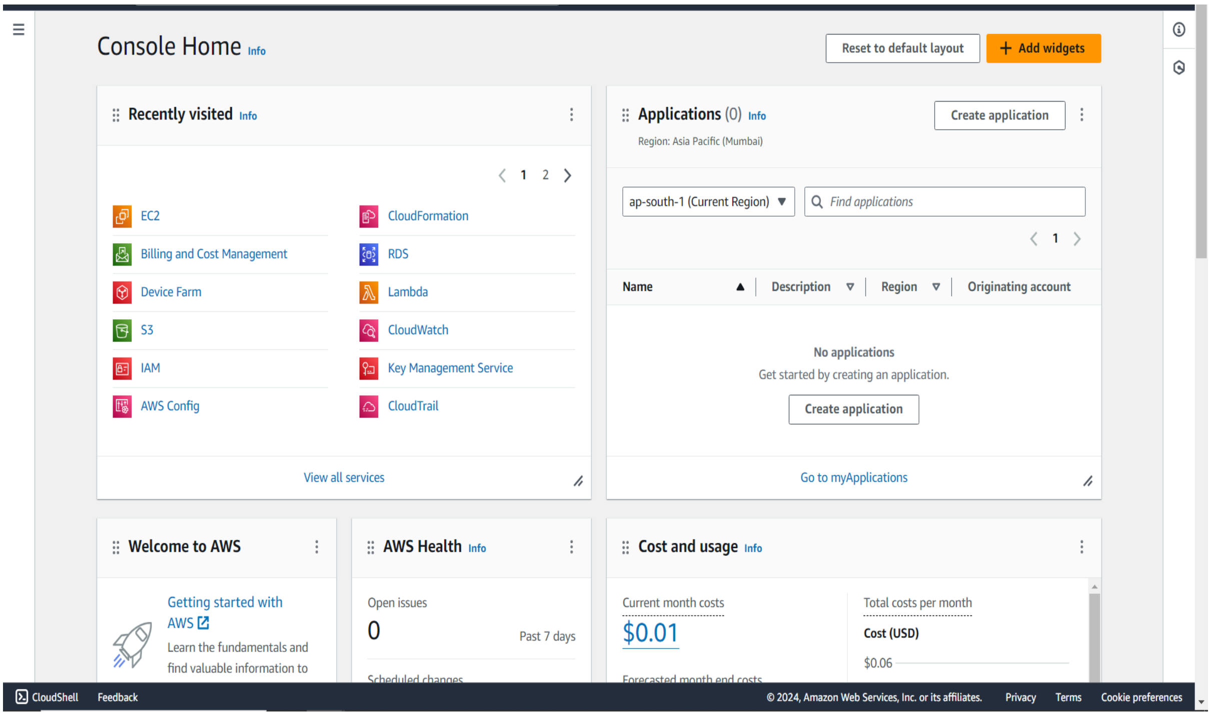
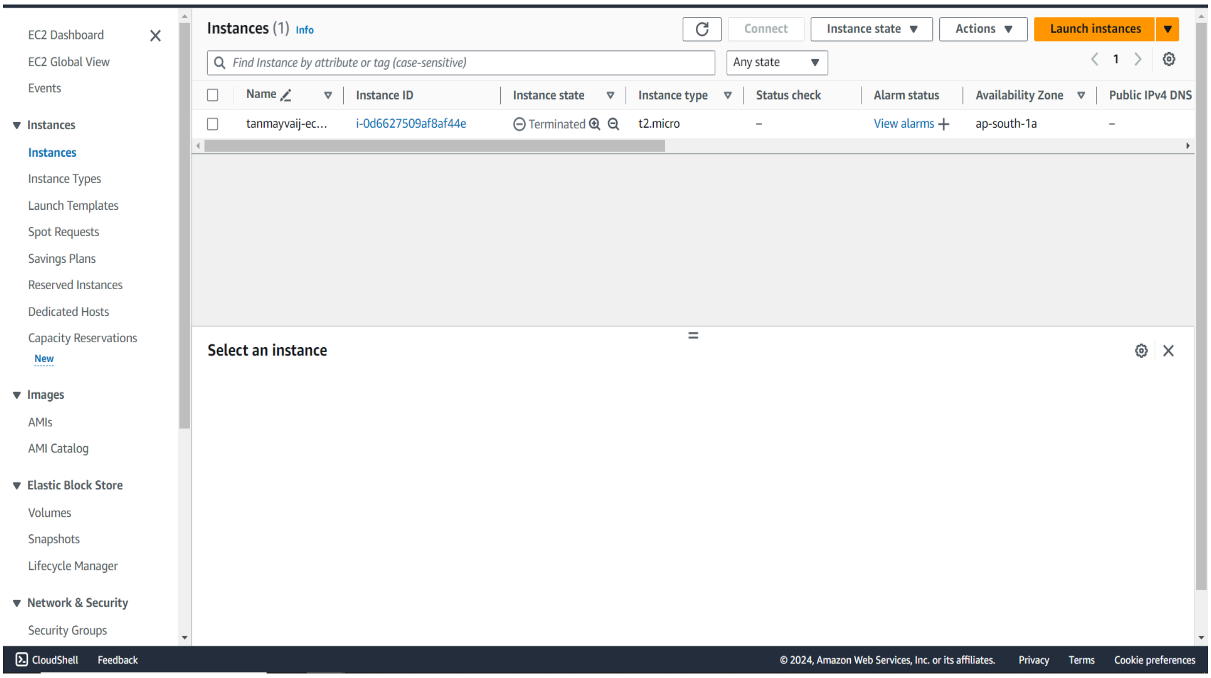
Create a docker image and push it to the docker hub (public repository).

