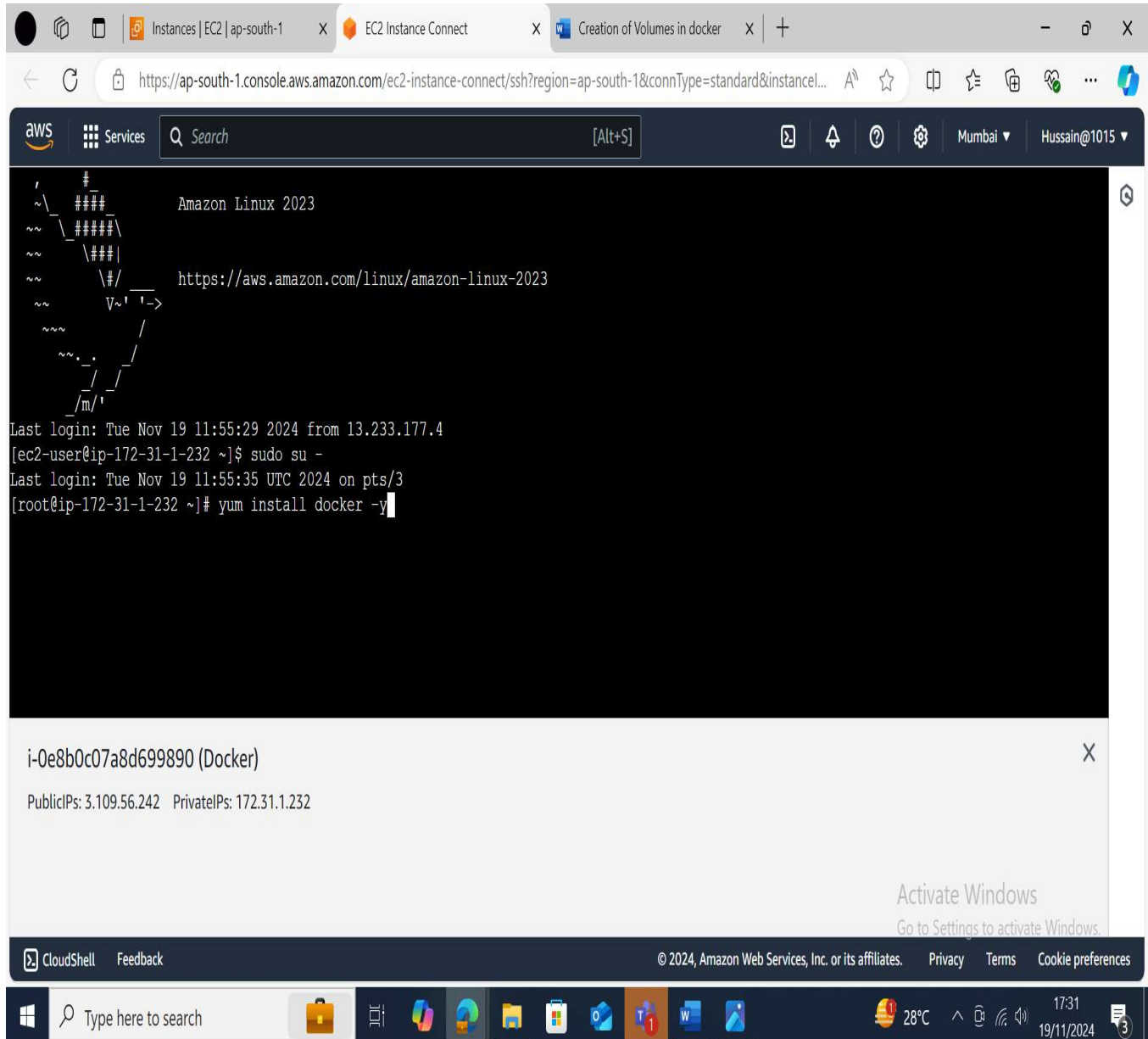


# DOCKER INSTALLATION IN LINUX

First, we have to install the docker to install the docker in linux

we have to enter `yum install docker -y`



## CREATIONS OF VOLUMES IN DOCKER

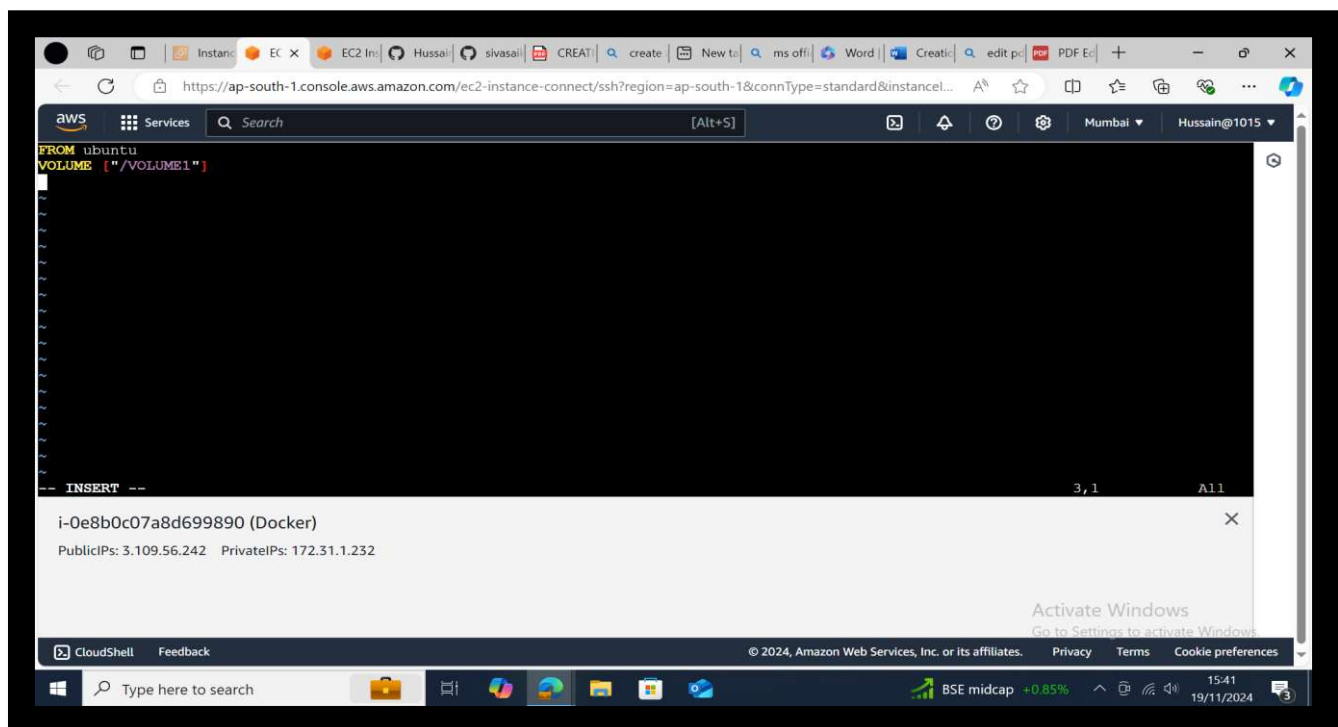
To create volumes in a docker there are 3 ways we:

1) Using the Dockerfile

2) Create a volume will create a container using command

3) Creating a local volume and mount it to the container

**1) Using Docker file:** To create a volume using a Dockerfile is a method in which we create Dockerfile and build it as an image using that image creating a container, In the Dockerfile using components we can create a volume always remember that in Dockerfile starting letter 'd' will be always upper case.



