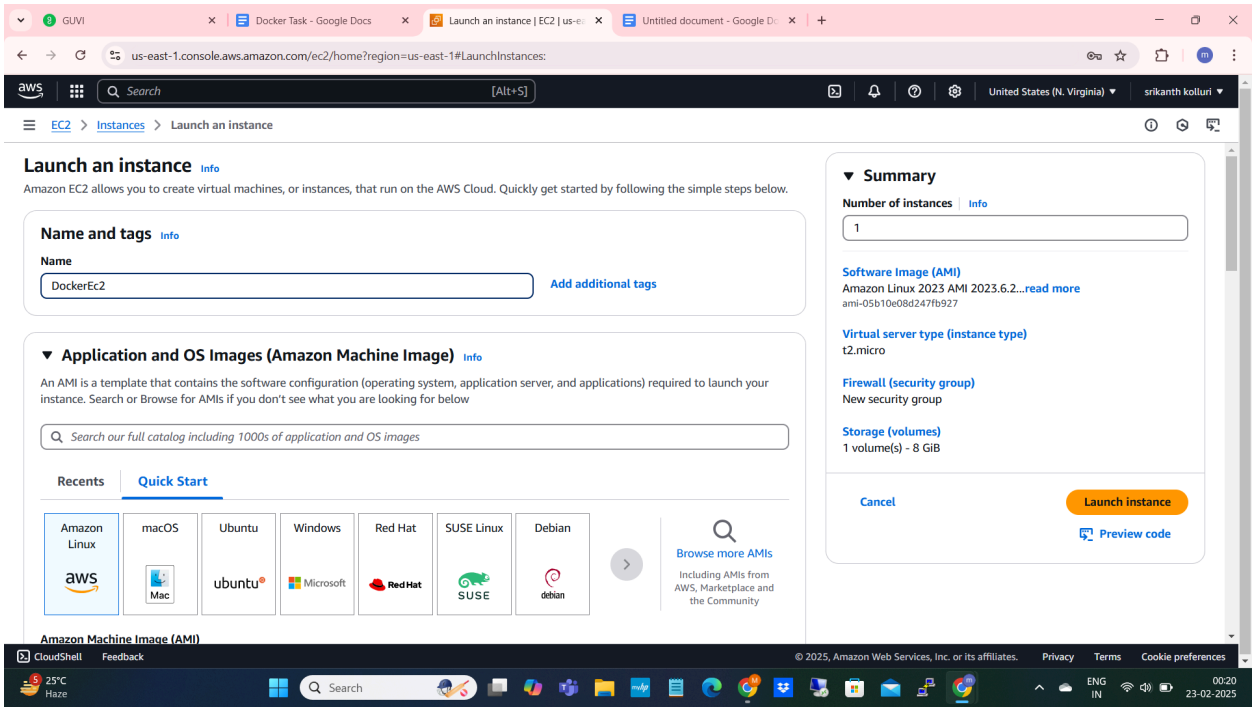
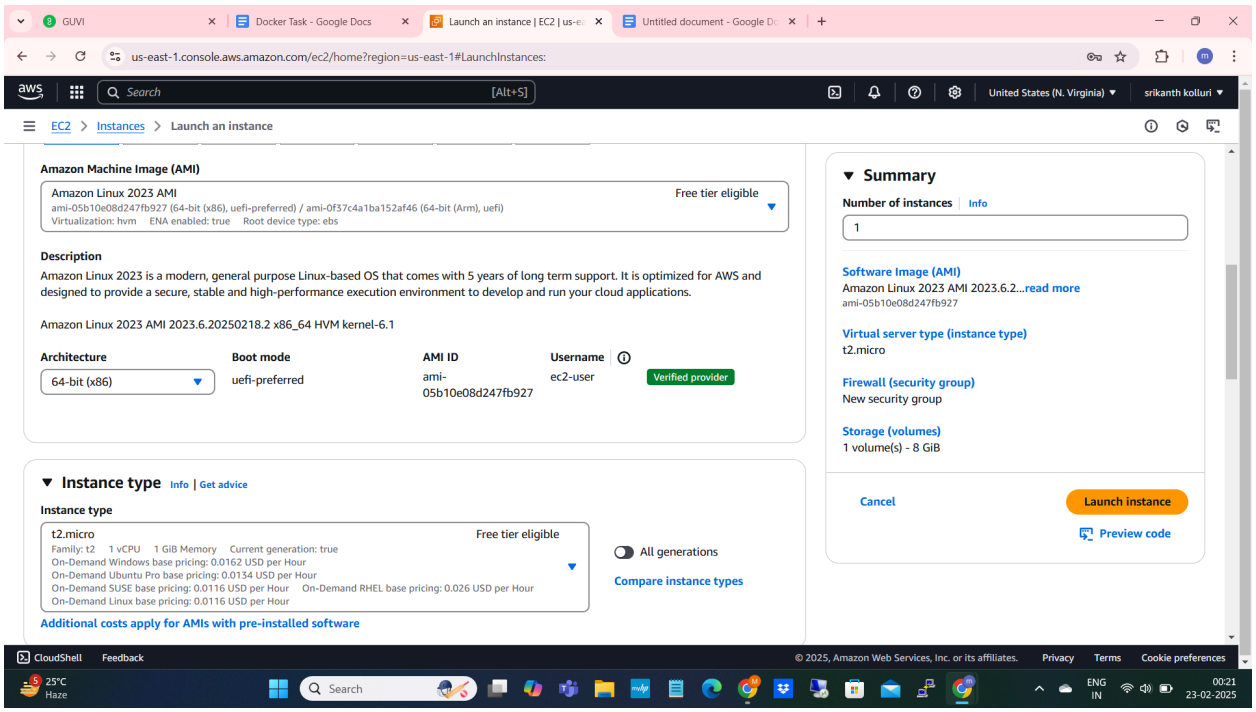
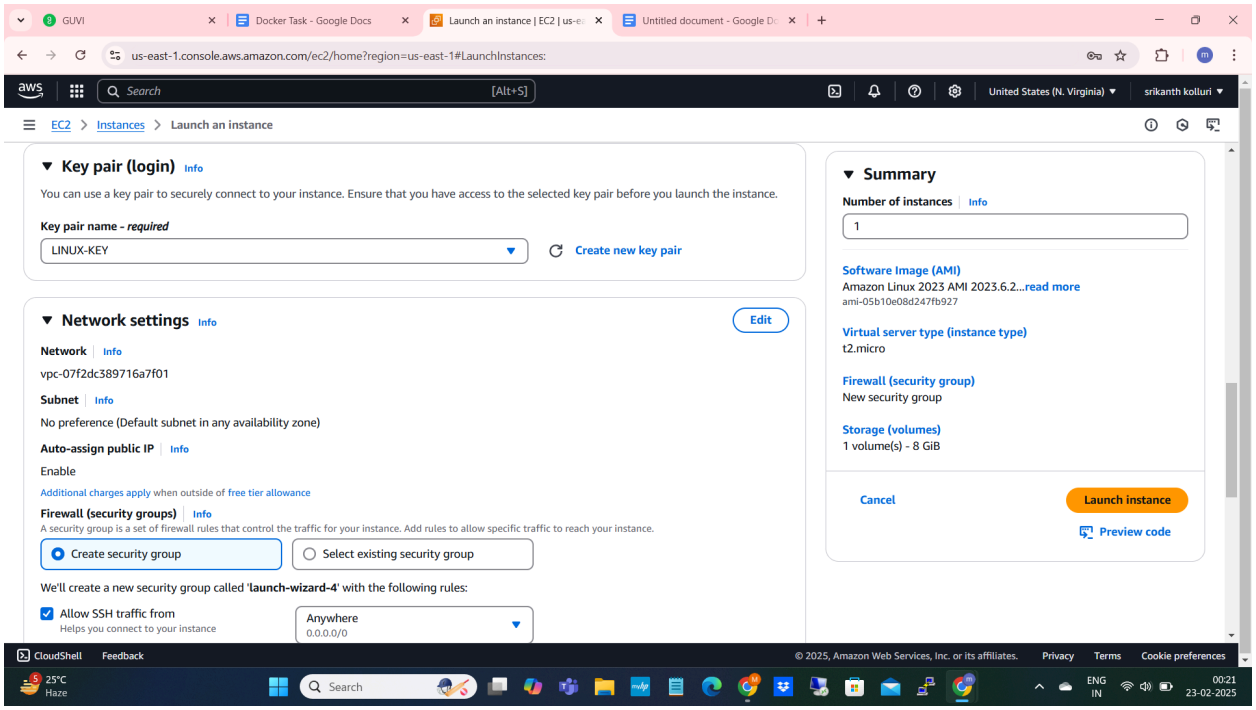
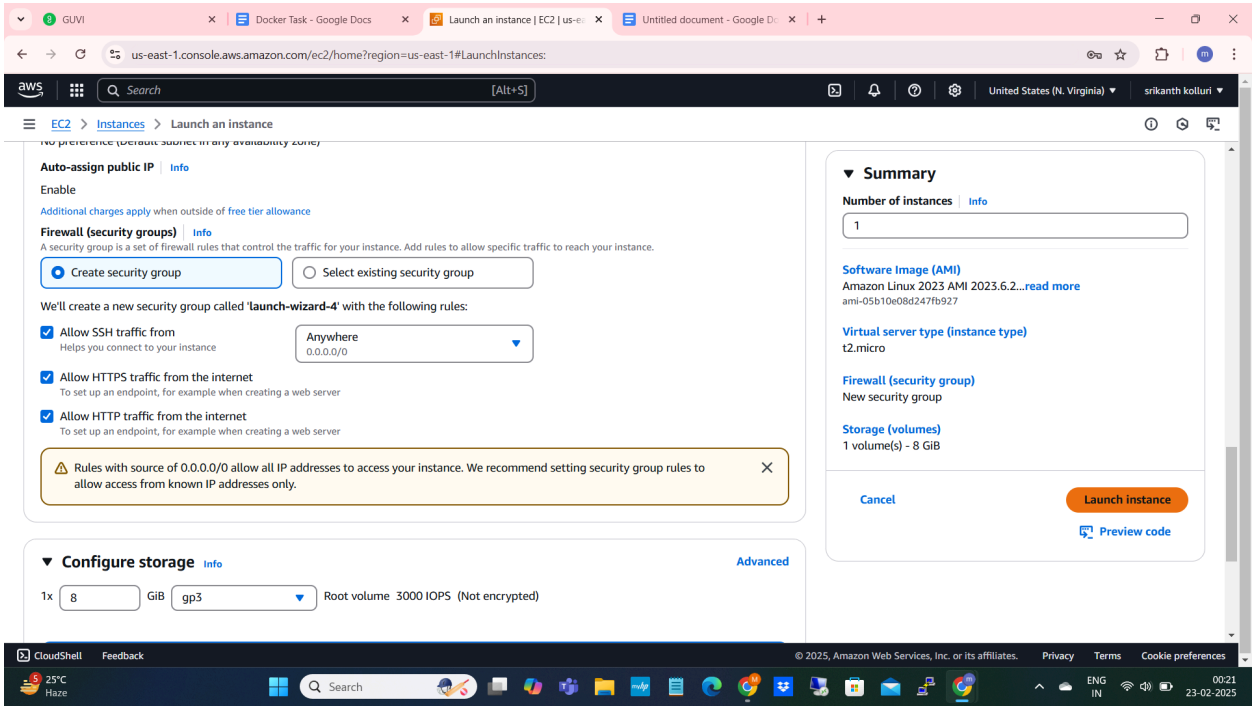
Docker Task