Step 1: Sign in to aws account

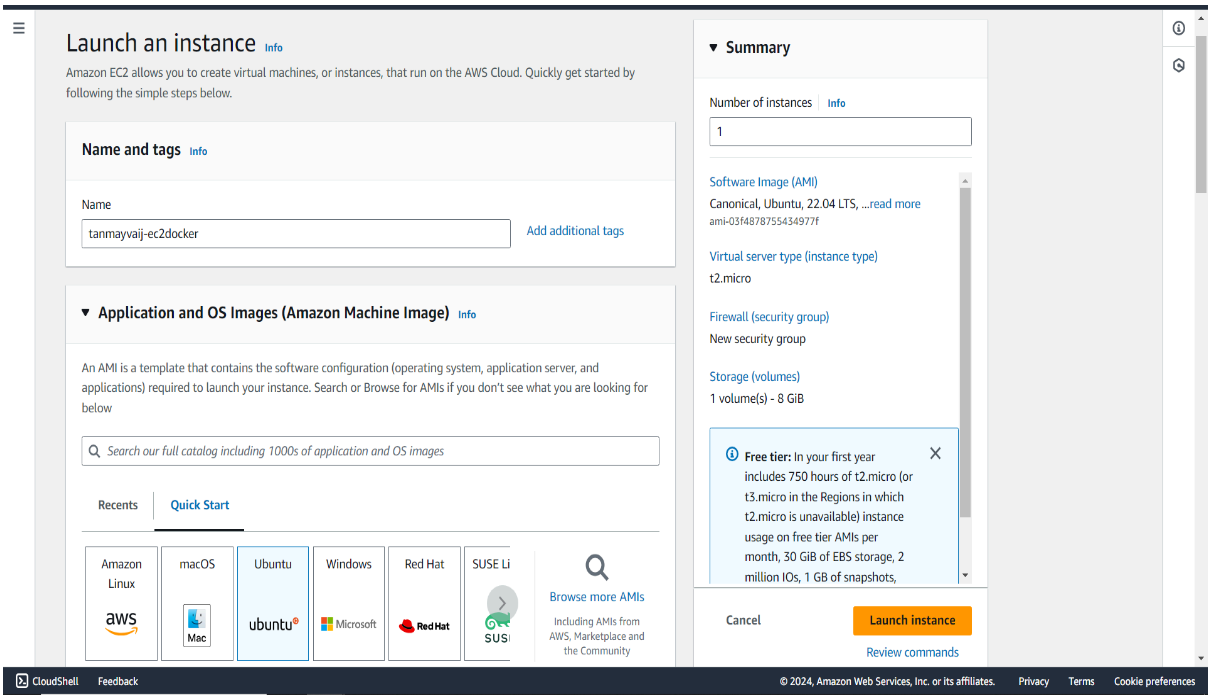


Step 2: Go to 'instances' in the EC2 dashboard and click on 'Launch Instances'



□	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
□	tanmayvaij-ec...	i-0d6627509af8af44e	Terminated	t2.micro	–	View alarms +	ap-south-1a	–

Step 3: Give a name to the instance and select ubuntu os.



Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: tanmayvaij-ec2docker

Add additional tags

▼ Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents | Quick Start

Amazon Linux | macOS | Ubuntu | Windows | Red Hat | SUSE Li

Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

▼ Summary

Number of instances Info: 1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-03f4878755434977f

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

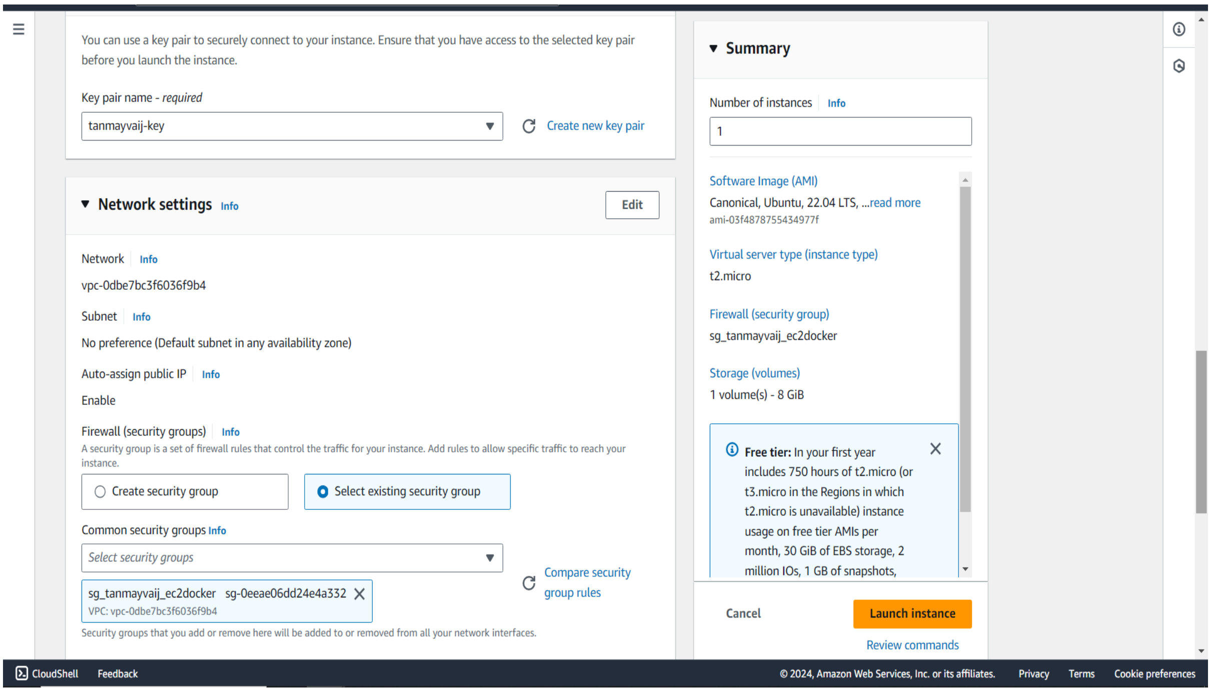
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots,

Cancel | Launch instance

Review commands

Step 4: Select key pair and security group.