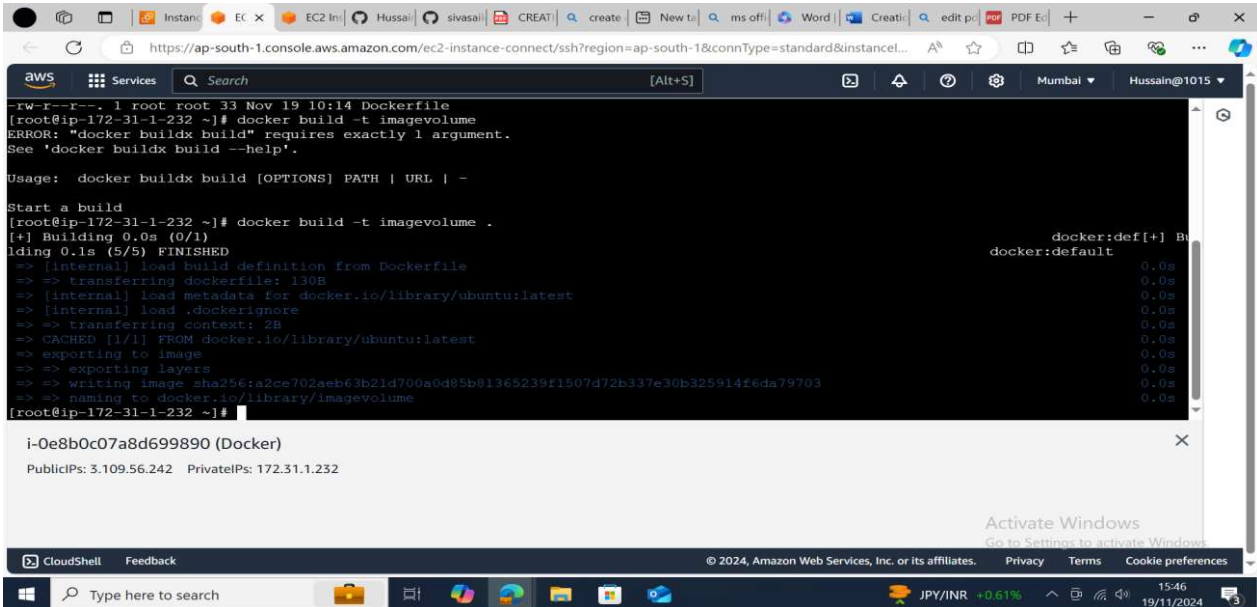
The above image is the script of Dockerfile description of components:

**FROM:** it is main component in Dockerfile using this only the image will be built

**VOLUME:** it is component that is used to create a volume will create a container after this component the name of the container will be provide in ["/\*"] place of star

After creating this file execute this command to build the image using that file

**docker build -t imagevolume .**



```
-rw-r--r-- 1 root root 33 Nov 19 10:14 Dockerfile
[root@ip-172-31-1-232 ~]# docker build -t imagevolume .
ERROR: "docker buildx build" requires exactly 1 argument.
See 'docker buildx build --help'.

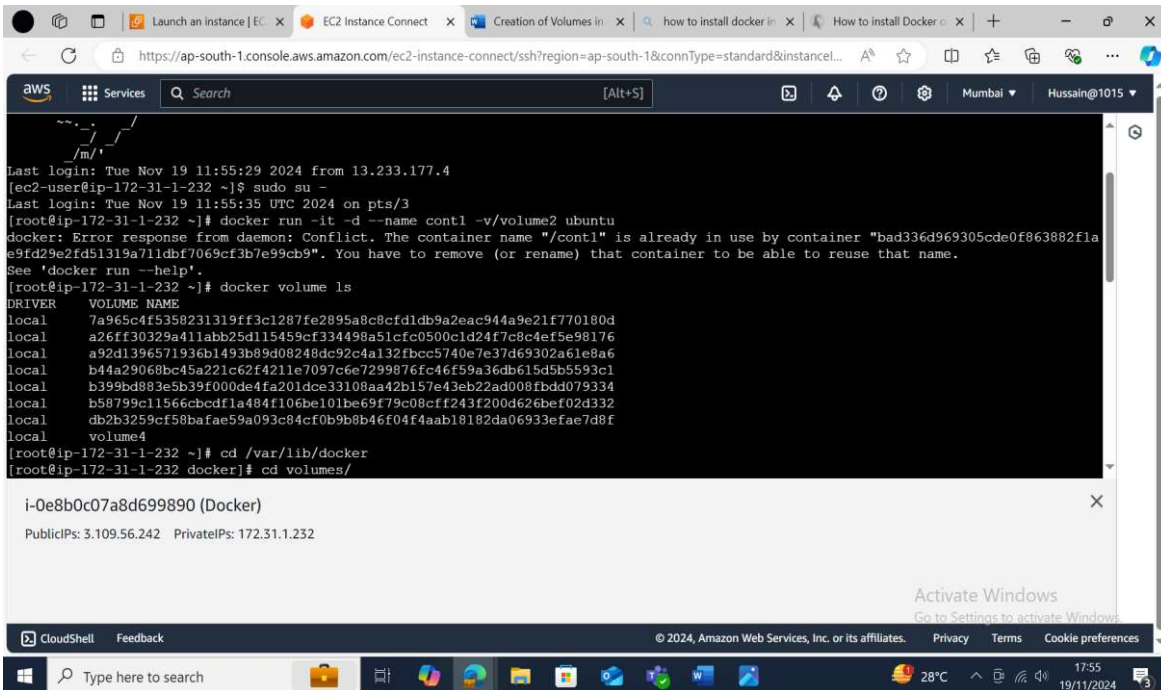
Usage: docker buildx build [OPTIONS] PATH | URL | -

Start a build
[root@ip-172-31-1-232 ~]# docker build -t imagevolume .
[*] Building 0.0s (0/1)
lding 0.1s (5/5) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 130B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [internal] load .dockerignore
=> => transferring context: 2B
=> CACHED [1/1] FROM docker.io/library/ubuntu:latest
=> exporting to image
=> => exporting layers
=> => writing image sha256:a2ce762aeb63b21d700a0d85b81365239f1507d72b337e30b325914f6da79703
=> => naming to docker.io/library/imagevolume
[root@ip-172-31-1-232 ~]#
```

