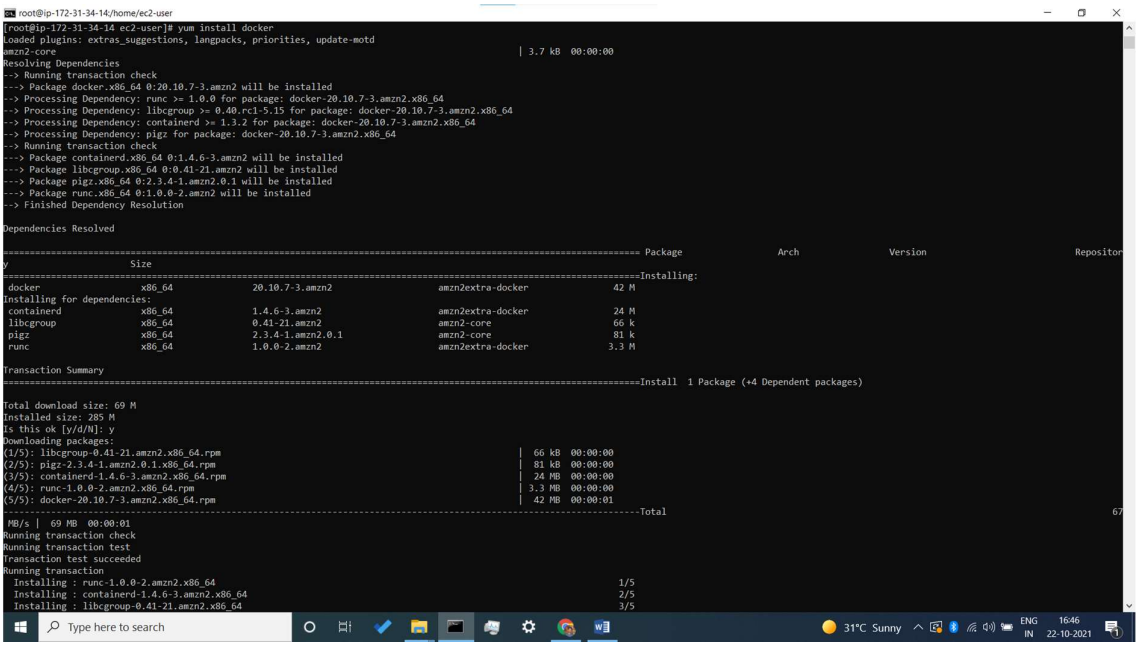
Basics commands of Docker

➔ Install docker in your system

➔ Create Amazon ec2 machine

yum install docker



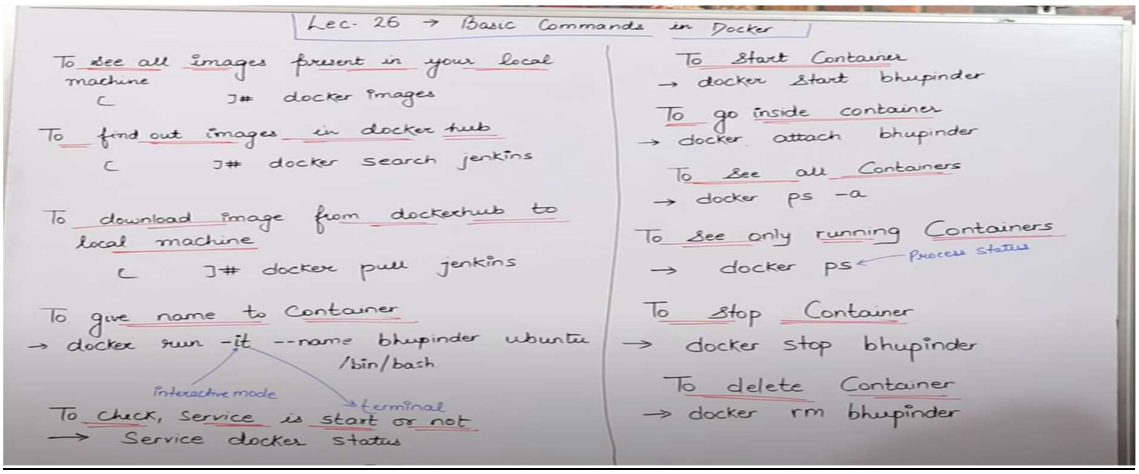
➔ **check Docker path**

```
[root@ip-172-31-34-14 ec2-user]# which docker
/bin/docker
[root@ip-172-31-34-14 ec2-user]#
```

