DOCKER

DEFINITION:

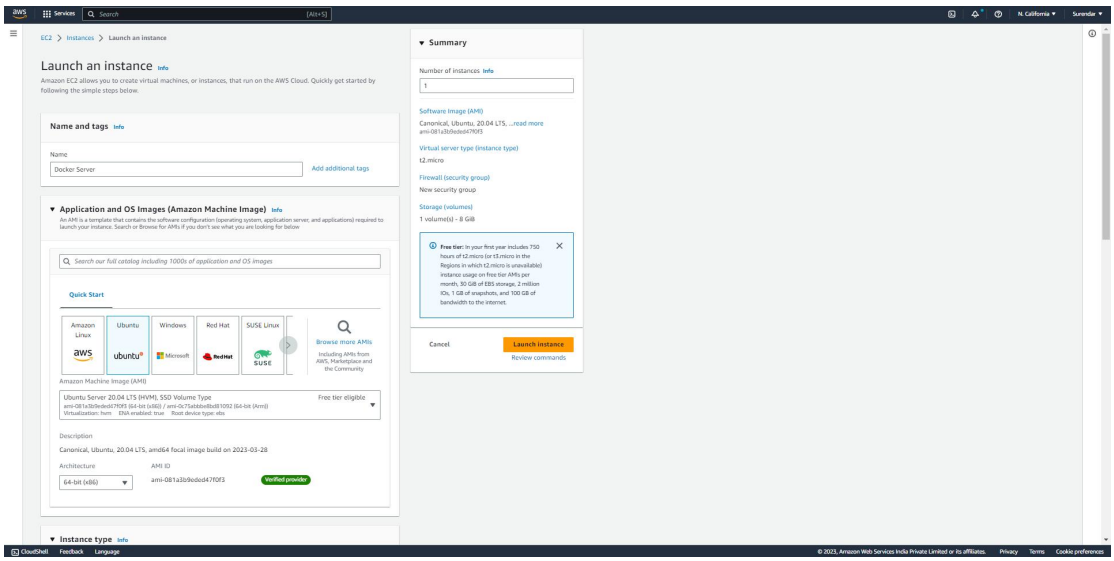
Docker is a tool which is used to automate the deployment of application in lightweight containers so that application can work efficiently in different environment.

Note:

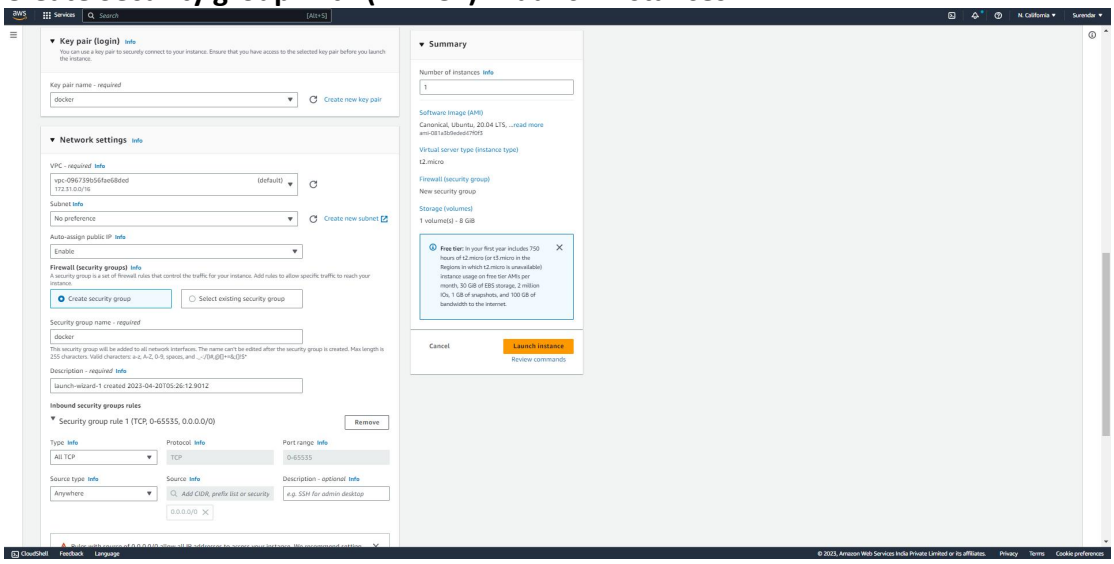
Container is a software package that consists of all the dependencies required to run an applications.

STEPS TO CREATE A CONTAINER AND HOST AN APPLICATION.(using HTTPD Image)

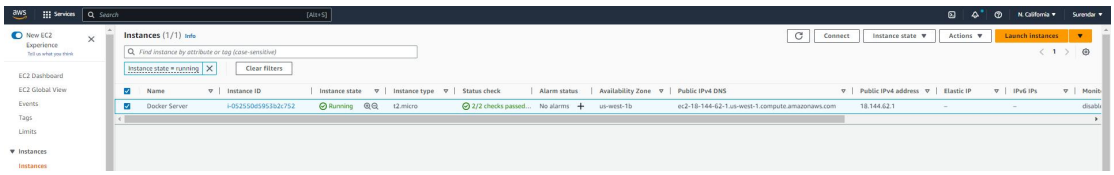
In Amazon EC2 - Launch an Docker Server - Ubuntu 20.04 AMI



Create Security group with (All TCP) - Launch instances.



Connect the Instances.



In Docker Server

$sudo -i

#apt-get update -y

```
ubuntu@ip-172-31-27-105:~$ sudo -i
root@ip-172-31-27-105:~# apt-get update -y
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal InRelease
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:4 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:5 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 Packages [8628 kB]
Get:6 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/universe Translation-en [5124 kB]
Get:7 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/universe amd64 c-n-f Metadata [265 kB]
Get:8 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 Packages [144 kB]
Get:9 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse Translation-en [104 kB]
Get:10 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal/multiverse amd64 c-n-f Metadata [9136 B]
Get:11 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 Packages [2498 kB]
Get:12 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main Translation-en [424 kB]
Get:13 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/main amd64 c-n-f Metadata [16.4 kB]
Get:14 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [1775 kB]
Get:15 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted Translation-en [249 kB]
Get:16 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 c-n-f Metadata [636 B]
Get:17 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 Packages [1051 kB]
Get:18 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe Translation-en [248 kB]
Get:19 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/universe amd64 c-n-f Metadata [24.2 kB]
Get:20 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [25.2 kB]
Get:21 http://us-west-1.ec2.archive.ubuntu.com/ubuntu focal-updates/multiverse Translation-en [7408 B]
```

Install Docker Package

#apt-get install docker.io -y

```
root@ip-172-31-27-105:~# apt-get install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base libidn11 pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools cgroupfs-mount | cgroup-lite debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils containerd dns-root-data dnsmasq-base docker.io libidn11 pigz runc ubuntu-fan
0 upgraded, 9 newly installed, 0 to remove and 27 not upgraded.
Need to get 66.1 MB of archives.
After this operation, 293 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Verify docker package installed properly using version check command.

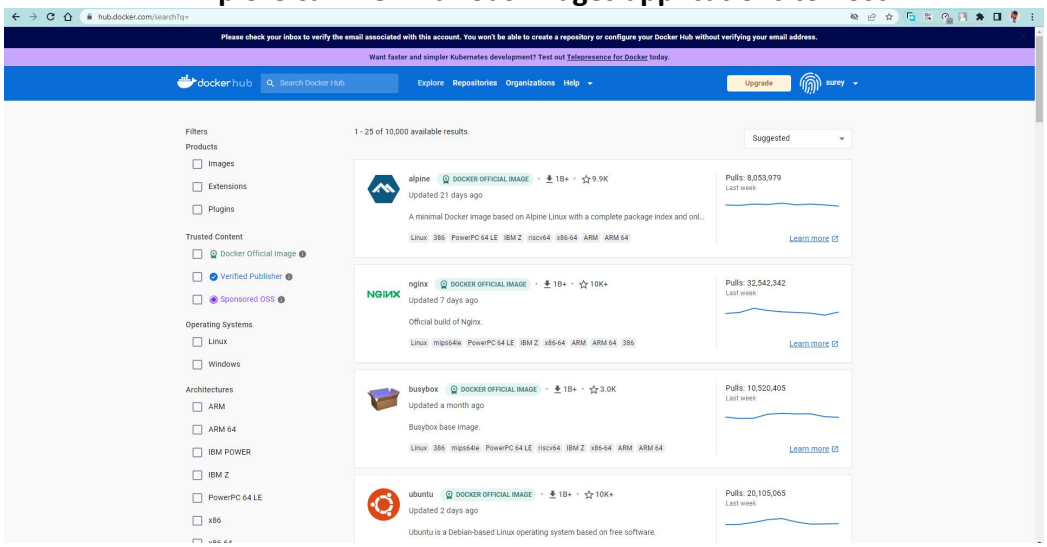
#docker --version