i-0e8b0c07a8d699890 (Docker)  
PublicIPs: 3.109.56.242 PrivateIPs: 172.31.1.232

Now create container using the newly created image by execute this command

**docker run -it -d --name cont1 -v/volume2 ubuntu**



```
Last login: Tue Nov 19 11:55:29 2024 from 13.233.177.4
[ec2-user@ip-172-31-1-232 ~]$ sudo su -
Last login: Tue Nov 19 11:55:35 UTC 2024 on pts/3
[root@ip-172-31-1-232 ~]# docker run -it -d --name cont1 -v/volume2 ubuntu
docker: Error response from daemon: Conflict. The container name "/cont1" is already in use by container "bad336d969305cde0f863882f1a
e9fd29e2fd51319a711dbf7069cf3b7e99cb9". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# docker volume ls
DRIVER      VOLUME NAME
local       7a965c4f5358231319ff3c1287fe2895a8c8cfd1db9a2eac944a9e21f770180d
local       a26ff30329a411abb25d115459cf334498a51cfc0500c1d24f7c8c4ef5e98176
local       a92d1396571936b1493b89d08248dc92c4a132fbcc5740e7e37d69302a61e8a6
local       b44a29068bc45a221c62f4211e7097c6e7299876fc46f59a36db615d5b5593c1
local       b399bd883e5b39f000de4fa201dce33108aa42b157e43eb22ad008fbd4079334
local       b58799c11566cbcd1a484f106be101be69f79c08cfff243f200d626bef02d332
local       db2b3259cf58bafae59a093c84cf0b9b8b46f04f4aab18182da06933efae7d8f
local       volume4
[root@ip-172-31-1-232 ~]# cd /var/lib/docker
[root@ip-172-31-1-232 docker]# cd volumes/
```