You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: tanmayvaij-key

Create new key pair

▼ Network settings Info | Edit

Network Info
vpc-0dbe7bc3f6036f9b4

Subnet Info
No preference (Default subnet in any availability zone)

Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

○ Create security group ● Select existing security group

Common security groups Info
Select security groups

sg_tanmayvaij_ec2docker sg-0eeae06dd24e4a332
VPC: vpc-0dbe7bc3f6036f9b4

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

▼ Summary

Number of instances Info: 1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-03f4878755434977f

Virtual server type (instance type)
t2.micro

Firewall (security group)
sg_tanmayvaij_ec2docker

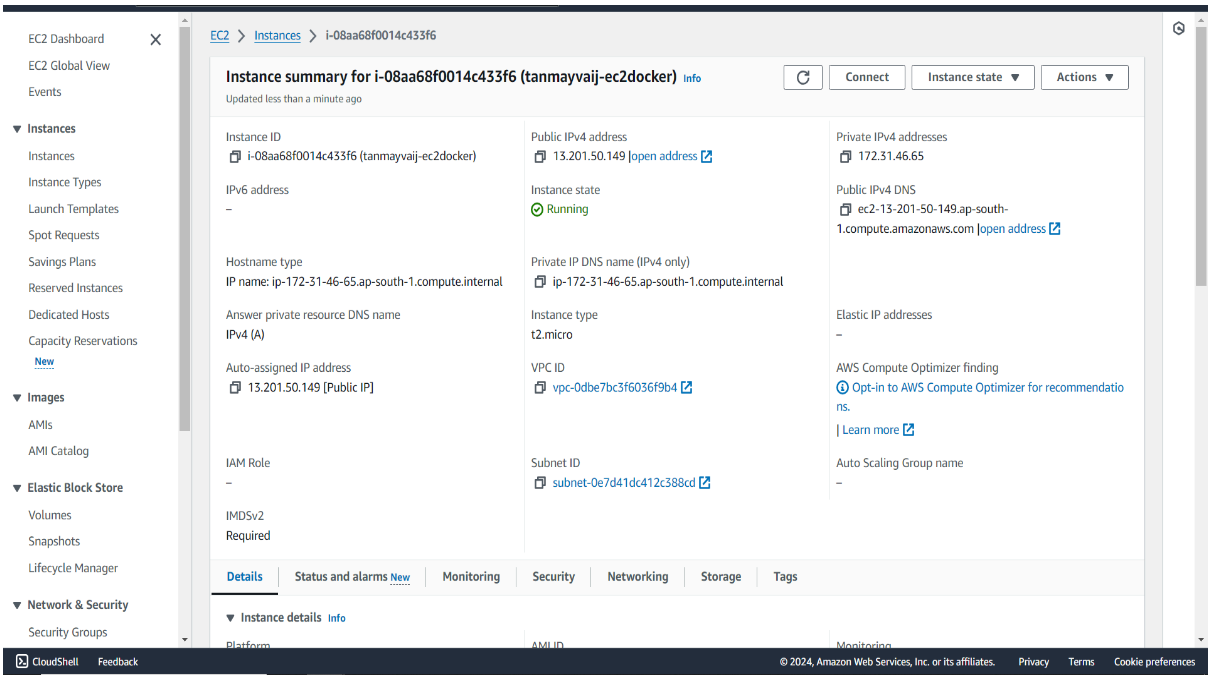
Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million IOs, 1 GB of snapshots,

Cancel | Launch instance

Review commands

Step 5: Click on the created running instance and click on 'Connect'



Instance summary for i-08aa68f0014c433f6 (tanmayvaij-ec2docker) Info

Updated less than a minute ago

Instance ID: i-08aa68f0014c433f6 (tanmayvaij-ec2docker)

Public IPv4 address: 13.201.50.149 | open address

Private IPv4 addresses: 172.31.46.65

IPv6 address: –

Instance state: Running

Public IPv4 DNS: ec2-13-201-50-149.ap-south-1.compute.amazonaws.com | open address

Hostname type: IP name: ip-172-31-46-65.ap-south-1.compute.internal

Private IP DNS name (IPv4 only): ip-172-31-46-65.ap-south-1.compute.internal

Answer private resource DNS name: IPv4 (A)

Instance type: t2.micro

Elastic IP addresses: –

Auto-assigned IP address: 13.201.50.149 [Public IP]