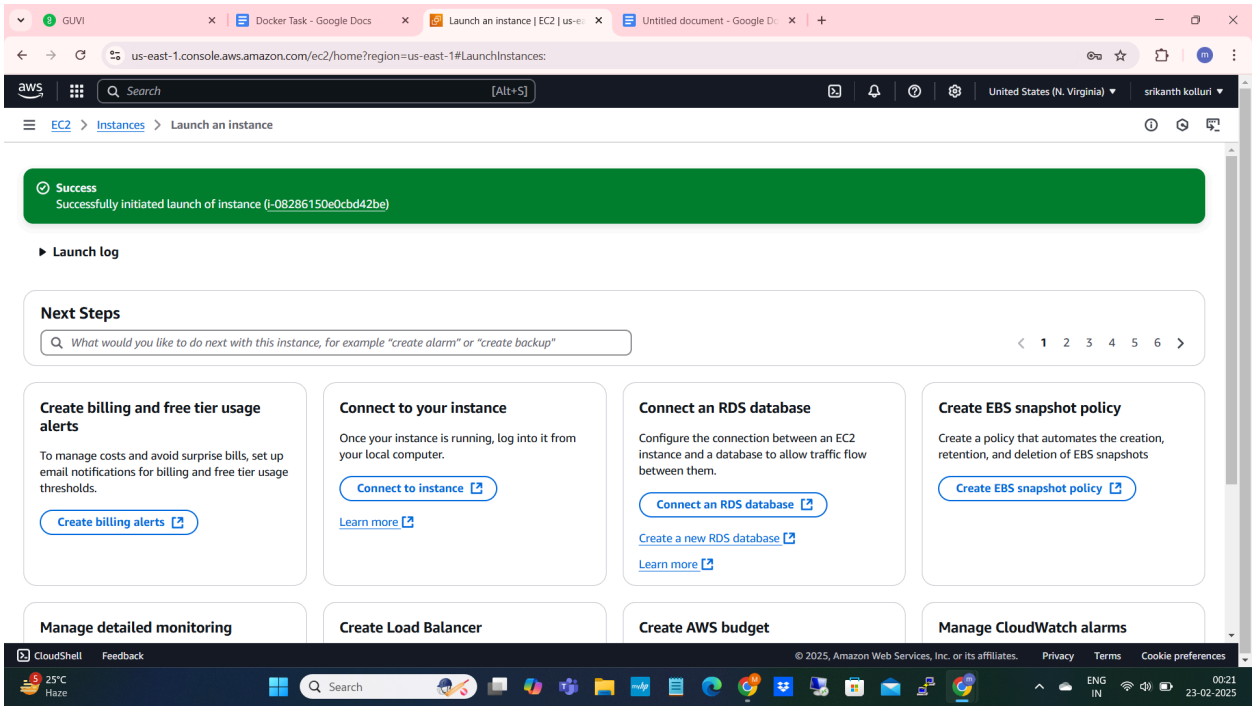
Launching Ec2 Instance

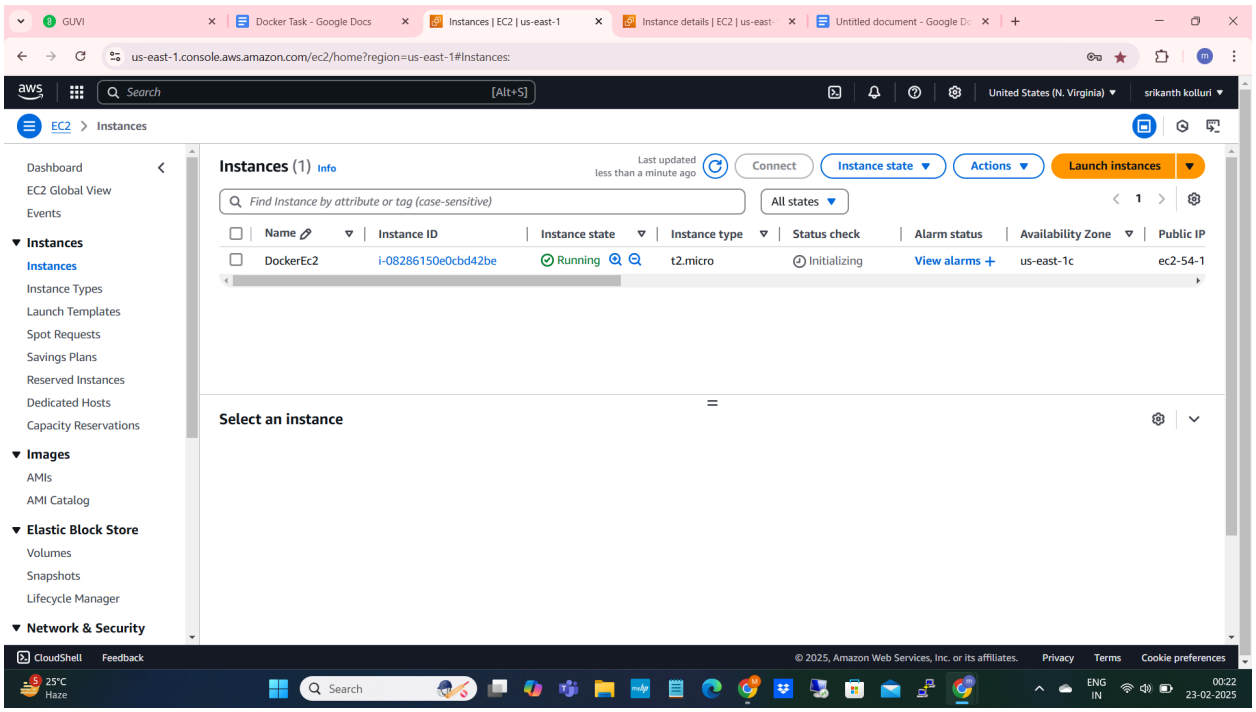


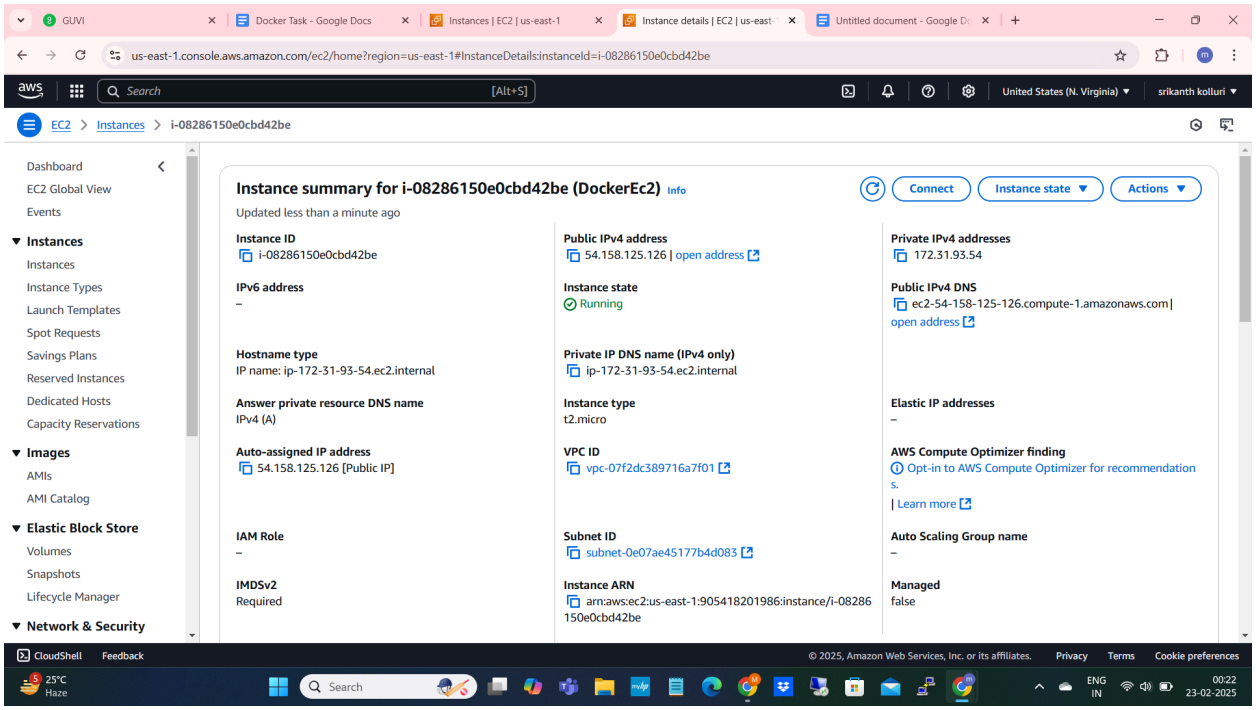


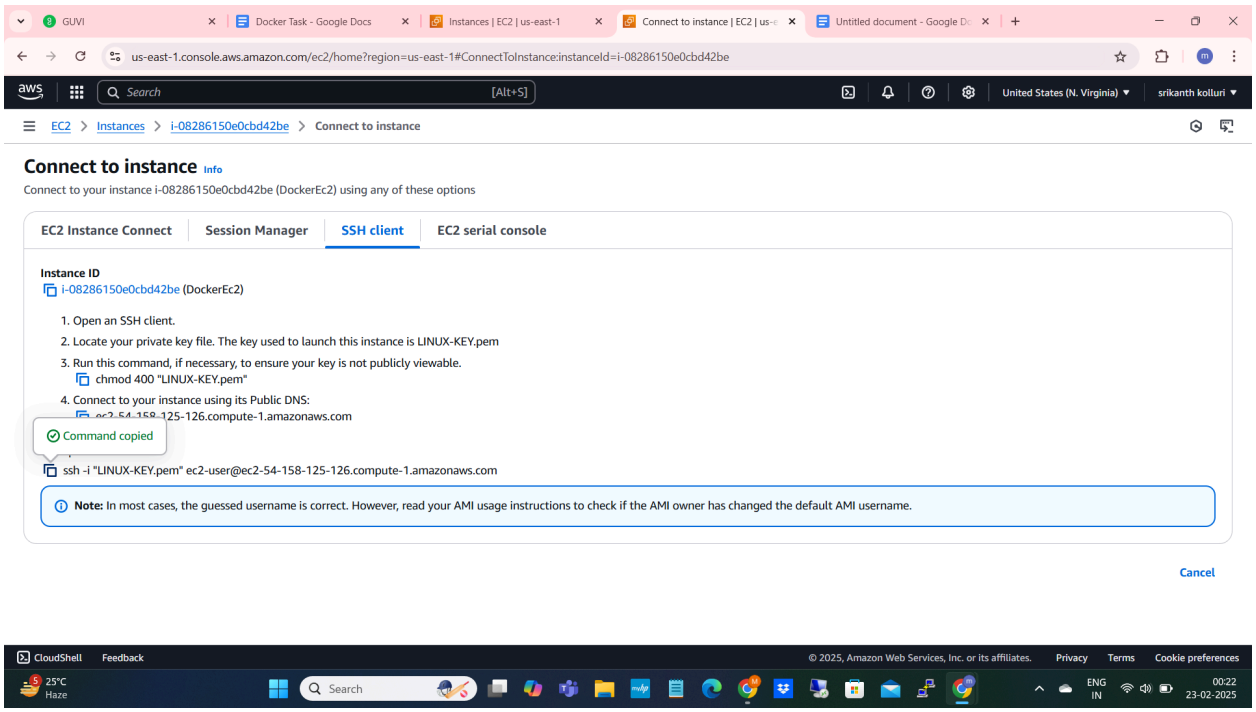

GUVI
Docker Task - Google Docs
Launch an instance | EC2 | us-ea
Untitled document - Google Do
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances > Launch an instance
Key pair (login) Info
You can use 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
LINUX-KEY
Create new key pair
Network settings Info
Edit
Network Info
vpc-07f2dc389716a7f01
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
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
We'll create a new security group called 'launch-wizard-4' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-05b10e08d247fb927
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
00:21
23-02-2025


GUVI
Docker Task - Google Docs
Launch an instance | EC2 | us-ea
Untitled document - Google Do
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances > Launch an instance
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
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
We'll create a new security group called 'launch-wizard-4' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.
Configure storage Info
Advanced
1x
8
GiB
gp3
Root volume 3000 IOPS (Not encrypted)
Summary
Number of instances Info
1
Software Image (AMI)
Amazon Linux 2023 AMI 2023.6.2...read more
ami-05b10e08d247fb927
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Preview code
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
00:21
23-02-2025