i-0e8b0c07a8d699890 (Docker)  
PublicIPs: 3.109.56.242 PrivateIPs: 172.31.1.232

Now go in to the container using exec command and go to root directory there use the command 'ls' you will see the created volume as a directory as shown in the below image

```
CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
a9f57d3cc6b6    httpd    "httpd-foreground"    3 hours ago    Up 3 hours    80/tcp    contb
b7fd466d58bf    imagea    "/bin/bash"    3 hours ago    Up 3 hours    80/tcp    conta
3b27501d3bc7    nginx    "/docker-entrypoint..."    4 hours ago    Up 4 hours    80/tcp    cont6
924407ccdb05    httpd    "httpd-foreground"    4 hours ago    Up 4 hours    80/tcp    cont5
2b1194661a8d    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont4
99d82fb821a4    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont43
1ba5bcb14bb4    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont3
bbd129a393f8    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont2
bad336d96930    imagevolume    "/bin/bash"    4 hours ago    Up 4 hours    cont1

[root@ip-172-31-1-232 volumes]# ls
7a96504f5358231319f301287fe2895a8b0cfd1db9a2eac944a9e21f770180d  b58799c11566b0cfd1a484f106be101be69f79c08cff243f200d626bef02d332
a26ff30329a411abb25d115459ef334498a51cfc0500c1d24e7c8e4ef5e98176  backingFsBlockDev
a92d1396571936b1493b89d08248dc92c4a132fbcc5740e7e37d69302a61e8a6  db2b3259cf58bafae59a093c84cf0b9b8b46f04f4aab18182da06933efae7d8f
b399bd883e5b39f000de4fa201dce33108aa42b157e43eb22ad008fbd079334  metadata.db
b44a29068bc45a221c62f4211e7097c6e7299876fc46f59a36db615d5b5593c1  volume4

[root@ip-172-31-1-232 volumes]# docker exec -it cont /bin/bash
Error response from daemon: No such container: cont
[root@ip-172-31-1-232 volumes]# docker exec -it cont1 /bin/bash
root@bad336d96930:/# ls