VPC ID: vpc-0dbe7bc3f6036f9b4

AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more

IAM Role: –

Subnet ID: subnet-0e7d41dc412c388cd

Auto Scaling Group name: –

IMDSv2: Required

Instance details Info

Platform

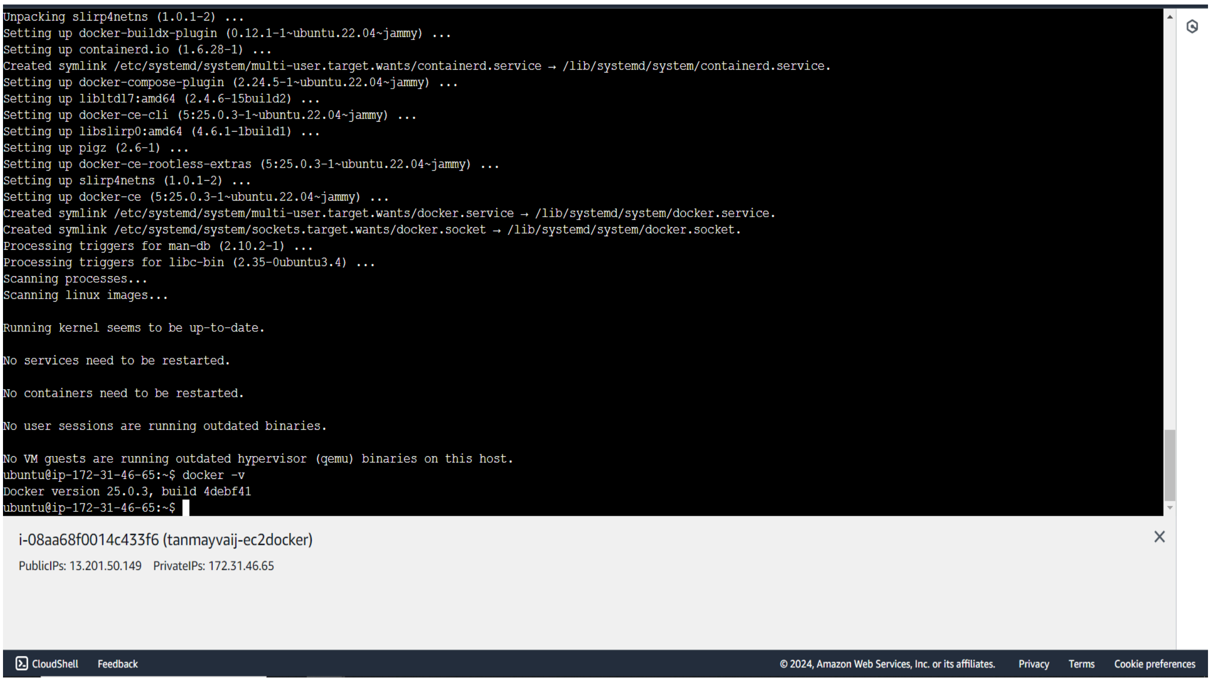
AMI ID

Monitoring

Step 6: After entering the aws ec2 terminal, install docker on the instance using following commands.

Commands:

```
$ sudo apt update -y

$ sudo apt install apt-transport-https curl ca-certificates gnupg-agent \
software-properties-common -y

$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

$ sudo add-apt-repository "deb [arch=amd64] \
https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"

$ sudo apt install docker-ce docker-ce-cli containerd.io
```



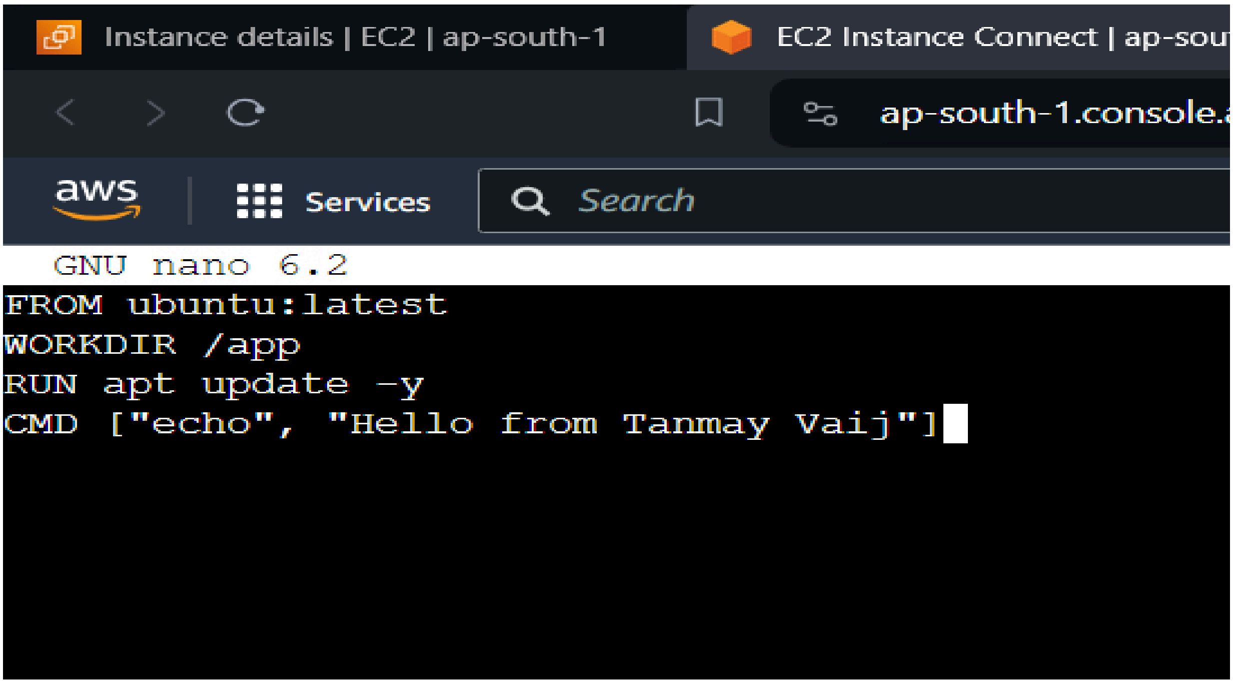
```
Unpacking slirp4netns (1.0.1-2) ...
Setting up docker-buildx-plugin (0.12.1-1~ubuntu.22.04~jammy) ...
Setting up containerd.io (1.6.28-1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service → /lib/systemd/system/containerd.service.
Setting up docker-compose-plugin (2.24.5-1~ubuntu.22.04~jammy) ...
Setting up libltdl7:amd64 (2.4.6-15build2) ...
Setting up docker-ce-cli (5:25.0.3-1~ubuntu.22.04~jammy) ...
Setting up libslirp0:amd64 (4.6.1-1build1) ...
Setting up pigz (2.6-1) ...
Setting up docker-ce-rootless-extras (5:25.0.3-1~ubuntu.22.04~jammy) ...
Setting up slirp4netns (1.0.1-2) ...
Setting up docker-ce (5:25.0.3-1~ubuntu.22.04~jammy) ...
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /lib/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/systemd/system/docker.socket.
Processing triggers for man-db (2.10.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-46-65:~$ docker -v
Docker version 25.0.3, build 4debf41
ubuntu@ip-172-31-46-65:~$ 
```

i-08aa68f0014c433f6 (tanmayvaij-ec2docker)

PublicIPs: 13.201.50.149 PrivateIPs: 172.31.46.65

Step 7: Create a ‘Dockerfile’ and add the following text in it and save the file

FROM ubuntu:latest
WORKDIR /app
RUN apt update -y
CMD [“echo”, “Hello from Tanmay Vaij”]