➔ **check Docker --version**

```
[root@ip-172-31-34-14 ec2-user]# docker --version
Docker version 20.10.7, build f0df350
[root@ip-172-31-34-14 ec2-user]#
```

➔ **Basics command of Docker**



Service docker status

➔ We use this command to check our docker service is start or stop, but right now your docker service is stop.

```
[root@ip-172-31-34-14 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: inactive (dead)
     Docs: https://docs.docker.com
```

docker info

➔ You can also use this command to check your docker service is start or stop, but right now your docker service is stop.

➔ You also see in error your docker dameon engine is not connected to your account

```
[root@ip-172-31-34-14 ec2-user]# docker info
Client:
 Context:    default
 Debug Mode: false

Server:
ERROR: Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
errors pretty printing info
[root@ip-172-31-34-14 ec2-user]#
```

Service docker start

➔ Now, your docker service is start

```
[root@ip-172-31-34-14 ec2-user]# service docker start
Redirecting to /bin/systemctl start docker.service
```

Service docker status

➔ We use this command to check our docker service is start or stop, so right now our docker service is start.

```
[root@ip-172-31-34-14 ec2-user]# service docker status
Redirecting to /bin/systemctl status docker.service
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; disabled; vendor preset: disabled)
   Active: active (running) since Fri 2021-10-22 11:38:38 UTC; 22s ago
     Docs: https://docs.docker.com
  Process: 32534 ExecStartPre=/usr/libexec/docker/docker-setup-runtimes.sh (code=exited, status=0/SUCCESS)
  Process: 32524 ExecStartPre=/bin/mkdir -p /run/docker (code=exited, status=0/SUCCESS)
 Main PID: 32539 (dockerd)
    Tasks: 7
   Memory: 37.4M
   CGroup: /system.slice/docker.service
           └─32539 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.719659097Z" level=info msg="scheme \"unix\" not...e=grpc
Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.719957485Z" level=info msg="ccResolverWrapper: ...e=grpc
Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.720239905Z" level=info msg="ClientConn switchin...e=grpc
Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.769692614Z" level=info msg="Loading containers: start."
Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.928335673Z" level=info msg="Default bridge (doc...dress"
Oct 22 11:38:37 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:37.983824675Z" level=info msg="Loading containers: done."
Oct 22 11:38:38 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:38.003059162Z" level=info msg="Docker daemon" comm...0.10.7
Oct 22 11:38:38 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:38.003570080Z" level=info msg="Daemon has complete...ation"
Oct 22 11:38:38 ip-172-31-34-14.ap-south-1.compute.internal systemd[1]: Started Docker Application Container Engine.
Oct 22 11:38:38 ip-172-31-34-14.ap-south-1.compute.internal dockerd[32539]: time="2021-10-22T11:38:38.027305900Z" level=info msg="API listen on /run/....sock"
Hint: Some lines were ellipsized, use -l to show in full.
[root@ip-172-31-34-14 ec2-user]#
```

docker info

➔ Now, you see your docker service is start & you see all the detail about your docker engine .