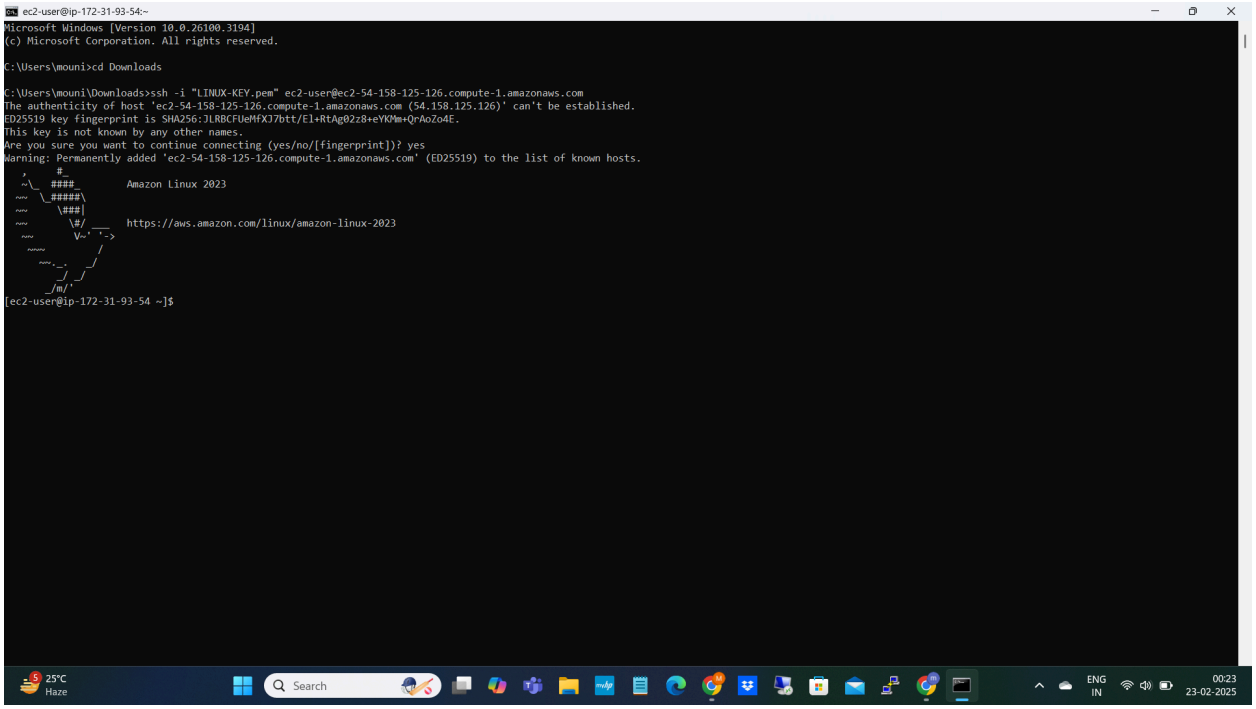

GUVI
Docker Task - Google Docs
Launch an instance | EC2 | us-e
Untitled document - Google D
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchInstances:
Search
[Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances > Launch an instance
Success
Successfully initiated launch of instance (i-08286150e0cbd42be)
Launch log
Next Steps
What would you like to do next with this instance, for example "create alarm" or "create backup"
1 2 3 4 5 6
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts
Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more
Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more
Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy
Manage detailed monitoring
Create Load Balancer
Create AWS budget
Manage CloudWatch alarms
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
00:21
23-02-2025


GUVI
Docker Task - Google Docs
Instances | EC2 | us-east-1
Instance details | EC2 | us-east-
Untitled document - Google D
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:
Search
[Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instances (1) Info
Last updated less than a minute ago
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
1
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IP
DockerEc2
i-08286150e0cbd42be
Running
t2.micro
Initializing
View alarms +
us-east-1c
ec2-54-1
Select an instance
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C
Haze
Search
ENG
IN
00:22
23-02-2025


GUVI
Docker Task - Google Docs
Instances | EC2 | us-east-1
Instance details | EC2 | us-east-1
Untitled document - Google Do
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#InstanceDetails:instanceId=i-08286150e0cbd42be
Search [Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances > i-08286150e0cbd42be
Dashboard
EC2 Global View
Events
Instances
Instances
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances
Dedicated Hosts
Capacity Reservations
Images
AMIs
AMI Catalog
Elastic Block Store
Volumes
Snapshots
Lifecycle Manager
Network & Security
Instance summary for i-08286150e0cbd42be (DockerEc2) Info
Updated less than a minute ago
Connect
Instance state
Actions
Instance ID
i-08286150e0cbd42be
IPv6 address
–
Hostname type
IP name: ip-172-31-93-54.ec2.internal
Answer private resource DNS name
IPv4 (A)
Auto-assigned IP address
54.158.125.126 [Public IP]
IAM Role
–
IMDSv2
Required
Public IPv4 address
54.158.125.126 | open address
Instance state
Running
Private IP DNS name (IPv4 only)
ip-172-31-93-54.ec2.internal
Instance type
t2.micro
VPC ID
vpc-07f2dc389716a7f01
Subnet ID
subnet-0e07ae45177b4d083
Instance ARN
arn:aws:ec2:us-east-1:905418201986:instance/i-08286150e0cbd42be
Private IPv4 addresses
172.31.93.54
Public IPv4 DNS
ec2-54-158-125-126.compute-1.amazonaws.com | open address
Elastic IP addresses
–
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
Auto Scaling Group name
–
Managed
false
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C Haze
Search
ENG IN
00:22 23-02-2025

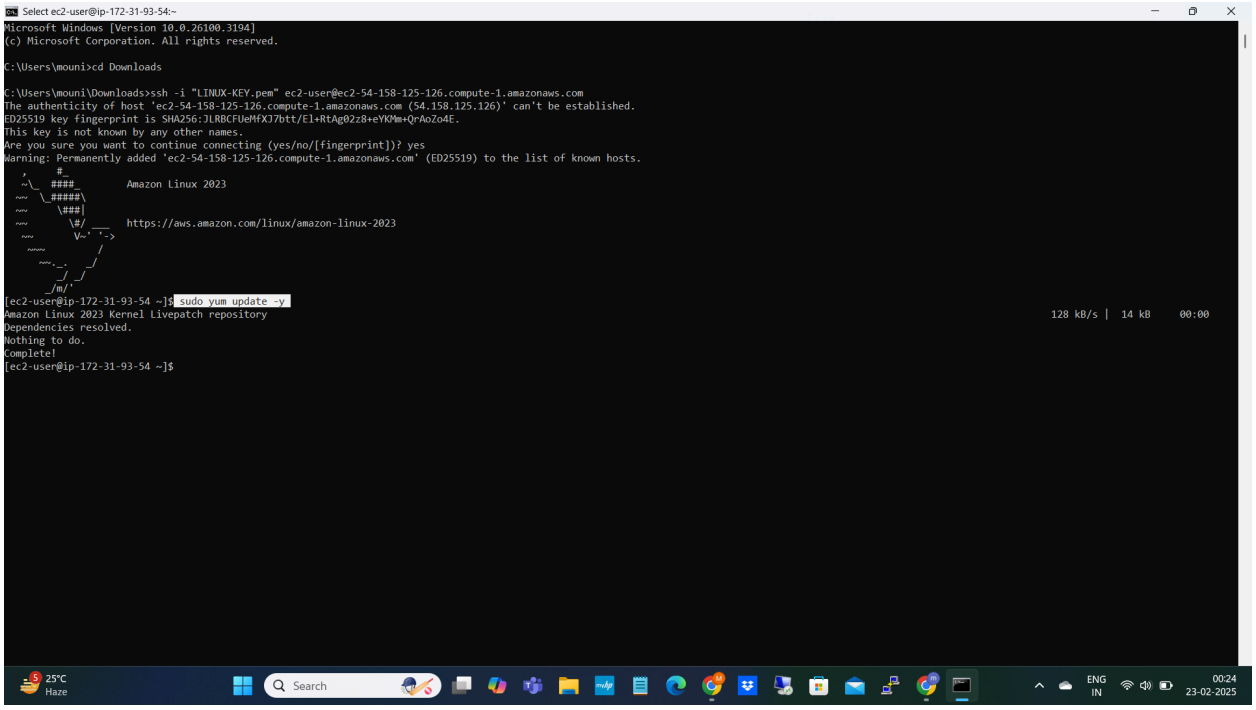

GUVI
Docker Task - Google Docs
Instances | EC2 | us-east-1
Connect to instance | EC2 | us-e
Untitled document - Google Do
us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#ConnectToInstance:instanceId=i-08286150e0cbd42be
Search [Alt+S]
United States (N. Virginia)
srikanth kolluri
EC2 > Instances > i-08286150e0cbd42be > Connect to instance
Connect to instance Info
Connect to your instance i-08286150e0cbd42be (DockerEc2) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-08286150e0cbd42be (DockerEc2)
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is LINUX-KEY.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "LINUX-KEY.pem"
4. Connect to your instance using its Public DNS:
Command copied
ssh -i "LINUX-KEY.pem" ec2-user@ec2-54-158-125-126.compute-1.amazonaws.com
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
CloudShell
Feedback
© 2025, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences
25°C Haze
Search
ENG IN
00:22 23-02-2025