VOLUME1  bin  boot  dev  dir1  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  usr  var
root@bad336d96930:/#
```

**2) create a volume will create a container using commands:** In this method we use a simple command to create a volume using any of the required base image

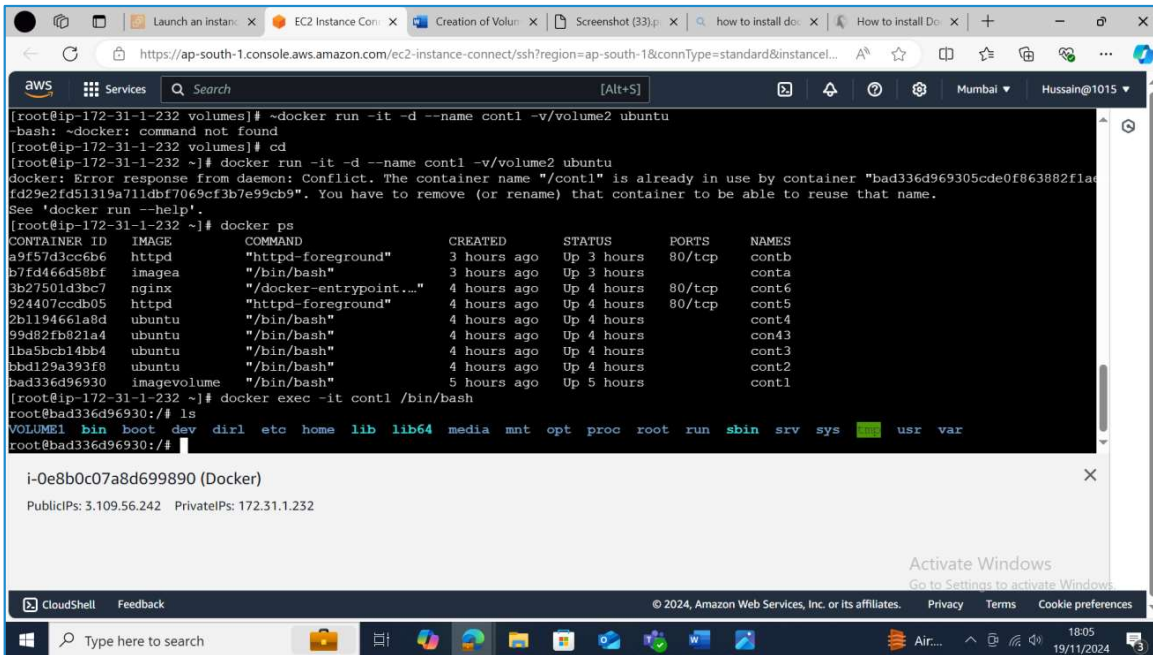
```
[root@ip-172-31-1-232 volumes]# ~docker run -it -d --name cont1 -v/volume2 ubuntu
-bash: ~docker: command not found
[root@ip-172-31-1-232 volumes]# cd
[root@ip-172-31-1-232 ~]# docker run -it -d --name cont1 -v/volume2 ubuntu
docker: Error response from daemon: Conflict. The container name "/cont1" is already in use by container "bad336d969305cde0f863882f1a0fd29e2fd51319a711dbf7069cf3b7e99cb9". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# docker ps
CONTAINER ID    IMAGE    COMMAND    CREATED    STATUS    PORTS    NAMES
a9f57d3cc6b6    httpd    "httpd-foreground"    3 hours ago    Up 3 hours    80/tcp    contb
b7fd466d58bf    imagea    "/bin/bash"    3 hours ago    Up 4 hours    80/tcp    conta
3b27501d3bc7    nginx    "/docker-entrypoint..."    4 hours ago    Up 4 hours    80/tcp    cont6
924407ccdb05    httpd    "httpd-foreground"    4 hours ago    Up 4 hours    80/tcp    cont5
2b1194661a8d    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont4
99d82fb821a4    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont43
1ba5bcb14bb4    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont3
bbd129a393f8    ubuntu    "/bin/bash"    4 hours ago    Up 4 hours    cont2
bad336d96930    imagevolume    "/bin/bash"    5 hours ago    Up 5 hours    cont1

[root@ip-172-31-1-232 ~]# docker exec -it cont1 /bin/bash
root@bad336d96930:/# ls
VOLUME1  bin  boot  dev  dir1  etc  home  lib  lib64  media  mnt  opt  proc  root  run  sbin  srv  sys  usr  var
root@bad336d96930:/#
```