root@ip-172-31-34-14:/home/ec2-user

```
[root@ip-172-31-34-14 ec2-user]# docker info
Client:
 Context:    default
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 20.10.7
 Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
  userxattr: false
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Cgroup Version: 1
 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: runc io.containerd.runc.v2 io.containerd.runtime.v1.linux
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: d71fcd7d8303cbf684402823e425e9dd2e99285d
 runc version: 84113eef6fc27af1b01b3181f31bbaf708715301
 init version: de40ad0
 Security Options:
  seccomp
   Profile: default
 Kernel Version: 4.14.246-187.474.amzn2.x86_64
 Operating System: Amazon Linux 2
 OSType: linux
 Architecture: x86_64
```

```
Architecture: x86_64
CPUs: 1
Total Memory: 983.3MiB
Name: ip-172-31-34-14.ap-south-1.compute.internal
ID: L37E:EVBI:ZHTW:L6DS:ZLLG:FR3E:WGAF:L7Q3:LPFS:74WN:3E3O:A6KW
Docker Root Dir: /var/lib/docker
Debug Mode: false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
 127.0.0.0/8
Live Restore Enabled: false

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

docker images

➔ We use this command to view our complete docker image in our engine, earlier we didn't create image so now we don't have any image.

```
root@ip-172-31-34-14:/home/ec2-user
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY    TAG         IMAGE ID    CREATED    SIZE
[root@ip-172-31-34-14 ec2-user]#
```

docker container

➔ If we want to see only our running container we use this command .

➔ docker ps

```
[root@ip-172-31-34-14 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
```

➔ If we want to see our whole like running or stop both container we use this command.

➔ docker ps-a

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-34-14 ec2-user]#
```

➔ we run this command :-

[root@ip-172-31-34-14 ec2-user]# docker run -it ubuntu /bin/bash

➔ agar humane ye command chalayi aur isko humare locally machine mein ye image nahi milli toh kaam ye karega ki ye humare docker hub par jayega aur vaha sein image pull karke uthaa layega.

➔ Hum dekh sakte hai usane eak image docker-hub sein download kiya aur apane docker engine m vo image save karke rakh liya .

➔ Aur humane command mein run likha hai matlab ye hai ki image bana kein phir usko run kara dc container ban jaye, aur right now hum jis ubuntu image ki container ko banaya hai uskein andar hi hai, kyu hi hum sab sein neeche vali line mein dekhe toh root kein aage container number laga hai

```
[root@ip-172-31-34-14 ec2-user]# docker run -it ubuntu /bin/bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
7b1a6ab2e44d: Pull complete
Digest: sha256:626ffe58f6e7566e00254b638eb7e0f3b11d4da9675088f4781a50ae288f3322
Status: Downloaded newer image for ubuntu:latest
root@21ef6206a3fe:/#
```

➔ We see container id & in container we use ls command to see all files inside a container .

```
root@21ef6206a3fe:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@21ef6206a3fe:/#
```

➔ If we want to check which operating system we use in our container we use this command.

➔ Now lets see we use Ubuntu operating system in our container.

```
root@21ef6206a3fe:/# cat /etc/os-release
NAME="Ubuntu"
VERSION="20.04.3 LTS (Focal Fossa)"
ID=ubuntu
ID_LIKE=debian
PRETTY_NAME="Ubuntu 20.04.3 LTS"
VERSION_ID="20.04"
HOME_URL="https://www.ubuntu.com/"
SUPPORT_URL="https://help.ubuntu.com/"
BUG_REPORT_URL="https://bugs.launchpad.net/ubuntu/"
PRIVACY_POLICY_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"
VERSION_CODENAME=focal
UBUNTU_CODENAME=focal
root@21ef6206a3fe:/#
```

➔ If we want to stop our container put this command you see your container status is stopped.

```
root@21ef6206a3fe:/# exit
exit
[root@ip-172-31-34-14 ec2-user]#
```

➔ Now we see in docker engine we have 1 image is created

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
ubuntu        latest    ba6acccedd29   6 days ago    72.8MB
[root@ip-172-31-34-14 ec2-user]#
```

➔ Now lets see when again created a new image and again run the image we see we got new container id, hum jab bhi koi container ko stop karate hai aur phir sein usi container ko start karate ehai toh container id sae rehati hai but agar koi naya container banate hai toh uski container id change ho jati hai