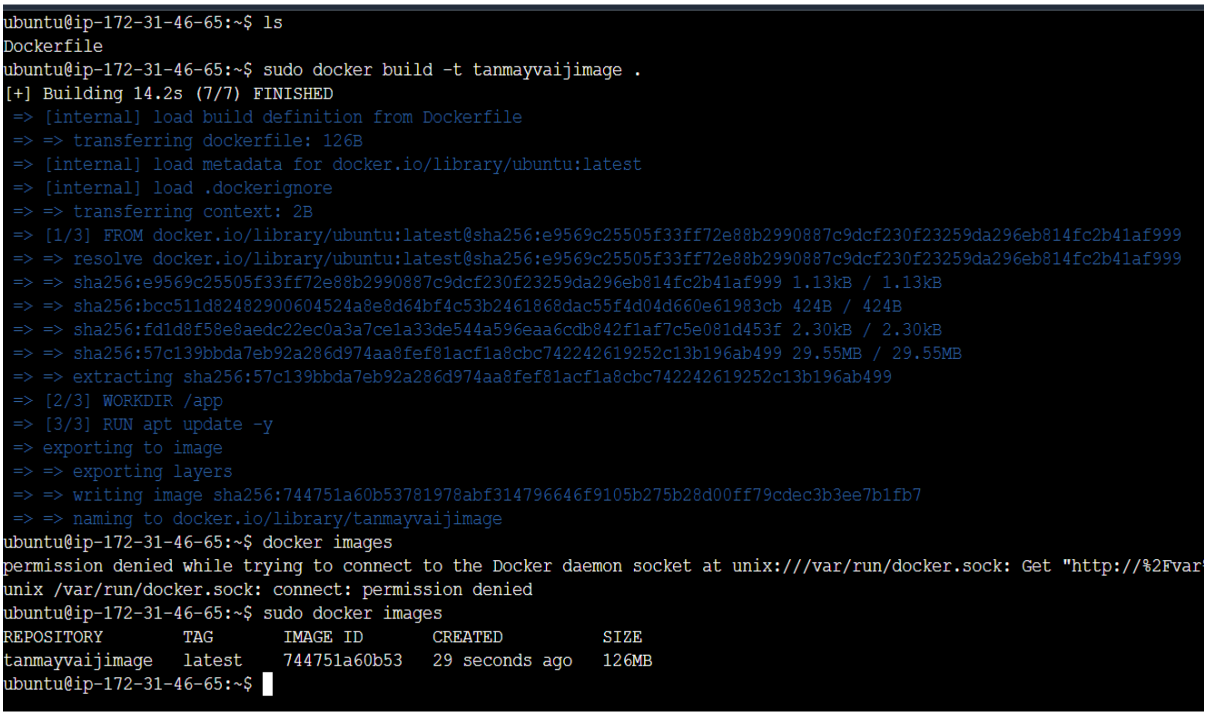


Instance details | EC2 | ap-south-1
EC2 Instance Connect | ap-sou
ap-south-1.console.a
aws
Services
Search

```
  GNU nano 6.2
FROM ubuntu:latest
WORKDIR /app
RUN apt update -y
CMD ["echo", "Hello from Tanmay Vaij"]
```

Step 8: After saving Dockerfile, build image using the following command

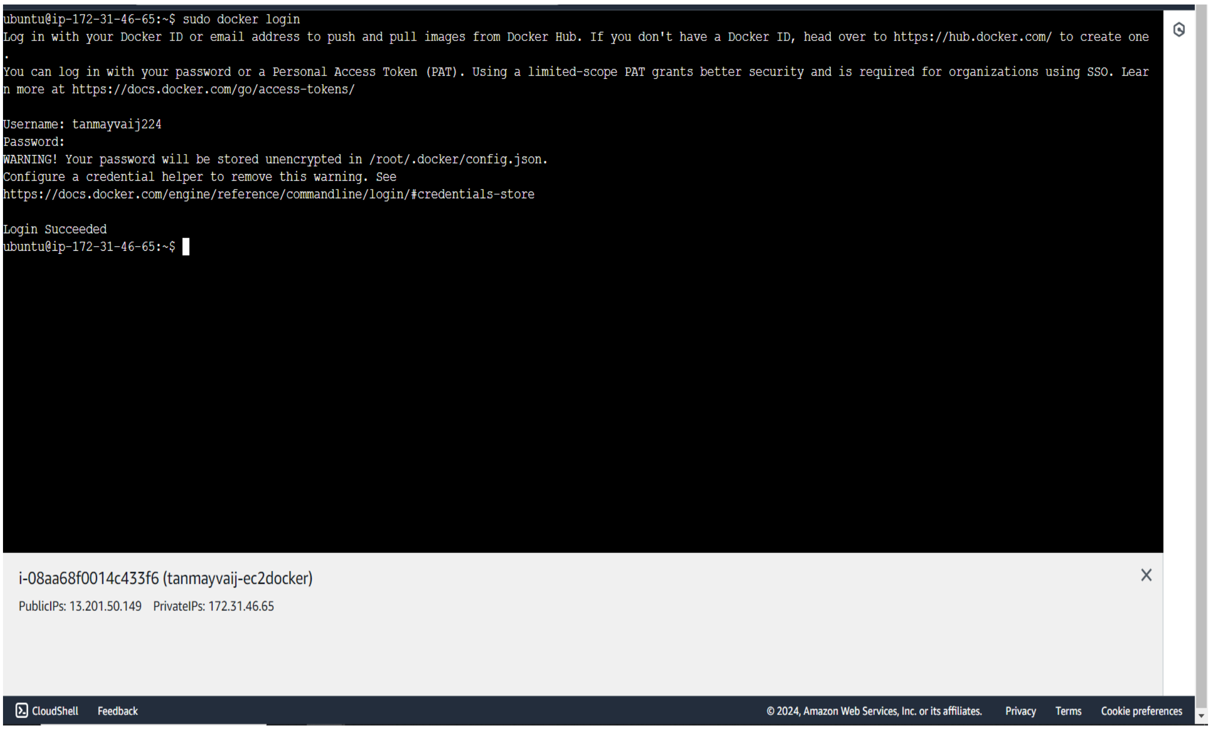
Command: $ sudo docker build -t tanmayvaijimage .



```
ubuntu@ip-172-31-46-65:~$ ls
Dockerfile
ubuntu@ip-172-31-46-65:~$ sudo docker build -t tanmayvaijimage .
[+] Building 14.2s (7/7) FINISHED
 => [internal] load build definition from Dockerfile
 => => transferring dockerfile: 126B
 => [internal] load metadata for docker.io/library/ubuntu:latest
 => [internal] load .dockerignore
 => => transferring context: 2B
 => [1/3] FROM docker.io/library/ubuntu:latest@sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
 => => resolve docker.io/library/ubuntu:latest@sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999
 => => sha256:e9569c25505f33ff72e88b2990887c9dcf230f23259da296eb814fc2b41af999 1.13kB / 1.13kB
 => => sha256:bcc511d82482900604524a8e8d64bf4c53b2461868dac55f4d04d660e61983cb 424B / 424B
 => => sha256:fd1d8f58e8aedc22ec0a3a7ce1a33de544a596eaa6cdb842f1af7c5e081d453f 2.30kB / 2.30kB
 => => sha256:57c139bbda7eb92a286d974aa8fef81acf1a8cbc742242619252c13b196ab499 29.55MB / 29.55MB
 => => extracting sha256:57c139bbda7eb92a286d974aa8fef81acf1a8cbc742242619252c13b196ab499
 => [2/3] WORKDIR /app
 => [3/3] RUN apt update -y
 => exporting to image
 => => exporting layers
 => => writing image sha256:744751a60b53781978abf314796646f9105b275b28d00ff79cdec3b3ee7b1fb7
 => => naming to docker.io/library/tanmayvaijimage
ubuntu@ip-172-31-46-65:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar
unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-46-65:~$ sudo docker images
REPOSITORY        TAG       IMAGE ID       CREATED          SIZE
tanmayvaijimage   latest    744751a60b53   29 seconds ago   126MB
ubuntu@ip-172-31-46-65:~$
```