```
root@ip-172-31-27-105:~# docker --version
Docker version 20.10.21, build 20.10.21-0ubuntu1~20.04.1
root@ip-172-31-27-105:~#
```

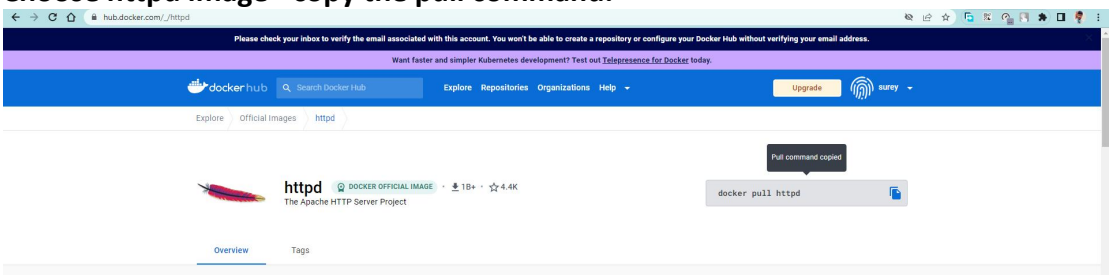
To get an sample images to host an application

Go to (hub.docker.com) page - create an account.

In Explore can view various images applications to host.



Choose httpd image - copy the pull command.



In docker server (paste pull command)

#docker pull httpd

```
root@ip-172-31-27-105:~#
root@ip-172-31-27-105:~# docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
26c5c85e47da: Pull complete
2d29d3837df5: Pull complete
2483414a5e59: Pull complete
e78016c4ba87: Pull complete
757908175415: Pull complete
Digest: sha256:a182ef2350699f04b8f8e736747104eb273e255e818cd55b6d7aa50a1490ed0c
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
root@ip-172-31-27-105:~#
```

To list the docker images

#docker images

```
root@ip-172-31-27-105:~# docker images
REPOSITORY   TAG       IMAGE ID       CREATED       SIZE
httpd        latest    4b7fc736cb48   8 days ago    145MB
root@ip-172-31-27-105:~#
```

To create an container using httpd image

#docker run -itd --name “Container1” -p “2729:80” httpd

i - interactive

t - terminal

d - detached

```
root@ip-172-31-27-105:~#
root@ip-172-31-27-105:~# docker run -itd --name "Container1" -p "2729:80" httpd
624ee5a1de058718f13f45593b77419d653e615dec6e991b4ae36329f8920a61
root@ip-172-31-27-105:~# docker container ls
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS                                   NAMES
624ee5a1de05   httpd     "httpd-foreground"   13 seconds ago   Up 13 seconds   0.0.0.0:2729->80/tcp, :::2729->80/tcp   Container1
root@ip-172-31-27-105:~#
```

To list the running conatiners

#docker ps

#docker container ls

To list all the container which is stop and running state

#docker ps -a

To Enter into the container for changing any content in httpd index.html file.

#docker exec -it <container ID> /bin/bash

#cd htdocs

```
root@ip-172-31-27-105:~# docker exec -it 624ee5a1de05 /bin/bash
root@624ee5a1de05:/usr/local/apache2# ls
bin  build  cgi-bin  conf  error  htdocs  icons  include  logs  modules
root@624ee5a1de05:/usr/local/apache2# cd htdocs
root@624ee5a1de05:/usr/local/apache2/htdocs# ls
index.html
root@624ee5a1de05:/usr/local/apache2/htdocs#
```

Vim editor is not installed inside the container so we need to install vim editor to edit the file.Before that we need to update the apt-get.

#apt-get update -y

```
root@624ee5a1de05:/usr/local/apache2/htdocs# apt-get update -y
Get:1 http://deb.debian.org/debian bullseye InRelease [116 kB]
Get:2 http://deb.debian.org/debian-security bullseye-security InRelease [48.4 kB]
Get:3 http://deb.debian.org/debian bullseye-updates InRelease [44.1 kB]
Get:4 http://deb.debian.org/debian bullseye/main amd64 Packages [8183 kB]
Get:5 http://deb.debian.org/debian-security bullseye-security/main amd64 Packages [237 kB]
Get:6 http://deb.debian.org/debian bullseye-updates/main amd64 Packages [14.6 kB]
Fetched 8643 kB in 2s (5374 kB/s)
Reading package lists... Done
root@624ee5a1de05:/usr/local/apache2/htdocs#
```

#apt-get install vim -y

```
root@624ee5a1de05:/usr/local/apache2/htdocs# apt-get install vim
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libgpm2 vim-common vim-runtime xxd
Suggested packages:
  gpm ctags vim-doc vim-scripts
The following NEW packages will be installed:
  libgpm2 vim vim-common vim-runtime xxd
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 8174 kB of archives.
After this operation, 36.9 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

Now we can able to edit the index.html file using vim editor

#vi index.html

#cat index.html

```
root@624ee5a1de05:/usr/local/apache2/htdocs# vi index.html
root@624ee5a1de05:/usr/local/apache2/htdocs# cat index.html
<html><body><h1>Getting response from Conatainer1</h1></body></html>
root@624ee5a1de05:/usr/local/apache2/htdocs#
```

Now copy the docker server public IP address paste in chrome tab

To host container1 image application in webserver mention the forward port number.

18.144.62.1:2729

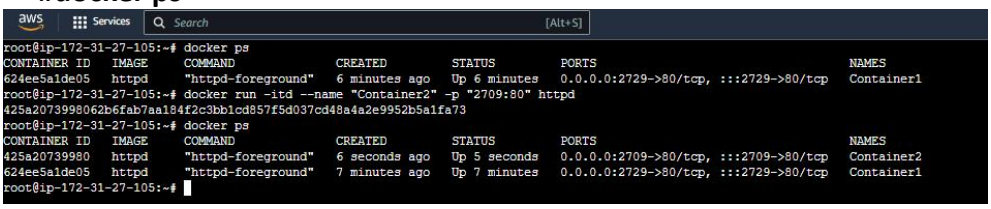
Successfully hosted httpd application using container1 from docker server.

Now creating an another container using same httpd docker image.

#docker run -itd --name “Container2” -p “2709:80” httpd

To list running docker containers

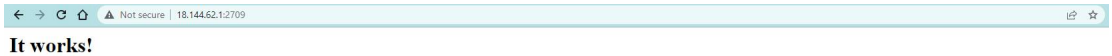
#docker ps



Successfully created another container (container2) using same httpd docker image

Copy docker server public IP address Paste in Chrome new tab.Mention the Container2 Port number.

18.144.62.1:2709



It works!

Successfully hosted another httpd application using container2 from docker server.

To pause the container

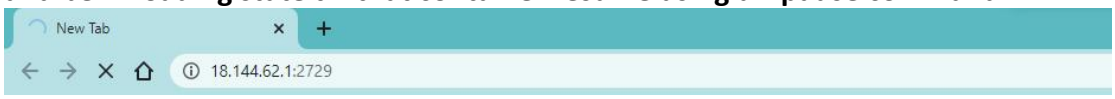
#docker pause <container ID>

Pause command is used to freeze the process running in the specified container.

```
root@ip-172-31-27-105:~# docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED         STATUS         PORTS                                     NAMES
425a20739980   httpd     "httpd-foreground"   6 seconds ago   Up 5 seconds   0.0.0.0:2709->80/tcp, :::2709->80/tcp     Container2
624ee5a1de05   httpd     "httpd-foreground"   7 minutes ago   Up 7 minutes   0.0.0.0:2729->80/tcp, :::2729->80/tcp     Container1
root@ip-172-31-27-105:~# docker pause 624ee5a1de05
624ee5a1de05
root@ip-172-31-27-105:~#
```

There we can see when we pause the Container1

Response from the container1 will be freeze and it will not show nothing and be in loading state till that container resume using un-pause command.



TO unpause the paused container

#docker unpause <container ID>

```
root@ip-172-31-27-105:~# docker unpause 624ee5a1de05
624ee5a1de05
root@ip-172-31-27-105:~#
```

Getting response from Conatainer1

Now Container1 response is working fine.

To remove the running container

#docker rm -f <containerID>

```
root@ip-172-31-27-105:~# docker ps
CONTAINER ID   IMAGE     COMMAND              CREATED          STATUS          PORTS                                     NAMES
425a20739980   httpd     "httpd-foreground"   5 minutes ago    Up 5 minutes    0.0.0.0:2709->80/tcp, :::2709->80/tcp     Container2
624ee5a1de05   httpd     "httpd-foreground"   12 minutes ago   Up 12 minutes   0.0.0.0:2729->80/tcp, :::2729->80/tcp     Container1
root@ip-172-31-27-105:~# docker rm -f 425a20739980
425a20739980
root@ip-172-31-27-105:~# docker rm -f 624ee5a1de05
624ee5a1de05
root@ip-172-31-27-105:~# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
root@ip-172-31-27-105:~#
```

To remove the Docker Image.

#docker rmi <image name>

To delete all the stopped container.

#docker container prune.

To stop and start the container

#docker stop <container ID>

#docker start <container ID>

To kill container and recover the killed container.

#docker kill <container ID>

#docker restart <container ID>

Note: if we know the killed Container ID can able to Recover the killed Container.