```
root@21ef6206a3fe:/# exit
exit
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY   TAG       IMAGE ID       CREATED      SIZE
ubuntu       latest    ba6acccedd29   6 days ago   72.8MB
[root@ip-172-31-34-14 ec2-user]# docker run -it ubuntu /bin/bash
root@92e59d43bc56:/# ls
bin  boot  dev  etc  home  lib  lib32  lib64  libx32  media  mnt  opt  proc  root  run  sbin  srv  sys  tmp  usr  var
root@92e59d43bc56:/#
```

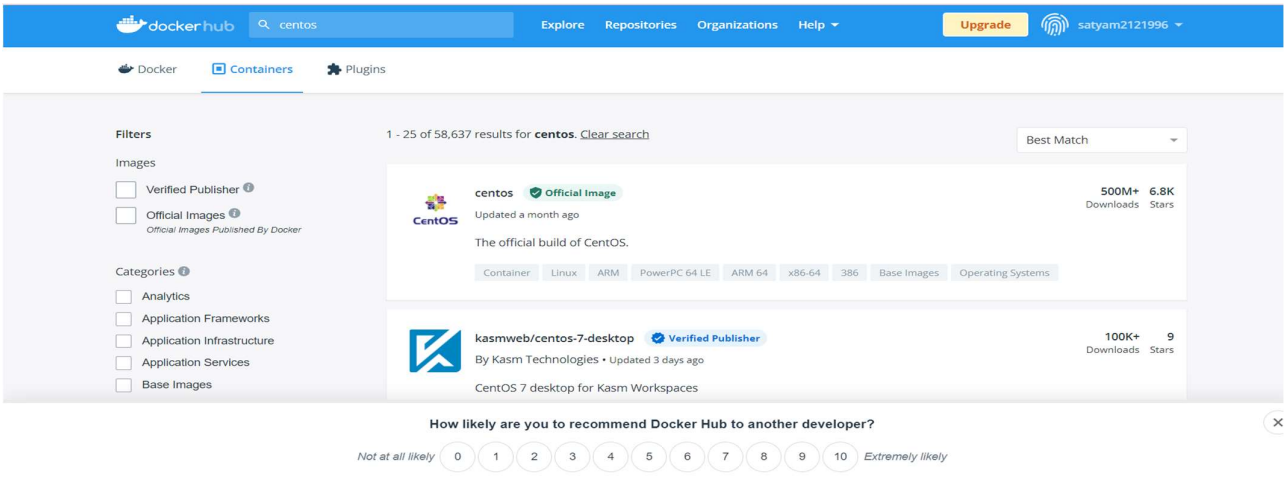
➔ Now we see we have two stopped containers

```
root@92e59d43bc56:/# exit
exit
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED          STATUS                      PORTS     NAMES
92e59d43bc56   ubuntu    "/bin/bash"   24 minutes ago   Exited (0) 8 seconds ago              infallible_euclid
21ef6206a3fe   ubuntu    "/bin/bash"   2 hours ago      Exited (0) 35 minutes ago             goofy_meninsky
[root@ip-172-31-34-14 ec2-user]#
```

➔ Now we see we have only one image but we have 2 containers, it means hum eak image sein kitane bhi naye containers bana sakte hai .

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY    TAG       IMAGE ID       CREATED       SIZE
ubuntu        latest    ba6acccedd29   6 days ago    72.8MB
[root@ip-172-31-34-14 ec2-user]#
```

➔ Agar mujhe docker hub sein koi bhi image chahiye toh uske liye hum docker hub par jakar us operating system ki image ko search karenge jo humko chahihe agar vo image mil jati hai toh hum usko hi **CLI** mein command dekar image ko pull kar lenge apane engine mein kyu ki vo dockerhub par already bani pad hai.