## Docker run -it -d --name cont1 -v/volume2 ubuntu

After executing the above command go into the container using exec command and go to root directory there use the command 'ls' you will see the created volume as a directory as shown in the below image.

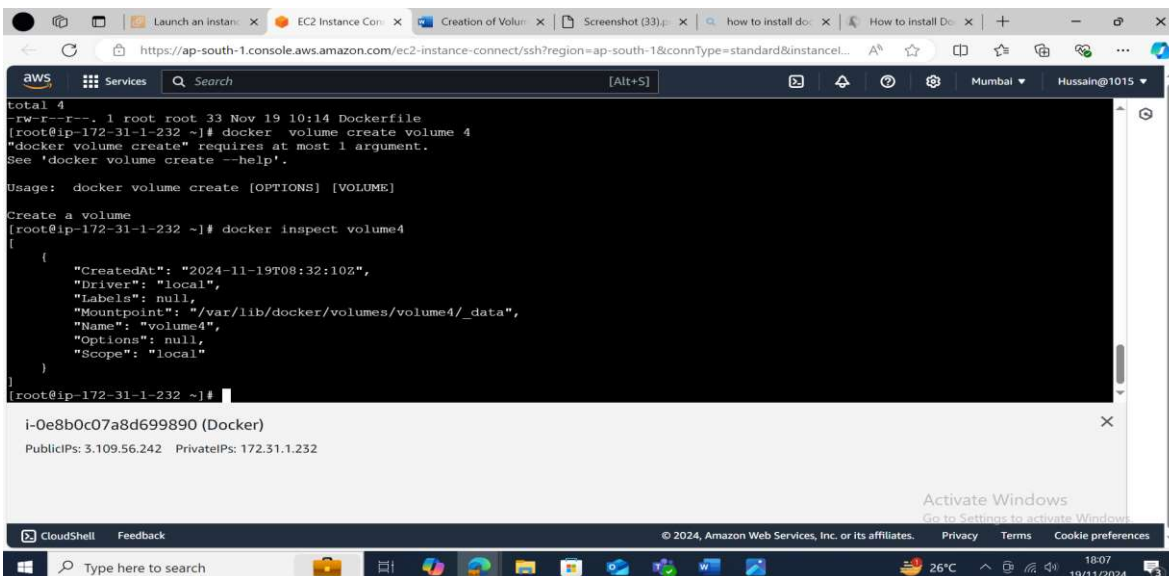


The screenshot shows a terminal window within the AWS CloudShell interface. The user is in a container named 'cont1' and has executed the command 'docker run -it -d --name cont1 -v/volume2 ubuntu'. The terminal output shows the container is already in use by another container, so the user runs 'docker ps' to list all containers. The output shows a table of containers, including 'cont1' which is running. The user then runs 'docker exec -it cont1 /bin/bash' to enter the container. The terminal output shows the user is now in the container's root directory and runs 'ls', which lists the contents of the container's root directory, including the 'volume2' directory.

```
[root@ip-172-31-1-232 volumes]# docker run -it -d --name cont1 -v/volume2 ubuntu
-bash: ~docker: command not found
[root@ip-172-31-1-232 volumes]# cd
[root@ip-172-31-1-232 ~]# docker run -it -d --name cont1 -v/volume2 ubuntu
docker: Error response from daemon: Conflict. The container name "/cont1" is already in use by container "bad336d969305cde0f863802fla
fd29e2fd51319a711dbf7069cf3b7e99cb9". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
a9f57d3cc6b6   httpd     "httpd-foreground"      3 hours ago Up 3 hours 80/tcp    contb
b7fd466d58bf   imagea    "/bin/bash"             3 hours ago Up 3 hours    conta
3b27501d3bc7   nginx     "/docker-entrypoint..." 4 hours ago Up 4 hours 80/tcp    cont6
924407ccdb05   httpd     "httpd-foreground"      4 hours ago Up 4 hours 80/tcp    cont5
2b1194661a8d   ubuntu    "/bin/bash"             4 hours ago Up 4 hours    cont4
99d82fb821a4   ubuntu    "/bin/bash"             4 hours ago Up 4 hours    con43
1ba5bcb14bb4   ubuntu    "/bin/bash"             4 hours ago Up 4 hours    cont3
bbd129a393f8   ubuntu    "/bin/bash"             4 hours ago Up 4 hours    cont2
bad336d96930   imagevolum "/bin/bash"             5 hours ago Up 5 hours    cont1
[root@ip-172-31-1-232 ~]# docker exec -it cont1 /bin/bash
root@bad336d96930:/# ls
VOLUME1 bin boot dev dir1 etc home lib lib64 media mnt opt proc root run sbin srv sys usr var
root@bad336d96930:/#
```

## 3) Creating a local volume and mount it to the container:

In this method first we create a local volume using the command docker volume create volume 4



