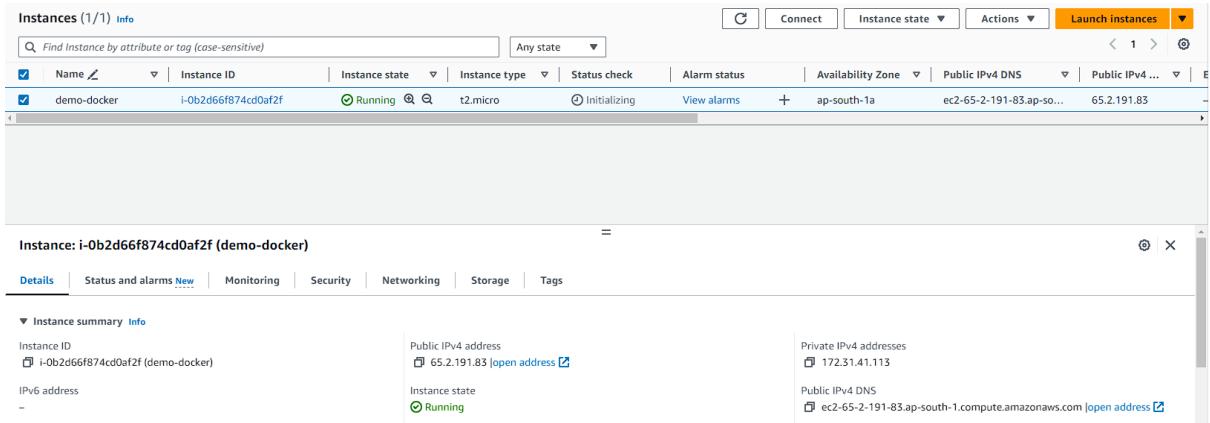


# Docker Setup

1. So, to setup Docker you need to create a Virtual Machine based on Ubuntu OS.
2. For that login to AWS Console and navigate to EC2. Then click on launch instances and create an instance based on Ubuntu OS.
3. Once the instance is launched then you need to SSH into it using Putty tool or command prompt.



The screenshot shows the AWS EC2 Instances page. At the top, there is a search bar labeled "Find Instance by attribute or tag (case-sensitive)" and a dropdown menu set to "Any state". Below the search bar is a table with one row, showing the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...
demo-docker	i-0b2d66f874cd0af2f	Running	t2.micro	Initializing	View alarms	ap-south-1a	ec2-65-2-191-83.ap-sou...	65.2.191.83

Below the table, the instance details are expanded. The instance ID is i-0b2d66f874cd0af2f (demo-docker). The public IPv4 address is 65.2.191.83, and the private IPv4 address is 172.31.41.113. The instance state is Running. The public IPv4 DNS is ec2-65-2-191-83.ap-south-1.compute.amazonaws.com.

```
ubuntu@ip-172-31-41-113: ~ + - 

System information as of Thu Feb  8 11:45:17 UTC 2024

System load: 0.6240234375      Processes:          100
Usage of /: 20.6% of 7.57GB    Users logged in:      0
Memory usage: 22%              IPv4 address for eth0: 172.31.41.113
Swap usage:   0%              

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-41-113:~$ |
```

4. Now you need to go to the docker documentation page to install it. You can follow the link below.

<https://docs.docker.com/engine/install/ubuntu/>

5. After that you need to run the commands to install docker. Run the below commands one by one or you can also use the documentation for the installation of docker.

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o
/etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc
```

```

# Add the repository to Apt sources:
echo \
    "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/ubuntu \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin

```

- Once the docker is installed then you need to run the command for its status check.

### **systemctl status docker**

```

root@ip-172-31-41-113 ~
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2024-02-08 13:52:14 UTC; 31s ago
     Docs: https://docs.docker.com
 Main PID: 2802 (dockerd)
   Tasks: 8
    Memory: 38.3M
      CPU: 338ms
     CGroup: /system.slice/docker.service
             └─2802 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock

Feb 08 13:52:13 ip-172-31-41-113 systemd[1]: Starting Docker Application Container Engine...
Feb 08 13:52:13 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:13.485534002Z level=info msg="Starting up"
Feb 08 13:52:13 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:13.488316402Z level=info msg="detected 127.0.0.53 nameserver, assuming systemd-resolved, so using resolv.conf: /run/systemd/resolve/resolv.conf"
Feb 08 13:52:13 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:13.680000000Z level=info msg="Loading containers: start"
Feb 08 13:52:13 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:14.002235000Z level=info msg="parsed configuration: /etc/docker/daemon.json"
Feb 08 13:52:13 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:14.079512000Z level=info msg="Docker daemon" commit=f4174735 containerd=snapshotted=false storage-driver=overlay2 version=25.0.3
Feb 08 13:52:14 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:14.030033182Z level=info msg="Daemon has completed initialization"
Feb 08 13:52:14 ip-172-31-41-113 dockerd[2802]: time=2024-02-08T13:52:14.085533227Z level=info msg="API listen on /run/docker.sock"
Feb 08 13:52:14 ip-172-31-41-113 systemd[1]: Started Docker Application Container Engine.

```

- If you run the command to look for docker images you will see that there is nothing for now.

### **docker images**

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
root				

- Currently you are in root user but if you will try to run the same command using the Ubuntu user then you will see the error. So, by default only root user can access the docker daemon. If you want any other user to use docker then you need to add that user.

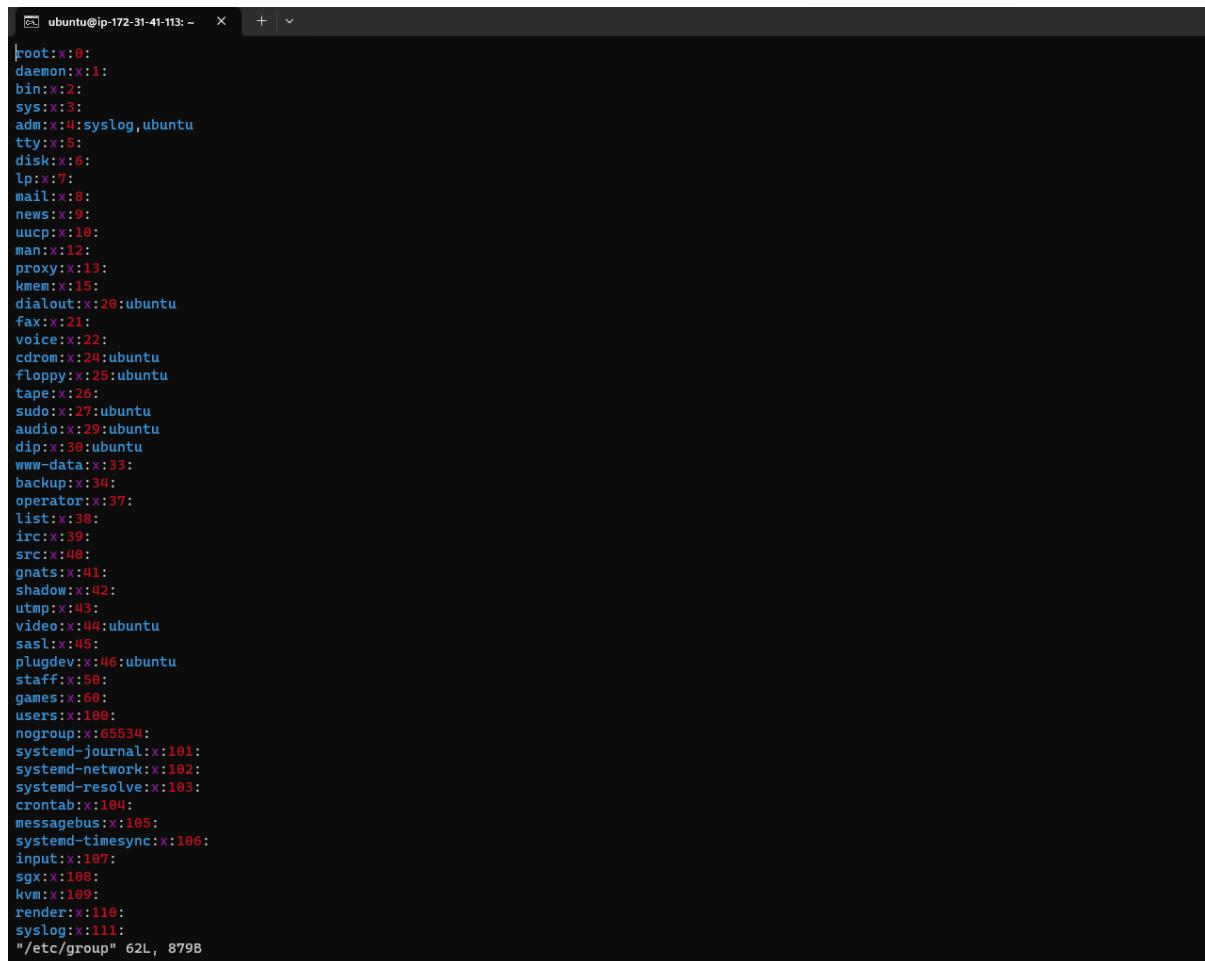
```

root@ip-172-31-41-113 ~
root@ip-172-31-41-113 ~# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-41-113 ~# logout
ubuntu@ip-172-31-41-113 ~$ docker images
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/images/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-41-113 ~$ 

```

9. Now if you want to look at the groups you can use this command.

```
sudo vim /etc/group
```

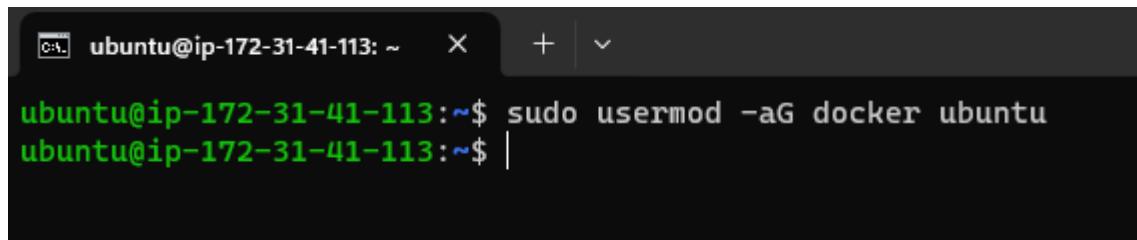


```
ubuntu@ip-172-31-41-113: ~ + ^
```

```
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:syslog,ubuntu
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:ubuntu
fax:x:21:
voice:x:22:
cdrom:x:24:ubuntu
floppy:x:25:ubuntu
tape:x:26:
sudo:x:27:ubuntu
audio:x:29:ubuntu
dip:x:30:ubuntu
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:::42:
utmp:x:43:
video:x:44:ubuntu
sasl:x:45:
plugdev:x:46:ubuntu
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
systemd-journal:x:101:
systemd-network:x:102:
systemd-resolve:x:103:
crontab:x:104:
messagebus:x:105:
systemd-timesync:x:106:
input:x:107:
sgx:::108:
kvm:x:109:
render:::110:
syslog:x:111:
"/etc/group" 62L, 879B
```

10. Now you'll use the usermod command to add Ubuntu as a user so that you can use docker on the Ubuntu user.

```
sudo usermod -aG docker ubuntu
```



```
ubuntu@ip-172-31-41-113: ~ + ^
```

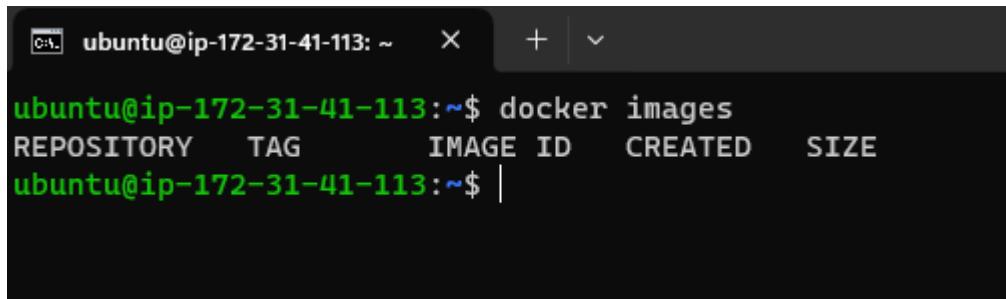
```
ubuntu@ip-172-31-41-113:~$ sudo usermod -aG docker ubuntu
ubuntu@ip-172-31-41-113:~$ |
```

11. Now if you will run this below command, you can see that ubuntu has been added in docker as a user.

```
id ubuntu
```

```
ubuntu@ip-172-31-41-113:~$ id ubuntu
uid=1000(ubuntu) gid=1000(ubuntu) groups=1000(ubuntu),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip)
,44(video),46(plugdev),119(netdev),120(lxd),999(docker)
ubuntu@ip-172-31-41-113:~$ |
```

12. So, if you will run the docker images command you will see that you are still getting permission denied. For that you need to logout or exit from it then login again in the session you will see that it is working.



```
ubuntu@ip-172-31-41-113:~$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
ubuntu@ip-172-31-41-113:~$ |
```

13. Now let's check and run some docker commands. The first command is for hello world.

**sudo docker run hello-world**

14. Here you will see that in the start it was unable to find the image it will try and pull the image from docker hub.

```
ubuntu@ip-172-31-41-113:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
c1ec31eb5944: Pull complete
Digest: sha256:4bd78111b6914a99dbc560e6a20eab57ff6655aea4a80c50b0c5491968cbc2e6
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/get-started/
ubuntu@ip-172-31-41-113:~$ |
```

15. Now run these commands to see you image.

**docker ps**

**docker ps -a**



A terminal window titled "ubuntu@ip-172-31-41-113: ~" showing the output of Docker commands. The first command, "docker ps", lists running containers. The second command, "docker ps -a", lists all containers, including ones that have exited. The output shows a single container named "infallible\_stonebraker" created 3 minutes ago, which has exited.

```
ubuntu@ip-172-31-41-113:~$ docker ps
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS
ubuntu@ip-172-31-41-113:~$ docker ps -a
CONTAINER ID   IMAGE      COMMAND   CREATED     STATUS      PORTS      NAMES
f6d4eeala345   hello-world   "/hello"   3 minutes ago   Exited (0) 3 minutes ago
ubuntu@ip-172-31-41-113:~$ |
```