➔

[root@ip-172-31-34-14 ec2-user]# docker run -it centos /bin/bash

➔ After run this command we see one centos image has been created in our docker engine.

```
[root@ip-172-31-34-14 ec2-user]# docker run -it centos /bin/bash
Unable to find image 'centos:latest' locally
latest: Pulling from library/centos
a1d0c7532777: Pull complete
Digest: sha256:a27fd8080b517143cbbbab9dfb7c8571c40d67d534bbdee55bd6c473f432b177
Status: Downloaded newer image for centos:latest
[root@ce7e2cde3189 /]#
```

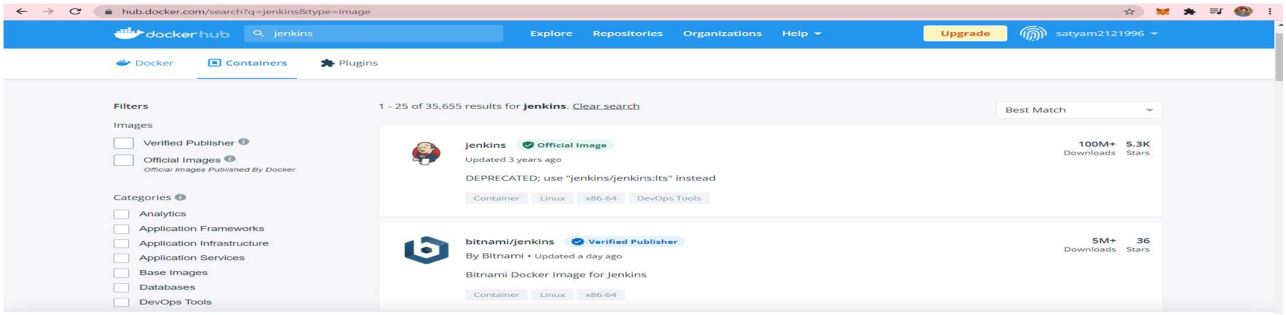
docker images

```
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
ubuntu        latest    ba6acccedd29   7 days ago     72.8MB
centos        latest    5d0da3dc9764   5 weeks ago    231MB
[root@ip-172-31-34-14 ec2-user]#
```

PULL IMAGES IN DOCKERHUB

➔ If we don't want to only pull images in docker-hub & store image in our docker engine, we use this command.

docker pull Jenkins/Jenkins




```
[root@ip-172-31-34-14 ec2-user]# docker pull jenkins/jenkins
Using default tag: latest
latest: Pulling from jenkins/jenkins
bb7d5a84853b: Pull complete
02850768abbf: Pull complete
58af81139dd7: Pull complete
bc03f8ec113a: Pull complete
83dfcca17b11: Pull complete
b0ddf932cd44: Pull complete
5aeb70fbb53c: Pull complete
d2db9f6674bc: Pull complete
121eb53a0f8e: Pull complete
f9cb844c40b8: Pull complete
0210a1b74738: Pull complete
ec63285050c8: Pull complete
573294a5eca2: Pull complete
553604f1c054: Pull complete
71f550d81e2e: Pull complete
a00a3937ff25: Pull complete
6bcf1e5eaa65: Pull complete
Digest: sha256:d7a4e14434f949afce4b87af91ea18691e995b54b1c29a1639cd14f0e1b43213
Status: Downloaded newer image for jenkins/jenkins:latest
docker.io/jenkins/jenkins:latest
[root@ip-172-31-34-14 ec2-user]#
```

docker images

➔ Now, we have 3 docker images.

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED        SIZE
jenkins/jenkins     latest    72d5c909f166   3 days ago     441MB
ubuntu              latest    ba6acccedd29   7 days ago     72.8MB
centos              latest    5d0da3dc9764   5 weeks ago    231MB
```

➔ Docker ps –a

➔ Hum dekhenge humare pass kewal 3 container hai aur humane abhi latest mein Jenkins image banayi but usko humane run nahi kiya isliye vo image humari container mein nahi aayega jaba tak us image ko hum run nahi karenge.

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE     COMMAND       CREATED        STATUS                     PORTS     NAMES
ce7e2cde3189   centos    "/bin/bash"   17 hours ago   Exited (0) 14 hours ago              recursing_maxwell
92e59d43bc56   ubuntu    "/bin/bash"   17 hours ago   Exited (0) 17 hours ago              infallible_euclid
21ef6206a3fe   ubuntu    "/bin/bash"   19 hours ago   Exited (0) 17 hours ago              goofy_meninsky
[root@ip-172-31-34-14 ec2-user]#
```

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
jenkins/jenkins     latest    72d5c909f166   3 days ago    441MB
ubuntu              latest    ba6acccedd29   7 days ago    72.8MB
centos              latest    5d0da3dc9764   5 weeks ago   231MB
[root@ip-172-31-34-14 ec2-user]# docker run -it jenkins/jenkins /bin/bash
jenkins@a036bf19efcb:/$
```

➔ If we have a image in our docker engine and we want to run those image we use this command

```
[root@ip-172-31-34-14 ec2-user]# docker images
REPOSITORY          TAG       IMAGE ID       CREATED       SIZE
jenkins/jenkins     latest    72d5c909f166   3 days ago    441MB
ubuntu              latest    ba6acccedd29   7 days ago    72.8MB
centos              latest    5d0da3dc9764   5 weeks ago   231MB
[root@ip-172-31-34-14 ec2-user]# docker run -it jenkins/jenkins /bin/bash
jenkins@a036bf19efcb:/$
```

➔ **If we want to search docker images in CLI we use this command .**

➔ **Docker search jenkins**

root@ip-172-31-34-14:/home/ec2-user

```
[root@ip-172-31-34-14 ec2-user]# docker search jenkins
NAME                                DESCRIPTION                                     STARS     OFFICIAL   AUTOMATED
jenkins                             DEPRECATED; use "jenkins/jenkins:lts" instead   5342      [OK]
jenkins/jenkins                     The leading open source automation server       2725
jenkinsci/blueocean                 https://jenkins.io/projects/blueocean           647
jenkinsci/jenkins                   Jenkins Continuous Integration and Delivery …   391
jenkins/jnlp-slave                  a Jenkins agent which can connect to Jenkins…   148                  [OK]
jenkinsci/jnlp-slave                A Jenkins slave using JNLP to establish conn…   134                  [OK]
jenkinsci/slave                     Base Jenkins slave docker image                 66                   [OK]
jenkins/slave                       base image for a Jenkins Agent, which includ…   48                   [OK]
jenkinsci/ssh-slave                 A Jenkins SSH Slave docker image                44                   [OK]
jenkins/ssh-slave                   A Jenkins slave using SSH to establish conne…   37                   [OK]
cloudbees/jenkins-enterprise        CloudBees Jenkins Enterprise (Rolling releas…   36                   [OK]
h1kkan/jenkins-docker               Extended Jenkins docker image, bundled wi…      29
xmartlabs/jenkins-android           Jenkins image for Android development.          28                   [OK]
openshift/jenkins-2-centos7         A Centos7 based Jenkins v2.x image for use w…   23
cloudbees/jenkins-operations-center CloudBees Jenkins Operation Center (Rolling …   14                   [OK]
jenkins/ssh-agent                   Docker image for Jenkins agents connected ov…   12
vfarcic/jenkins-swarm-agent         Jenkins agent based on the Swarm plugin         8                    [OK]
openshift/jenkins-slave-base-centos7 A Jenkins slave base image. DEPRECATED: see …  7
trion/jenkins-docker-client         Jenkins CI server with docker client            6                    [OK]
publicisworldwide/jenkins-slave     Jenkins Slave based on Oracle Linux             5                    [OK]
openshift/jenkins-1-centos7         DEPRECATED: A Centos7 based Jenkins v1.x ima…   4
ansibleplaybookbundle/jenkins-apb   An APB which deploys Jenkins CI                 1                    [OK]
amazeeio/jenkins-slave              A jenkins slave that connects to a master vi…   0                    [OK]
jameseckersall/jenkins              docker-jenkins (based on openshift jenkins 2…   0                    [OK]
mashape/jenkins                     Just a jenkins image with the AWS cli added …   0                    [OK]
[root@ip-172-31-34-14 ec2-user]#
```

➔ If we want to give name to our container we use this command :-

[root@ip-172-31-34-14 ec2-user]# docker run **-it --name satyam-Tripathi** centos /bin/bash