The screenshot shows a terminal window within the AWS CloudShell interface. The user is in a container named 'cont1' and has executed the command 'docker volume create volume 4'. The terminal output shows the volume is created successfully. The user then runs 'docker inspect volume4' to view the details of the volume. The output shows the volume is local and has a mountpoint of '/var/lib/docker/volumes/volume4/\_data'.

```
Total 4
-rw-r--r-- 1 root root 33 Nov 19 10:14 Dockerfile
[root@ip-172-31-1-232 ~]# docker volume create volume 4
"docker volume create" requires at most 1 argument.
See 'docker volume create --help'.

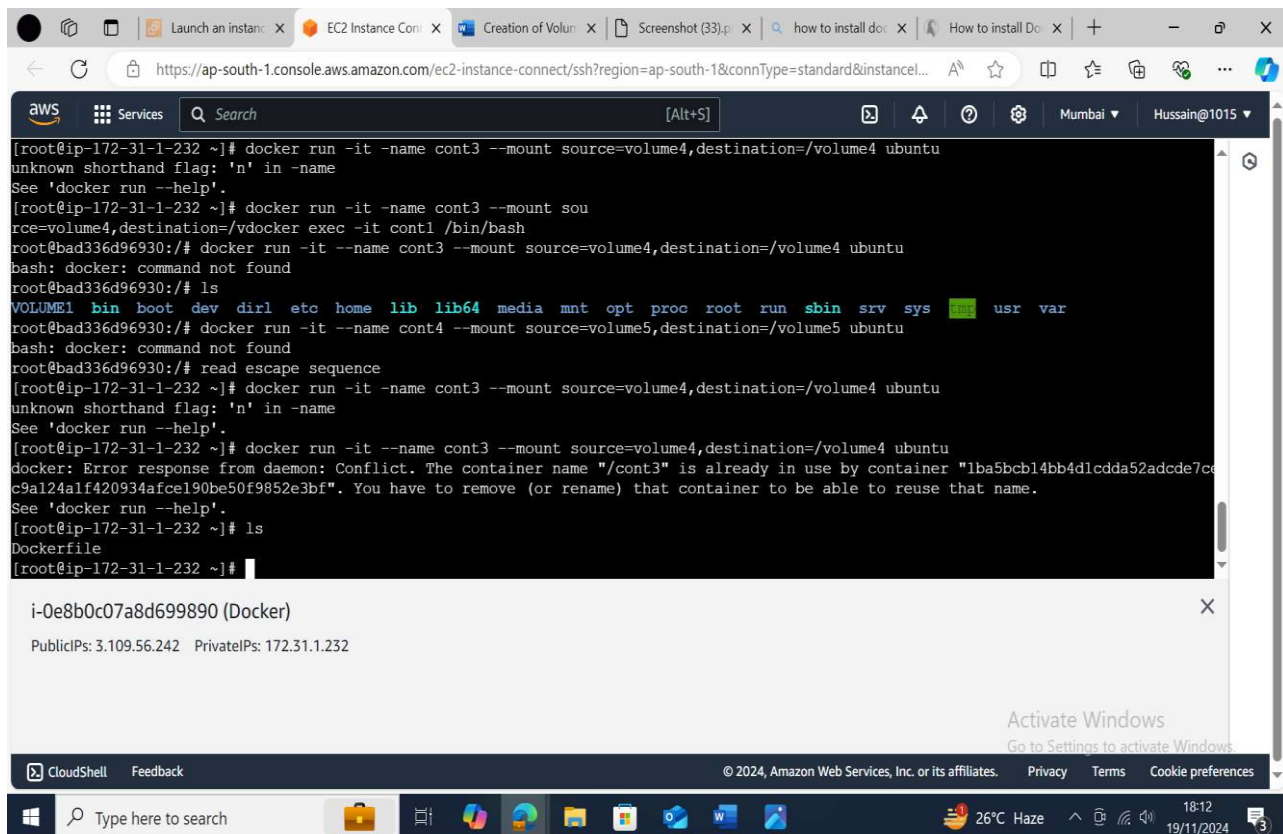
Usage: docker volume create [OPTIONS] [VOLUME]

Create a volume
[root@ip-172-31-1-232 ~]# docker inspect volume4
[
  {
    "CreatedAt": "2024-11-19T08:32:10Z",
    "Driver": "local",
    "Labels": null,
    "Mountpoint": "/var/lib/docker/volumes/volume4/_data",
    "Name": "volume4",
    "Options": null,
    "Scope": "local"
  }
]
```

After that execute the below command to mount the created volume and to create container.

**docker run -it --name cont3 --mount source=volume4, destination=/volume4 ubuntu**

after executing the above command go into the container using exec command and go to root directory there use the command 'ls' you will see the created volume as a directory as shown in the below image



```
[root@ip-172-31-1-232 ~]# docker run -it --name cont3 --mount source=volume4,destination=/volume4 ubuntu
unknown shorthand flag: 'n' in -name
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# docker run -it --name cont3 --mount sou
rce=volume4,destination=/vdocker exec -it cont1 /bin/bash
root@bad336d96930:/# docker run -it --name cont3 --mount source=volume4,destination=/volume4 ubuntu
bash: docker: command not found
root@bad336d96930:/# ls
VOLUME1 bin boot dev dir1 etc home lib lib64 media mnt opt proc root run sbin srv sys usr var
root@bad336d96930:/# docker run -it --name cont4 --mount source=volume5,destination=/volume5 ubuntu
bash: docker: command not found
root@bad336d96930:/# read escape sequence
[root@ip-172-31-1-232 ~]# docker run -it --name cont3 --mount source=volume4,destination=/volume4 ubuntu
unknown shorthand flag: 'n' in -name
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# docker run -it --name cont3 --mount source=volume4,destination=/volume4 ubuntu
docker: Error response from daemon: Conflict. The container name "/cont3" is already in use by container "1ba5bcb14bb4d1cdda52adcde7cc9a124a1f420934afce190be50f9852e3bf". You have to remove (or rename) that container to be able to reuse that name.
See 'docker run --help'.
[root@ip-172-31-1-232 ~]# ls
Dockerfile
[root@ip-172-31-1-232 ~]#
```

i-0e8b0c07a8d699890 (Docker)

PublicIPs: 3.109.56.242 PrivateIPs: 172.31.1.232

Activate Windows  
Go to Settings to activate Windows

CloudShell Feedback

© 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences