
> lsb-release
```

5. Adding dockers official GPG key :

```
sudo mkdir -p /etc/apt/keyrings
curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg
--dearmor -o /etc/apt/keyrings/docker.gpg
```

```
docker@ip-172-31-22-49:/root$ sudo mkdir -p /etc/apt/keyrings
docker@ip-172-31-22-49:/root$
```

```
Last login: Wed Oct 26 09:13:58 2022 from 49.43.101.178
ubuntu@ip-172-31-22-49:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /etc/apt/keyrings/docker.gpg
ubuntu@ip-172-31-22-49:~$
```

9) Setting up repository using commands :

```
echo \
    "deb [arch=$(dpkg --print-architecture)
signed-by=/etc/apt/keyrings/docker.gpg]
https://download.docker.com/linux/ubuntu \
    $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list >
/dev/null
```

```
ubuntu@ip-172-31-22-49:~$ echo \
> "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu \
$(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
ubuntu@ip-172-31-22-49:~$
```

10) Installing docker engine:

1. Update the apt package index:

```
ubuntu@ip-172-31-22-49:~$ sudo apt-get update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:4 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:5 https://download.docker.com/linux/ubuntu jammy InRelease [48.9 kB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 Packages [9481 B]
Fetched 58.3 kB in 0s (146 kB/s)
Reading package lists ... Done
ubuntu@ip-172-31-22-49:~$
```

Command : **sudo apt-get update**

2. Install Docker Engine, containerd, and Docker Compose.

Command : **sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin**

```

ubuntu@ip-172-31-22-49:~$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  docker-ce-rootless-extras docker-scan-plugin libltdl7 libslirp0 pigz slirp4netns
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras docker-compose-plugin docker-scan-plugin libltdl7 libslirp0 pigz
  slirp4netns
0 upgraded, 10 newly installed, 0 to remove and 64 not upgraded.
Need to get 111 MB of archives.
After this operation, 428 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 pigz amd64 2.6-1 [63.6 kB]
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libltdl7 amd64 2.4.6-15build2 [39.6 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libslirp0 amd64 4.6.1-1build1 [61.5 kB]
Get:4 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 slirp4netns amd64 1.0.1-2 [28.2 kB]
Get:5 https://download.docker.com/linux/ubuntu jammy/stable amd64 containerd.io amd64 1.6.9-1 [27.7 MB]
Get:6 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-cli amd64 5:20.10.21~3-0~ubuntu-jammy [41.5 MB]
Get:7 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce amd64 5:20.10.21~3-0~ubuntu-jammy [20.5 MB]
Get:8 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-ce-rootless-extras amd64 5:20.10.21~3-0~ubuntu-jammy [8389 kB]
Get:9 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-compose-plugin amd64 2.12.2~ubuntu-jammy [9566 kB]
Get:10 https://download.docker.com/linux/ubuntu jammy/stable amd64 docker-scan-plugin amd64 0.21.0~ubuntu-jammy [3622 kB]
Fetched 111 MB in 2s (50.9 MB/s)
Selecting previously unselected package pigz.
Reading database ... 80%

```

RUNNING BASIC COMMANDS IN DOCKER

1. Checking docker version - **\$docker --version**

```

Last login: Wed Oct 26 09:24:28 2022 from 49.43.101.178
ubuntu@ip-172-31-22-49:~$ docker --version
Docker version 20.10.21, build baeda1f
ubuntu@ip-172-31-22-49:~$

```

2. Checking docker version and information about engine etc. - **\$docker version**

```

ubuntu@ip-172-31-22-49:~$ docker version
Client: Docker Engine - Community
 Version: 20.10.21
  API version: 1.41
  Go version: go1.18.7
  Git commit: baeda1f
  Built: Tue Oct 25 18:01:58 2022
  OS/Arch: linux/amd64
  Context: default
  Experimental: true
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2Fdocker.sock/v1.24/version": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-22-49:~$

```

3. Getting help in the command line - **\$docker help**

```

Commands:
attach      Attach local standard input, output, and error streams to a running container
build       Build an image from a Dockerfile
commit      Create a new image from a container's changes
cp          Copy files/folders between a container and the local filesystem
create      Create a new container
diff        Inspect changes to files or directories on a container's filesystem
events      Get real time events from the server
exec        Run a command in a running container
export      Export a container's filesystem as a tar archive
history     Show the history of an image
images      List images
import      Import the contents from a tarball to create a filesystem image
info        Display system-wide information
inspect     Return low-level information on Docker objects
kill        Kill one or more running containers
load        Load an image from a tar archive or STDIN
login       Log in to a Docker registry
logout      Log out from a Docker registry
logs        Fetch the logs of a container
pause       Pause all processes within one or more containers
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image or a repository from a registry
push        Push an image or a repository to a registry
rename      Rename a container
restart     Restart one or more containers
rm          Remove one or more containers
rmi         Remove one or more images
run         Run a command in a new container
save        Save one or more images to a tar archive (streamed to STDOUT by default)
search      Search the Docker Hub for images
start       Start one or more stopped containers
stats       Display a live stream of container(s) resource usage statistics
stop        Stop one or more running containers
tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top         Display the running processes of a container

```

```

exec        Run a command in a running container
export      Export a container's filesystem as a tar archive
history     Show the history of an image
images      List images
import      Import the contents from a tarball to create a filesystem image
info        Display system-wide information
inspect     Return low-level information on Docker objects
kill        Kill one or more running containers
load        Load an image from a tar archive or STDIN
login       Log in to a Docker registry
logout      Log out from a Docker registry
logs        Fetch the logs of a container
pause       Pause all processes within one or more containers
port        List port mappings or a specific mapping for the container
ps          List containers
pull        Pull an image or a repository from a registry
push        Push an image or a repository to a registry
rename      Rename a container
restart     Restart one or more containers
rm          Remove one or more containers
rmi         Remove one or more images
run         Run a command in a new container
save        Save one or more images to a tar archive (streamed to STDOUT by default)
search      Search the Docker Hub for images
start       Start one or more stopped containers
stats       Display a live stream of container(s) resource usage statistics
stop        Stop one or more running containers
tag         Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
top         Display the running processes of a container
unpause     Unpause all processes within one or more containers
update      Update configuration of one or more containers
version     Show the Docker version information
wait        Block until one or more containers stop, then print their exit codes

```

Run 'docker COMMAND --help' for more information on a command.

To get more help with docker, check out our guides at <https://docs.docker.com/go/guides/>
 ubuntu@ip-172-31-22-49:~\$

4. Get information about docker client - **\$ docker info**

```

Server:
 Containers: 0
   Running: 0
   Paused: 0
   Stopped: 0
 Images: 0
 Server Version: 20.10.21
 Storage Driver: overlay2
   Backing Filesystem: extfs
   Supports d_type: true
   Native Overlay Diff: true
   userxattr: false
 Logging Driver: json-file
 Cgroup Driver: systemd
 Cgroup Version: 2
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: io.containerd.runc.v2 io.containerd.runtime.v1.linux runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 1c90a442489720eeca95342e1789ee8a5e1b9536f
 runc version: v1.1.4-0-g5fd4c4d
 init version: de40ad0
 Security Options:
  apparmor
  seccomp
   Profile: default
  cgroupns
 Kernel Version: 5.15.0-1019-aws
 Operating System: Ubuntu 22.04.1 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 1
 Total Memory: 966.2MiB

```

5. List all running containers - **\$ docker ps**

```

root@ip-172-31-22-49:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@ip-172-31-22-49:~#

```

6. List all running and all exited containers - **\$ docker ps -a**

```

root@ip-172-31-22-49:~# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS   NAMES
root@ip-172-31-22-49:~#

```

7. Log in to docker hub - **docker login**

```

root@ip-172-31-22-49:~# docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: priyamcoding
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/#credentials-store

Login Succeeded
root@ip-172-31-22-49:~#

```

8. Adding image to machine - **\$ docker run imagename**


```

root@ip-172-31-22-49:~# docker run mysql
Unable to find image 'mysql:latest' locally
latest: Pulling from library/mysql
50cbc88660a5: Pull complete
92ca853f7184: Pull complete
9a2047696230: Pull complete
fe3fea56f9fb: Pull complete
b058249d3104: Pull complete
9d5014a20163: Pull complete
906aa7388ee2: Pull complete
86b5e2150967: Pull complete
7c6b15dcd4e: Pull complete
21de4337b977: Pull complete
35dab154f2ae: Pull complete
Digest: sha256:06314a7a220f6043436cfd72fd9c7f174fd58ef69fe4b788625fa53be4ab66aa
Status: Downloaded newer image for mysql:latest
2022-10-26 10:07:37+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-26 10:07:37+00:00 [Note] [Entrypoint]: Switching to dedicated user 'mysql'
2022-10-26 10:07:37+00:00 [Note] [Entrypoint]: Entrypoint script for MySQL Server 8.0.31-1.el8 started.
2022-10-26 10:07:38+00:00 [ERROR] [Entrypoint]: Database is uninitialized and password option is not specified
You need to specify one of the following as an environment variable:
- MYSQL_ROOT_PASSWORD
- MYSQL_ALLOW_EMPTY_PASSWORD
- MYSQL_RANDOM_ROOT_PASSWORD
root@ip-172-31-22-49:~#

```

9. Getting images from our machine - \$ **docker images**

```

root@ip-172-31-22-49:~# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
mysql         latest    8fad08b3c84b   4 days ago    535MB
root@ip-172-31-22-49:~#

```

10. Removing images (forcefully) from the machine - \$ **docker rmi -f imagename**

```

root@ip-172-31-22-49:~# docker rmi -f mysql
Untagged: mysql:latest
Untagged: mysql@sha256:06314a7a220f6043436cfd72fd9c7f174fd58ef69fe4b788625fa53be4ab66aa
Deleted: sha256:8fad08b3c84be3e9164f86153224ab616bf71ee2c79677154c2e5cd3179cccfce
root@ip-172-31-22-49:~#

```

11. Logout of docker hub - \$ **docker logout**

```

root@ip-172-31-22-49:~# docker logout
Removing login credentials for https://index.docker.io/v1/
root@ip-172-31-22-49:~#

```