```
[root@ip-172-31-34-14 ec2-user]# docker run -it --name satyam-Tripathi centos /bin/bash
[root@ebb90ab9a647 /]# ls
bin  etc   lib    lost+found  mnt  proc  run   srv  tmp  var
dev  home  lib64  media       opt  root  sbin  sys  usr
```

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                      PORTS     NAMES
ebb90ab9a647   centos            "/bin/bash"              24 seconds ago   Exited (0) 14 seconds ago             satyam-Tripathi
a036bf19efcb   jenkins/jenkins   "/sbin/tini -- /usr/…"   3 hours ago      Exited (127) 2 hours ago              youthful_newton
ce7e2cde3189   centos            "/bin/bash"              20 hours ago     Exited (0) 17 hours ago               recursing_maxwell
92e59d43bc56   ubuntu            "/bin/bash"              20 hours ago     Exited (0) 20 hours ago               infallible_euclid
21ef6206a3fe   ubuntu            "/bin/bash"              22 hours ago     Exited (0) 20 hours ago               goofy_meninsky
[root@ip-172-31-34-14 ec2-user]#
```

➔ If we want to run our stop container we use this command

[root@ip-172-31-34-14 ec2-user]# docker start **satyam-Tripathi**

➔ After running this command we see that our one container is running.

➔ Humko jab bhi koi apne stop container ko run karna ho toh us container ka hum naam de dete hai for e.x = **docker start <container name>**

```
[root@ip-172-31-34-14 ec2-user]# docker ps -a
CONTAINER ID   IMAGE             COMMAND                  CREATED          STATUS                      PORTS     NAMES
ebb90ab9a647   centos            "/bin/bash"              24 seconds ago   Exited (0) 14 seconds ago             satyam-Tripathi
a036bf19efcb   jenkins/jenkins   "/sbin/tini -- /usr/…"   3 hours ago      Exited (127) 2 hours ago              youthful_newton
ce7e2cde3189   centos            "/bin/bash"              20 hours ago     Exited (0) 17 hours ago               recursing_maxwell
92e59d43bc56   ubuntu            "/bin/bash"              20 hours ago     Exited (0) 20 hours ago               infallible_euclid
21ef6206a3fe   ubuntu            "/bin/bash"              22 hours ago     Exited (0) 20 hours ago               goofy_meninsky
[root@ip-172-31-34-14 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
[root@ip-172-31-34-14 ec2-user]# docker start satyam-Tripathi
satyam-Tripathi
[root@ip-172-31-34-14 ec2-user]# docker ps
CONTAINER ID   IMAGE     COMMAND       CREATED         STATUS         PORTS     NAMES
ebb90ab9a647   centos    "/bin/bash"   9 minutes ago   Up 8 seconds             satyam-Tripathi
[root@ip-172-31-34-14 ec2-user]#
```

➔ If we want to go inside a container & do some work in our container we use this command.

```
[root@ip-172-31-34-14 ec2-user]# docker attach satyam-Tripathi
[root@ebb90ab9a647 /]#
```

➔ we use this command to start or stop our container

```
[root@ip-172-31-34-14 ec2-user]# docker start satyam-Tripathi
satyam-Tripathi
[root@ip-172-31-34-14 ec2-user]# docker stop satyam-Tripathi
satyam-Tripathi
[root@ip-172-31-34-14 ec2-user]#
```

Delete container

➔ if we want to delete our container we use this

```
[root@ip-172-31-34-14 ec2-user]# docker rm satyam-Tripathi
satyam-Tripathi
```

➔ If we want to **delete our container**, first we check that our container is not in running state if our container is in running state then we can't delete it, so first we stop our running container then we can run a delete command.