Step 9: Login in the DockerHub account via command line.

Command: $ sudo docker login



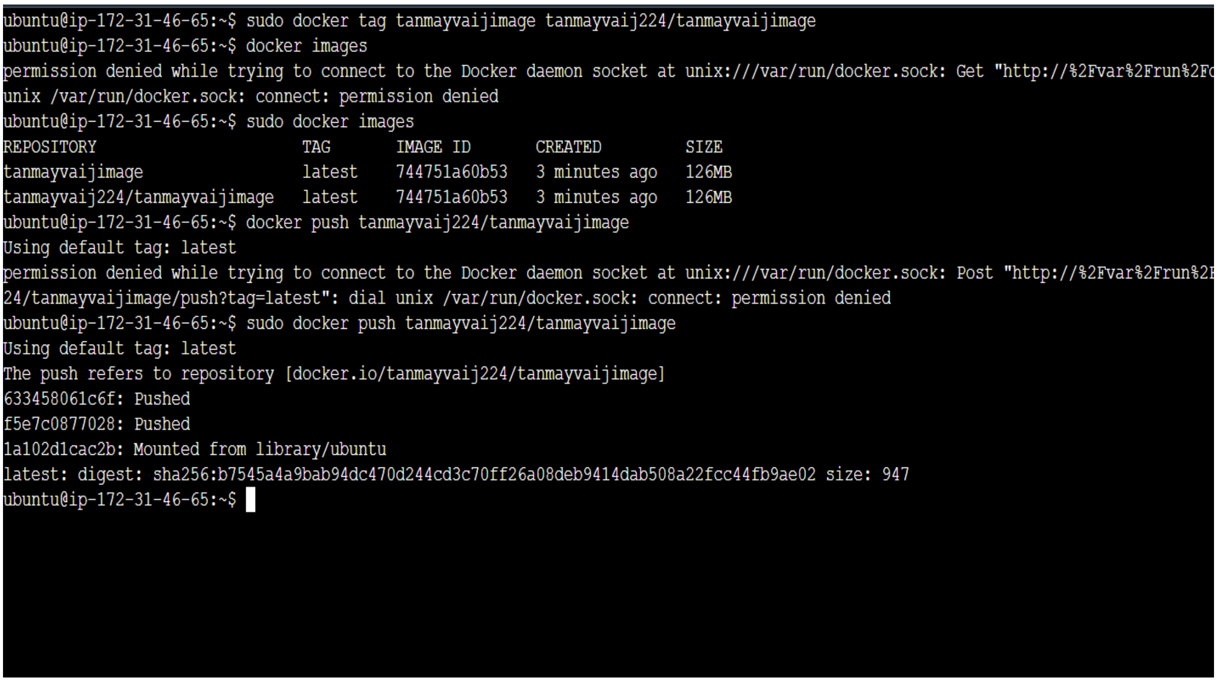
```
ubuntu@ip-172-31-46-65:~$ sudo docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one
.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Lear
n more at https://docs.docker.com/go/access-tokens/

Username: tanmayvaij224
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
ubuntu@ip-172-31-46-65:~$
```

i-08aa68f0014c433f6 (tanmayvaij-ec2docker)

PublicIPs: 13.201.50.149   PrivateIPs: 172.31.46.65

Step 10: Tag the created image with docker hub username as prefix, and push it docker hub

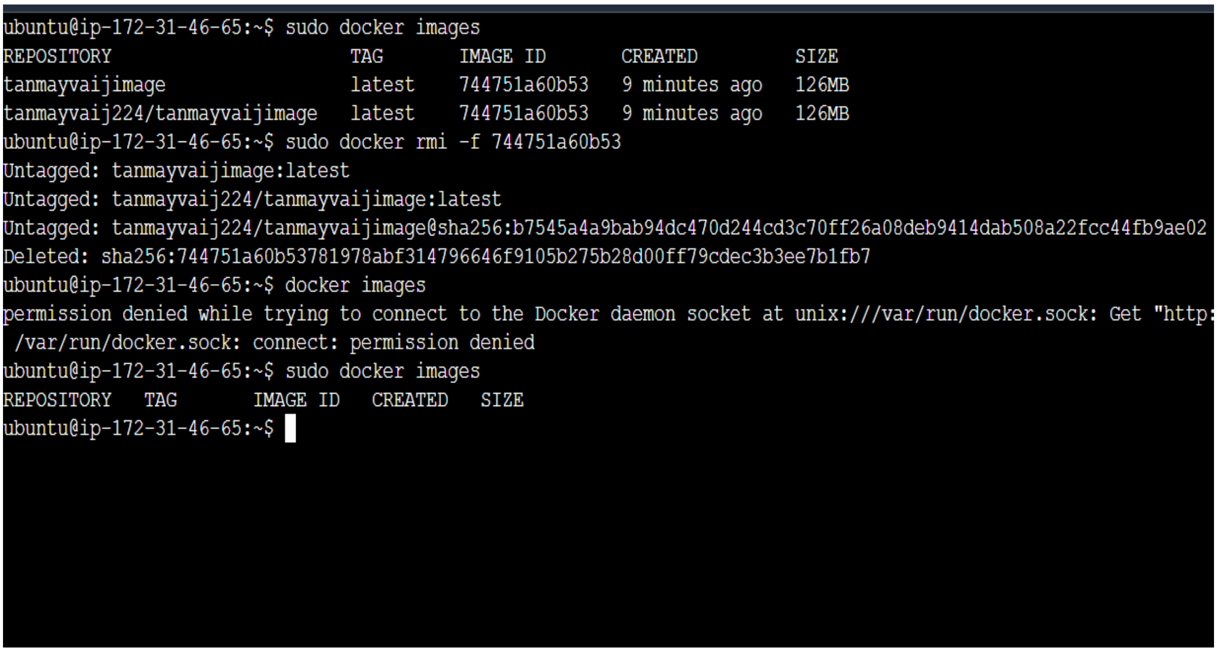
Command:
$ sudo docker tag tanmayvaijimage tanmayvaij224/tanmayvaijimage
$ sudo docker push tanmayvaij224/tanmayvaijimage



```
ubuntu@ip-172-31-46-65:~$ sudo docker tag tanmayvaijimage tanmayvaij224/tanmayvaijimage
ubuntu@ip-172-31-46-65:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://%2Fvar%2Frun%2F
unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-46-65:~$ sudo docker images
REPOSITORY                      TAG       IMAGE ID       CREATED         SIZE
tanmayvaijimage                 latest    744751a60b53   3 minutes ago   126MB
tanmayvaij224/tanmayvaijimage   latest    744751a60b53   3 minutes ago   126MB
ubuntu@ip-172-31-46-65:~$ docker push tanmayvaij224/tanmayvaijimage
Using default tag: latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2F
24/tanmayvaijimage/push?tag=latest": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-46-65:~$ sudo docker push tanmayvaij224/tanmayvaijimage
Using default tag: latest
The push refers to repository [docker.io/tanmayvaij224/tanmayvaijimage]
633458061c6f: Pushed
f5e7c0877028: Pushed
1a102d1cac2b: Mounted from library/ubuntu
latest: digest: sha256:b7545a4a9bab94dc470d244cd3c70ff26a08deb9414dab508a22fcc44fb9ae02 size: 947
ubuntu@ip-172-31-46-65:~$
```

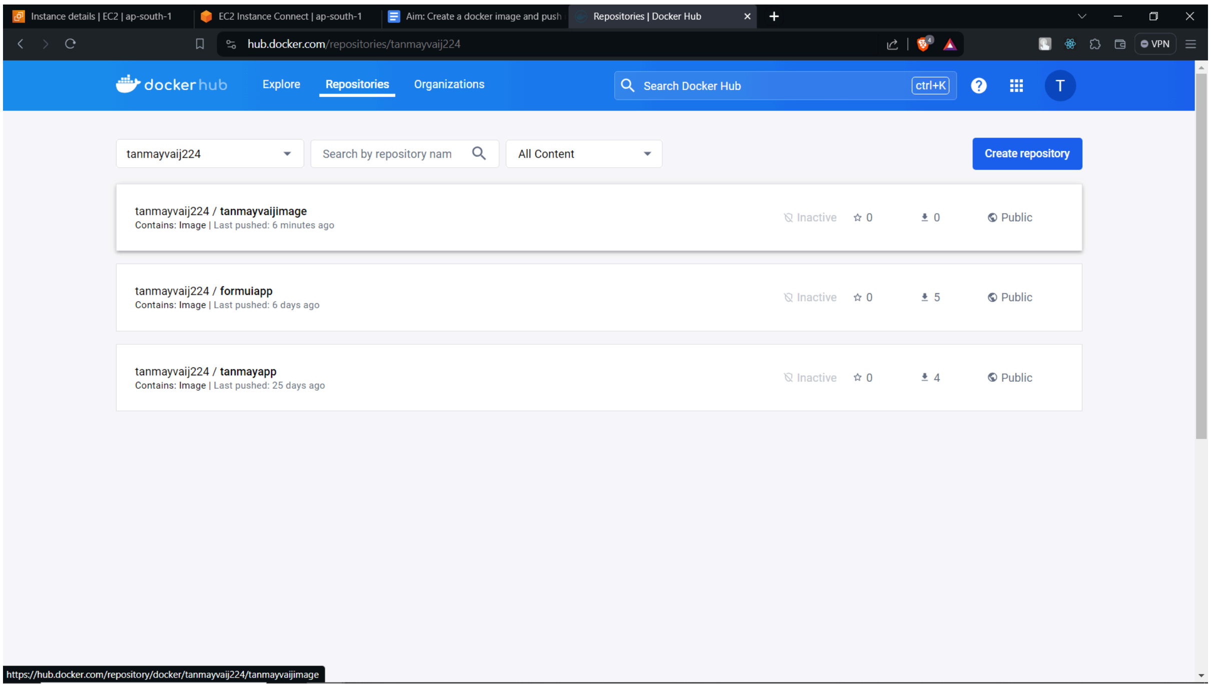
Step 11: Remove all the local images

Command: $ sudo docker rmi -f <image_id>



```
ubuntu@ip-172-31-46-65:~$ sudo docker images
REPOSITORY                      TAG       IMAGE ID       CREATED         SIZE
tanmayvaijimage                 latest    744751a60b53   9 minutes ago   126MB
tanmayvaij224/tanmayvaijimage   latest    744751a60b53   9 minutes ago   126MB
ubuntu@ip-172-31-46-65:~$ sudo docker rmi -f 744751a60b53
Untagged: tanmayvaijimage:latest
Untagged: tanmayvaij224/tanmayvaijimage:latest
Untagged: tanmayvaij224/tanmayvaijimage@sha256:b7545a4a9bab94dc470d244cd3c70ff26a08deb9414dab508a22fcc44fb9ae02
Deleted: sha256:744751a60b53781978abf314796646f9105b275b28d00ff79cdec3b3ee7b1fb7
ubuntu@ip-172-31-46-65:~$ docker images
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http:
 /var/run/docker.sock: connect: permission denied
ubuntu@ip-172-31-46-65:~$ sudo docker images
REPOSITORY   TAG       IMAGE ID   CREATED   SIZE
ubuntu@ip-172-31-46-65:~$
```

Step 12: Check the dockerhub account, the pushed image should appear there


Instance details | EC2 | ap-south-1
EC2 Instance Connect | ap-south-1
Aim: Create a docker image and push
Repositories | Docker Hub
hub.docker.com/repositories/tanmayvaij224
docker hub
Explore
Repositories
Organizations
Search Docker Hub
ctrl+K
T
tanmayvaij224
Search by repository nam
All Content
Create repository
https://hub.docker.com/repository/docker/tanmayvaij224/tanmayvaijimage

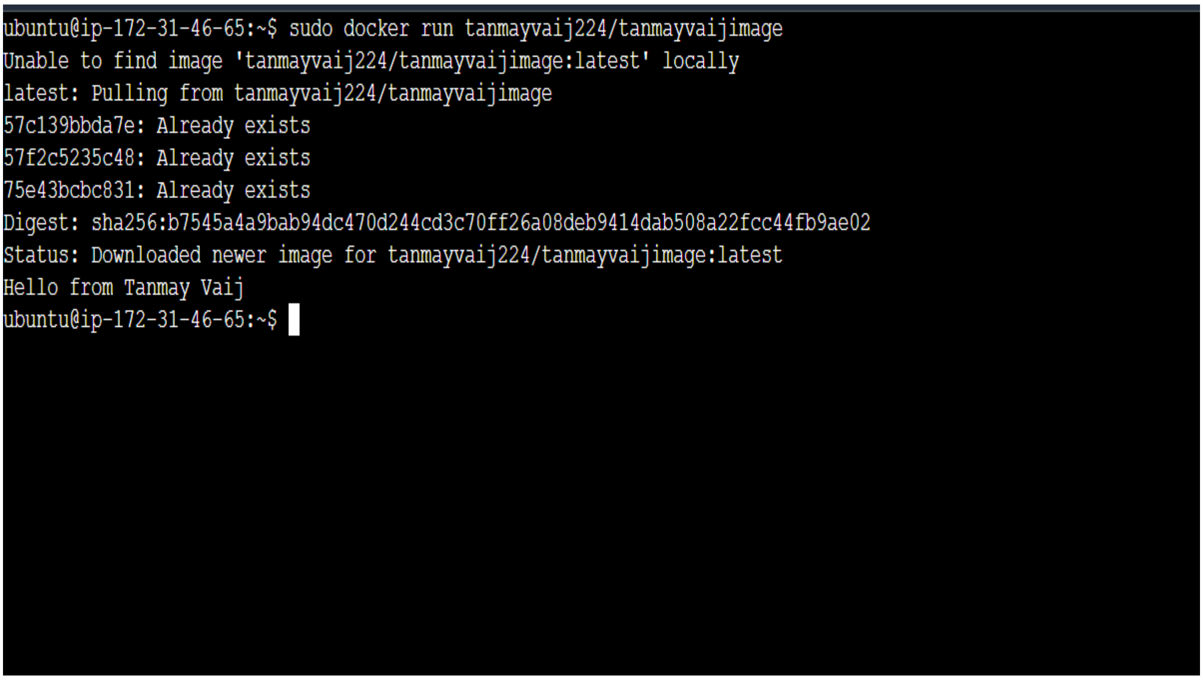
tanmayvaij224 / **tanmayvaijimage**
Contains: Image | Last pushed: 6 minutes ago — Inactive ☆ 0 ⬇ 0 Public

tanmayvaij224 / **formuiapp**
Contains: Image | Last pushed: 6 days ago — Inactive ☆ 0 ⬇ 5 Public

tanmayvaij224 / **tanmayapp**
Contains: Image | Last pushed: 25 days ago — Inactive ☆ 0 ⬇ 4 Public

Step 13: Run your pushed image on instance

$ sudo docker run tanmayvaij/tanmayvaijimage



```
ubuntu@ip-172-31-46-65:~$ sudo docker run tanmayvaij224/tanmayvaijimage
Unable to find image 'tanmayvaij224/tanmayvaijimage:latest' locally
latest: Pulling from tanmayvaij224/tanmayvaijimage
57c139bbda7e: Already exists
57f2c5235c48: Already exists
75e43bcbc831: Already exists
Digest: sha256:b7545a4a9bab94dc470d244cd3c70ff26a08deb9414dab508a22fcc44fb9ae02
Status: Downloaded newer image for tanmayvaij224/tanmayvaijimage:latest
Hello from Tanmay Vaij
ubuntu@ip-172-31-46-65:~$ 
```