Connecting to the Ec2



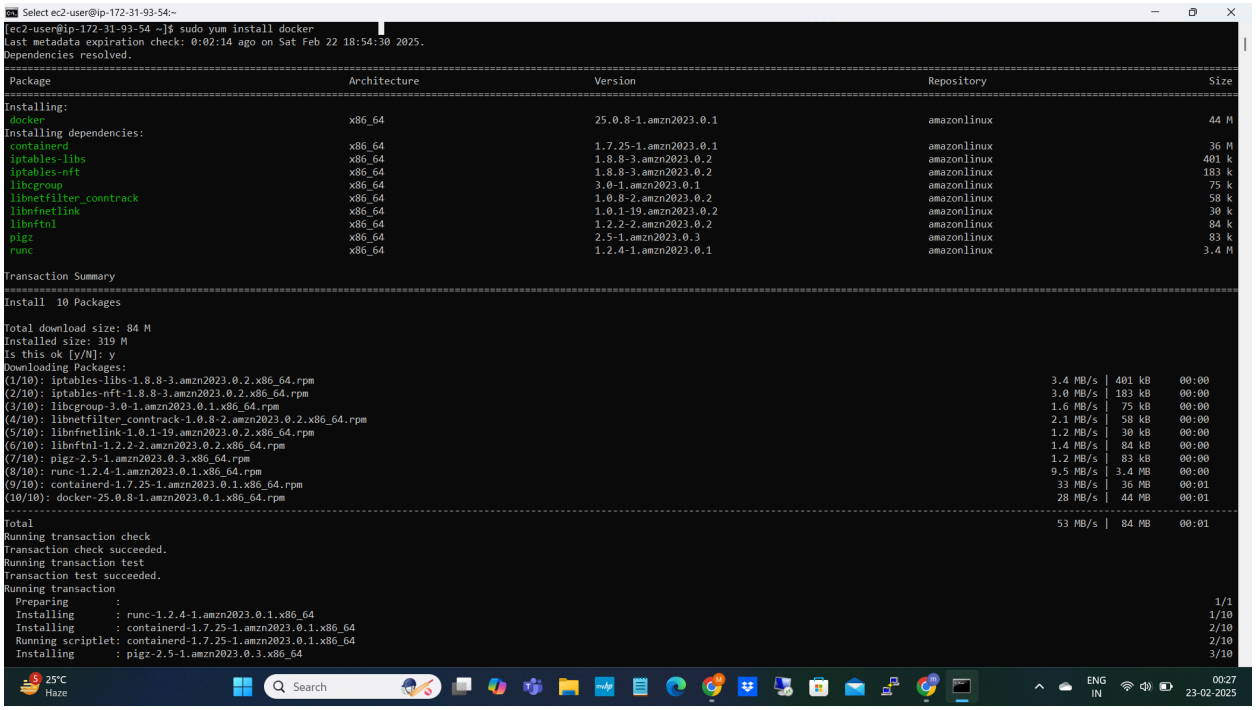
Update packages:

$sudo yum update -y

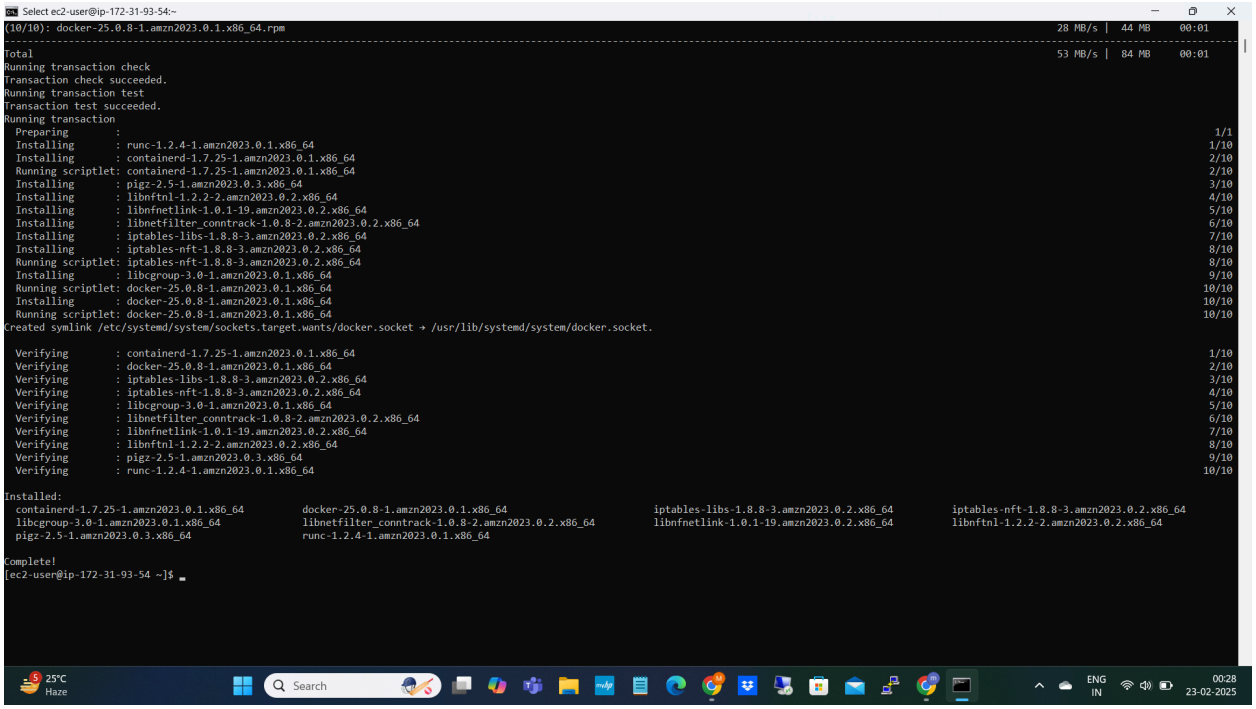


Install Docker:

$ sudo yum install docker



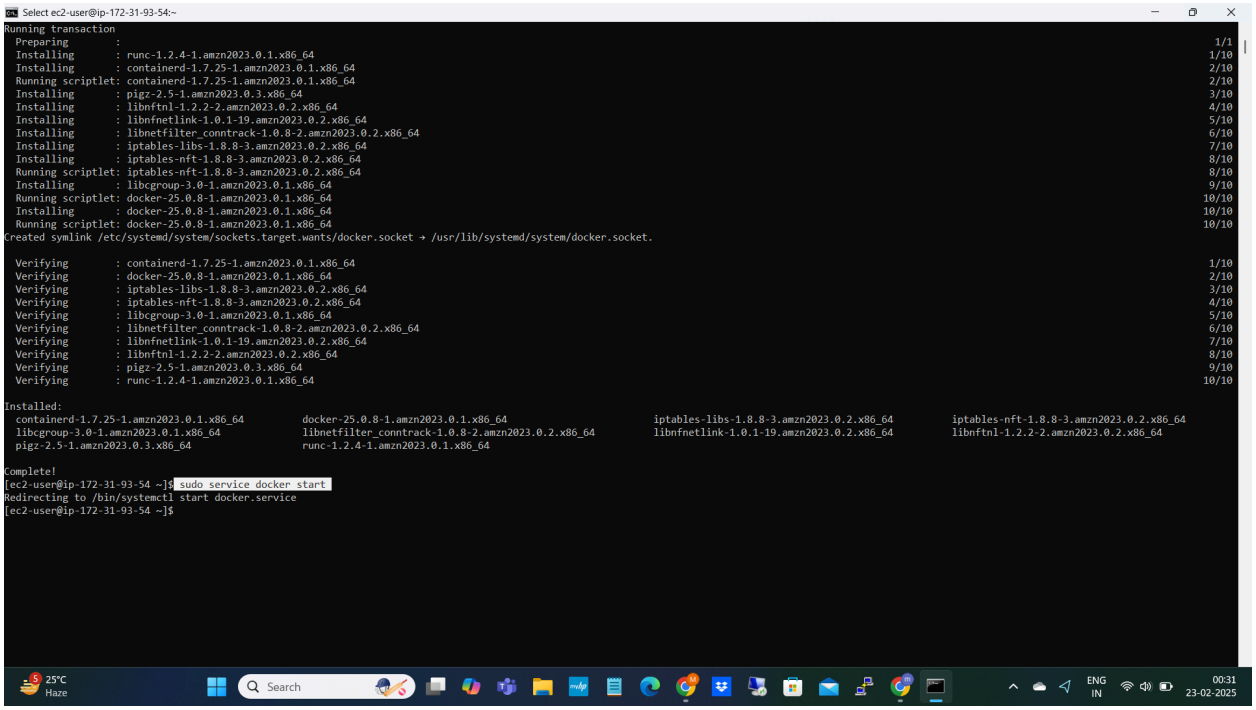
```
Select ec2-user@ip-172-31-93-54:~
[ec2-user@ip-172-31-93-54 ~]$ sudo yum install docker
Last metadata expiration check: 0:02:14 ago on Sat Feb 22 18:54:30 2025.
Dependencies resolved.
================================================================================
 Package                 Architecture  Version                  Repository  Size
================================================================================
Installing:
 docker                  x86_64        25.0.8-1.amzn2023.0.1    amazonlinux 44 M
Installing dependencies:
 containerd              x86_64        1.7.25-1.amzn2023.0.1    amazonlinux 36 M
 iptables-libs           x86_64        1.8.8-3.amzn2023.0.2     amazonlinux 401 k
 iptables-nft            x86_64        1.8.8-3.amzn2023.0.2     amazonlinux 183 k
 libcgroup               x86_64        3.0-1.amzn2023.0.1       amazonlinux 75 k
 libnetfilter_conntrack  x86_64        1.0.8-2.amzn2023.0.2     amazonlinux 58 k
 libnfnetlink            x86_64        1.0.1-19.amzn2023.0.2    amazonlinux 30 k
 libnftnl                x86_64        1.2.2-2.amzn2023.0.2     amazonlinux 84 k
 pigz                    x86_64        2.5-1.amzn2023.0.3       amazonlinux 83 k
 runc                    x86_64        1.2.4-1.amzn2023.0.1     amazonlinux 3.4 M

Transaction Summary
================================================================================
Install  10 Packages

Total download size: 84 M
Installed size: 319 M
Is this ok [y/N]: y
Downloading Packages:
(1/10): iptables-libs-1.8.8-3.amzn2023.0.2.x86_64.rpm   3.4 MB/s | 401 kB  00:00
(2/10): iptables-nft-1.8.8-3.amzn2023.0.2.x86_64.rpm    3.0 MB/s | 183 kB  00:00
(3/10): libcgroup-3.0-1.amzn2023.0.1.x86_64.rpm         1.6 MB/s |  75 kB  00:00
(4/10): libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64.rpm 2.1 MB/s |  58 kB  00:00
(5/10): libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64.rpm   1.2 MB/s |  30 kB  00:00
(6/10): libnftnl-1.2.2-2.amzn2023.0.2.x86_64.rpm        1.4 MB/s |  84 kB  00:00
(7/10): pigz-2.5-1.amzn2023.0.3.x86_64.rpm              1.2 MB/s |  83 kB  00:00
(8/10): runc-1.2.4-1.amzn2023.0.1.x86_64.rpm            9.5 MB/s | 3.4 MB  00:00
(9/10): containerd-1.7.25-1.amzn2023.0.1.x86_64.rpm      33 MB/s |  36 MB  00:01
(10/10): docker-25.0.8-1.amzn2023.0.1.x86_64.rpm         28 MB/s |  44 MB  00:01
--------------------------------------------------------------------------------
Total                                                    53 MB/s |  84 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : runc-1.2.4-1.amzn2023.0.1.x86_64                       1/10
  Installing       : containerd-1.7.25-1.amzn2023.0.1.x86_64                2/10
  Running scriptlet: containerd-1.7.25-1.amzn2023.0.1.x86_64                2/10
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                         3/10
```

```
Select ec2-user@ip-172-31-93-54:~
(10/10): docker-25.0.8-1.amzn2023.0.1.x86_64.rpm         28 MB/s |  44 MB  00:01
--------------------------------------------------------------------------------
Total                                                    53 MB/s |  84 MB  00:01
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing        :                                                        1/1
  Installing       : runc-1.2.4-1.amzn2023.0.1.x86_64                       1/10
  Installing       : containerd-1.7.25-1.amzn2023.0.1.x86_64                2/10
  Running scriptlet: containerd-1.7.25-1.amzn2023.0.1.x86_64                2/10
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                         3/10
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                   4/10
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64              5/10
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64     6/10
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64              7/10
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64               8/10
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64               8/10
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                    9/10
  Running scriptlet: docker-25.0.8-1.amzn2023.0.1.x86_64                    10/10
  Installing       : docker-25.0.8-1.amzn2023.0.1.x86_64                    10/10
  Running scriptlet: docker-25.0.8-1.amzn2023.0.1.x86_64                    10/10
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Verifying        : containerd-1.7.25-1.amzn2023.0.1.x86_64                1/10
  Verifying        : docker-25.0.8-1.amzn2023.0.1.x86_64                    2/10
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64              3/10
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64               4/10
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                    5/10
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64     6/10
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64              7/10
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                   8/10
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                         9/10
  Verifying        : runc-1.2.4-1.amzn2023.0.1.x86_64                      10/10

Installed:
  containerd-1.7.25-1.amzn2023.0.1.x86_64   docker-25.0.8-1.amzn2023.0.1.x86_64   iptables-libs-1.8.8-3.amzn2023.0.2.x86_64   iptables-nft-1.8.8-3.amzn2023.0.2.x86_64
  libcgroup-3.0-1.amzn2023.0.1.x86_64       libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64   libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64   libnftnl-1.2.2-2.amzn2023.0.2.x86_64
  pigz-2.5-1.amzn2023.0.3.x86_64            runc-1.2.4-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-93-54 ~]$
```

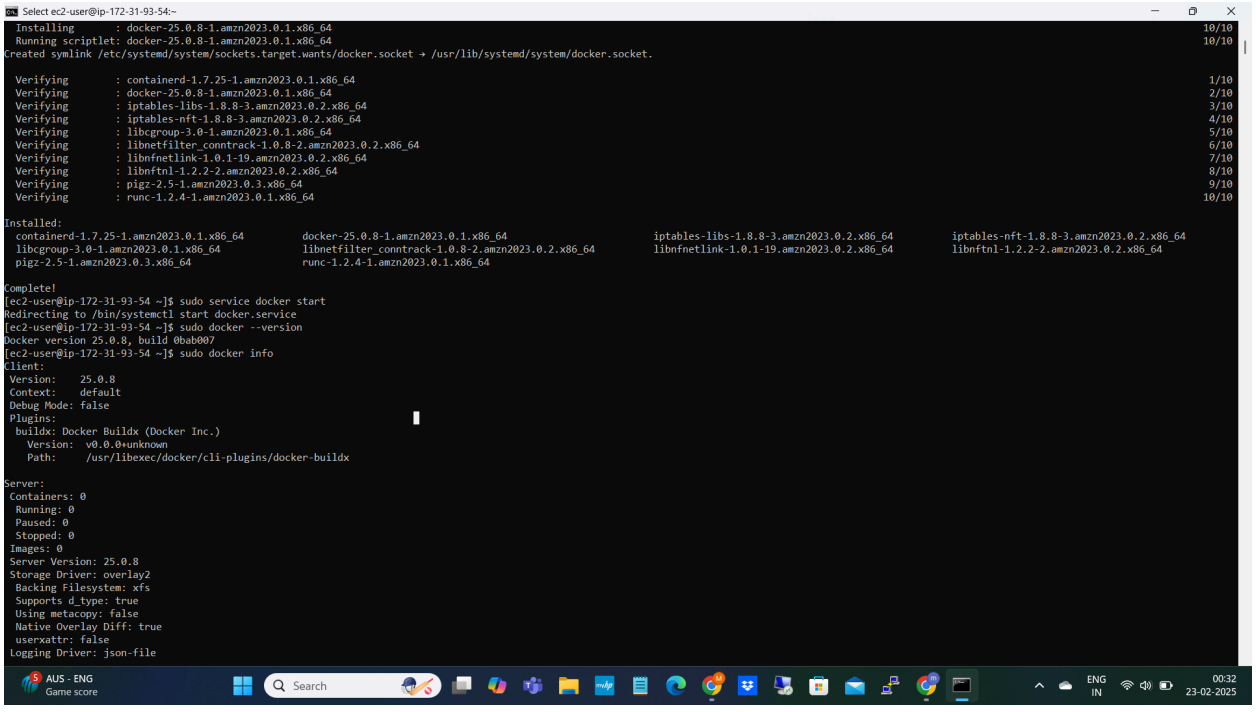
Start docker:

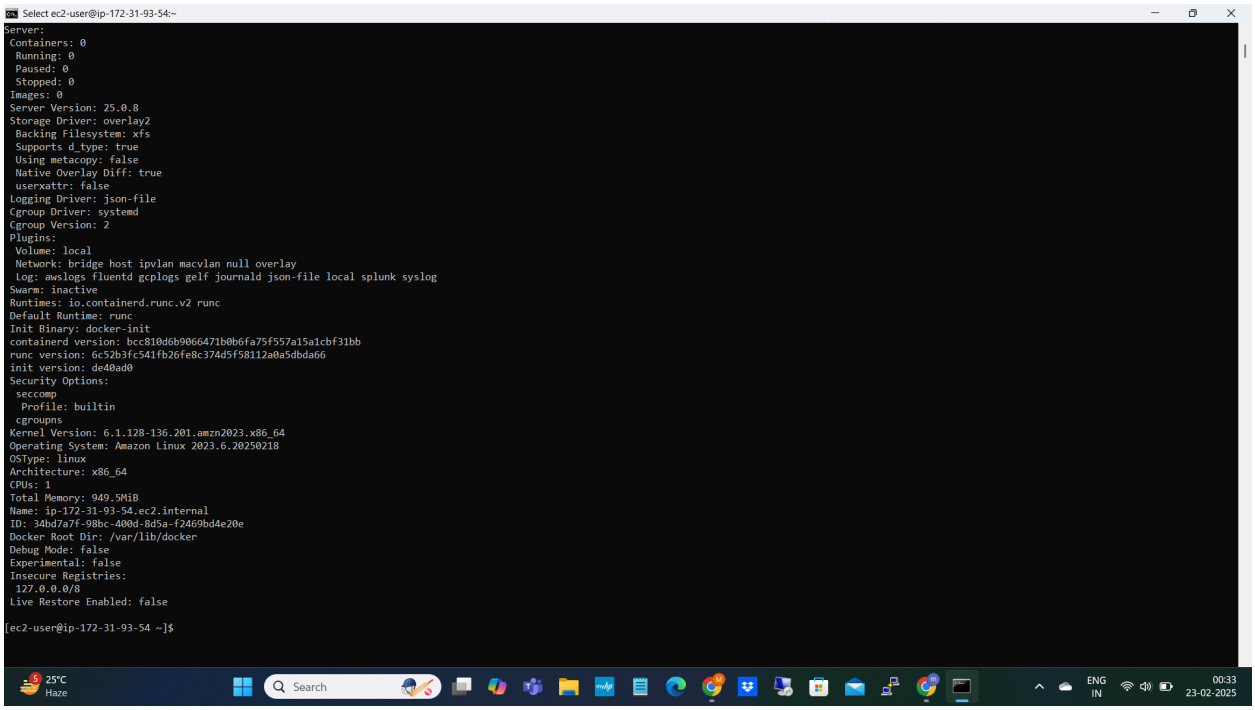
$ sudo service docker start



$ sudo docker –version

$ sudo docker info

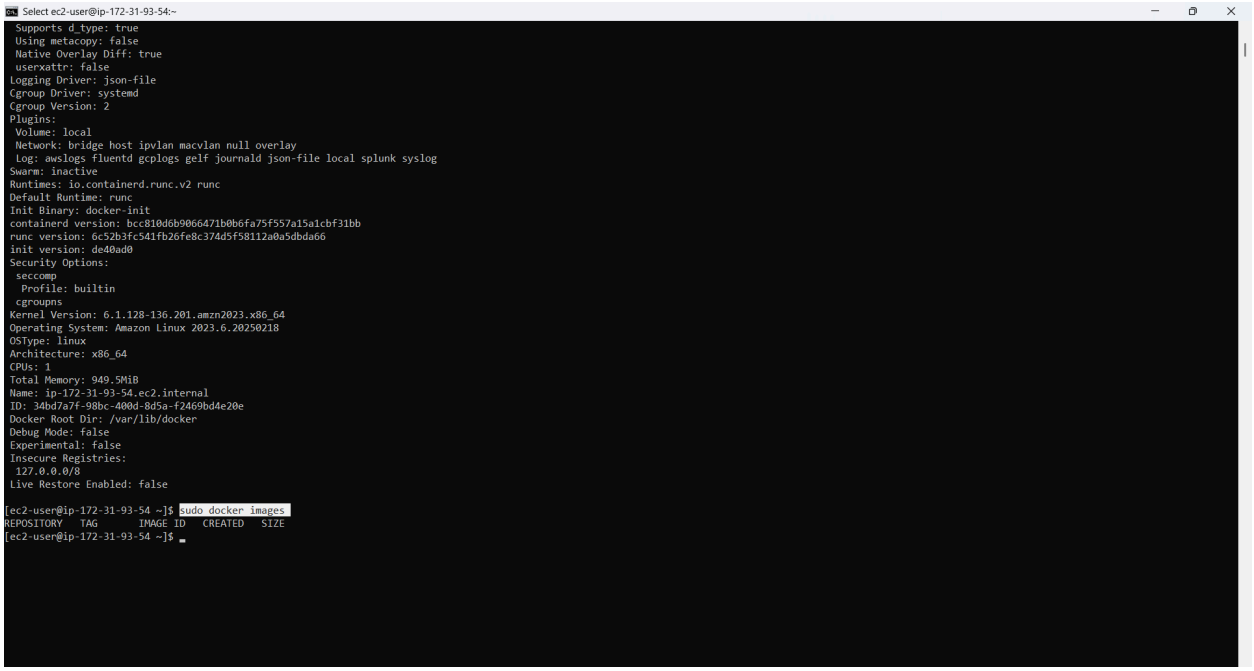




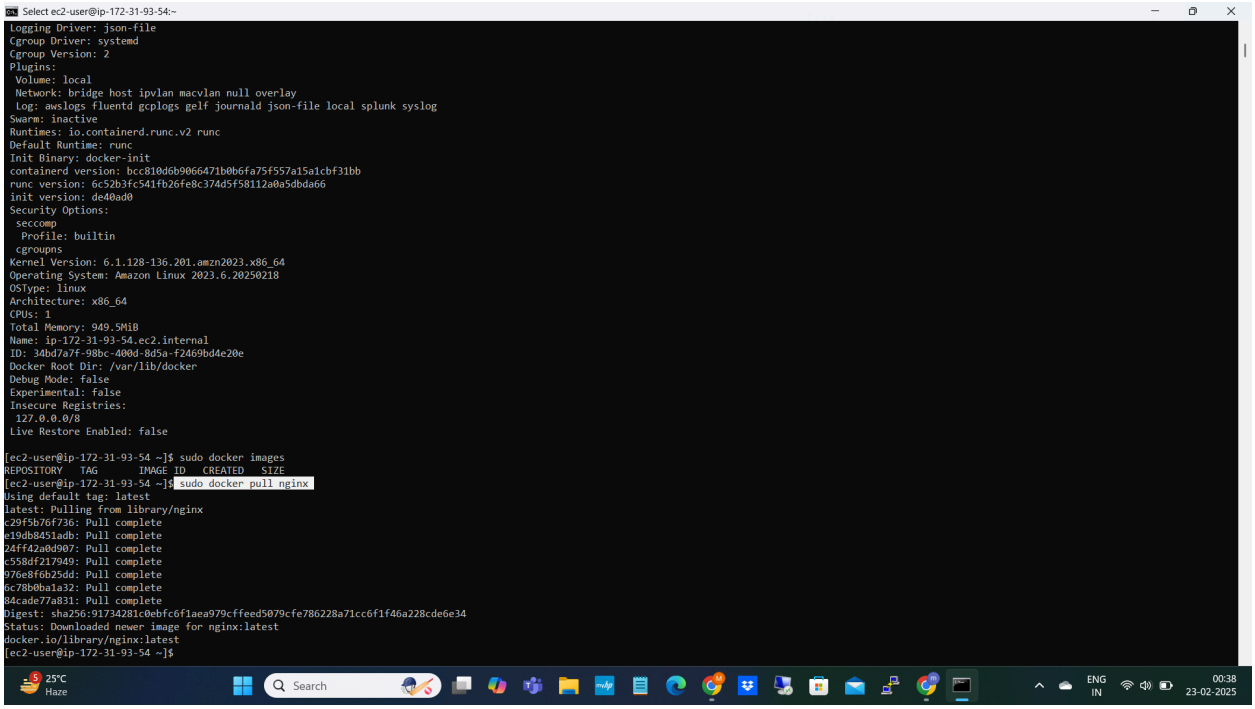
Docker Image Commands:

—----------

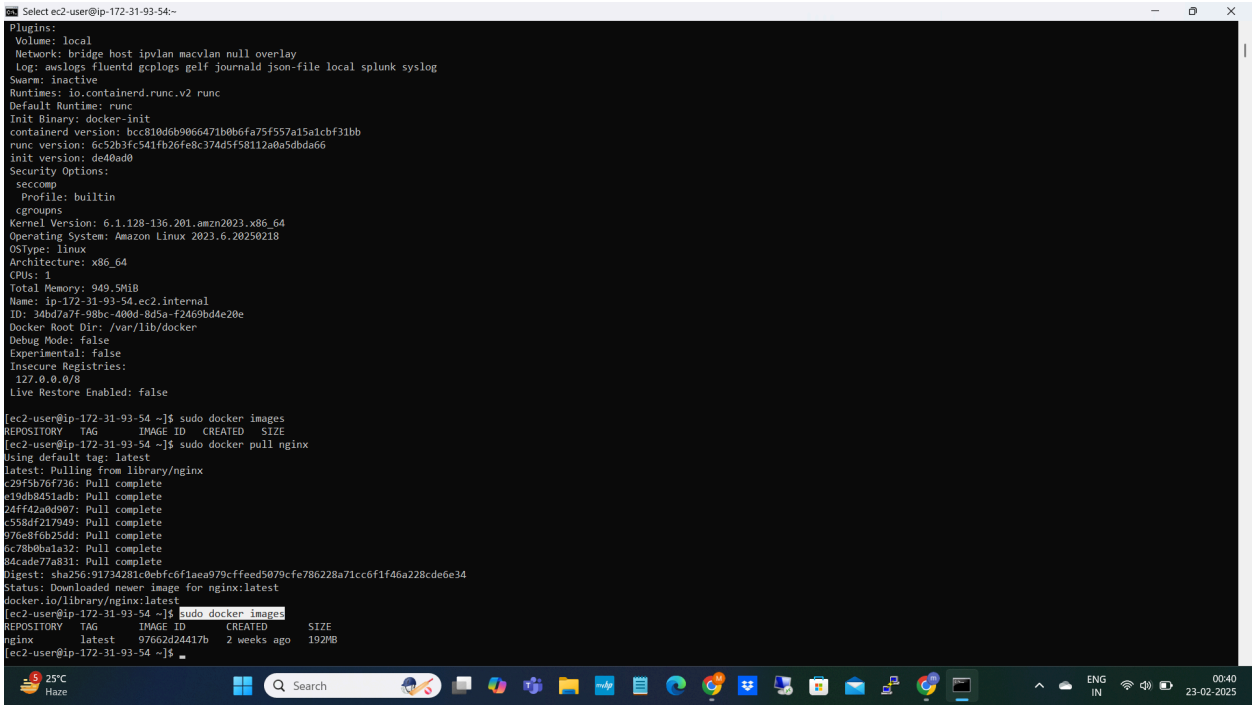
$ Sudo docker images → it will list the images



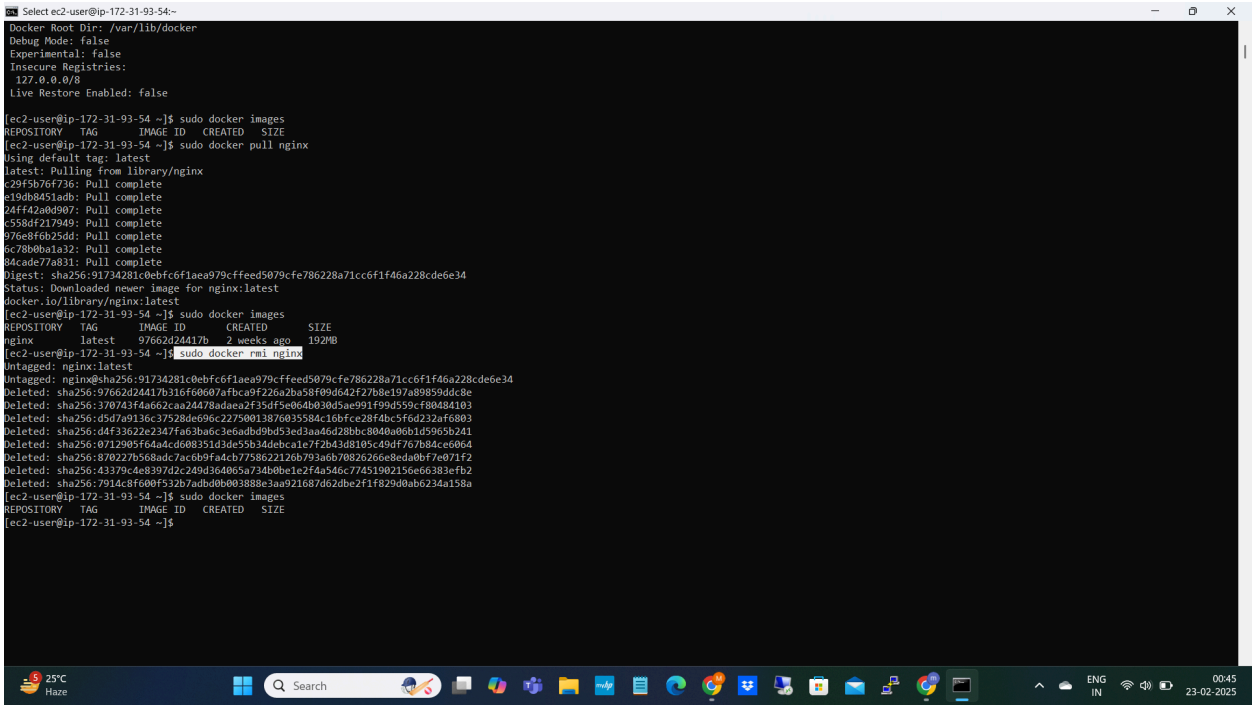
$ sudo docker pull nginx → it will pull the nginx image from docker hub



Now we can see the nginx image

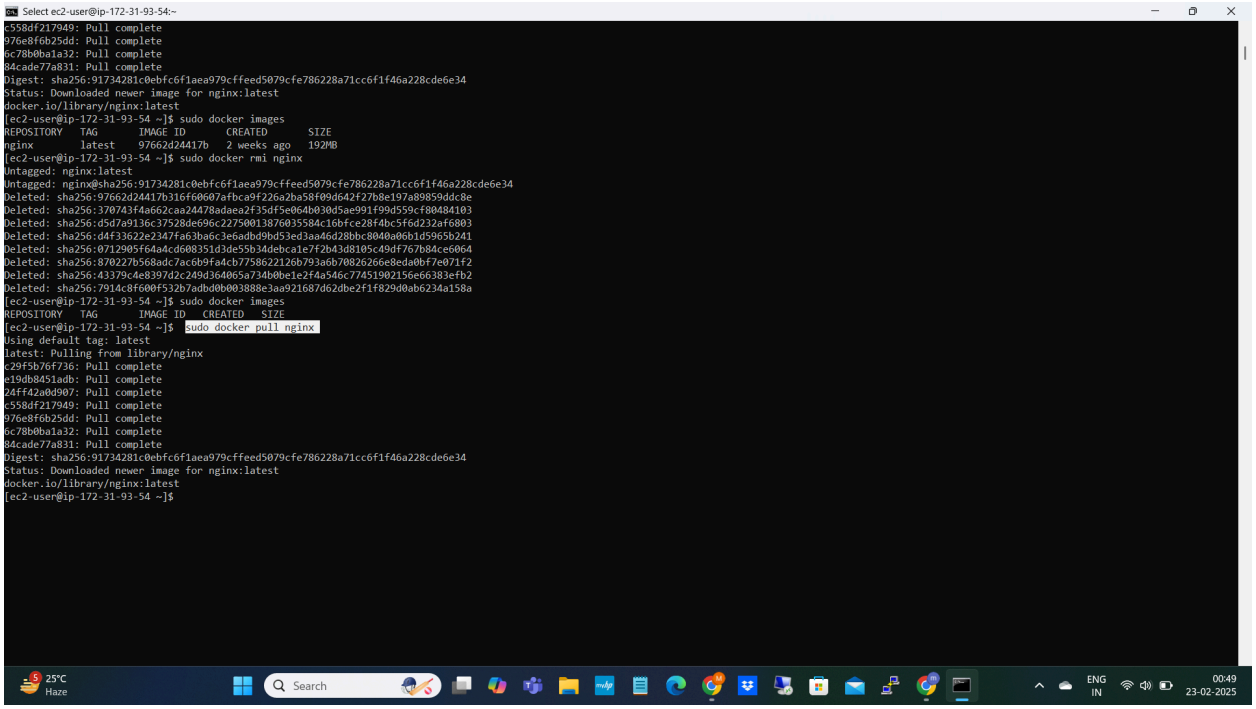


$ sudo docker rmi nginx → it will delete image

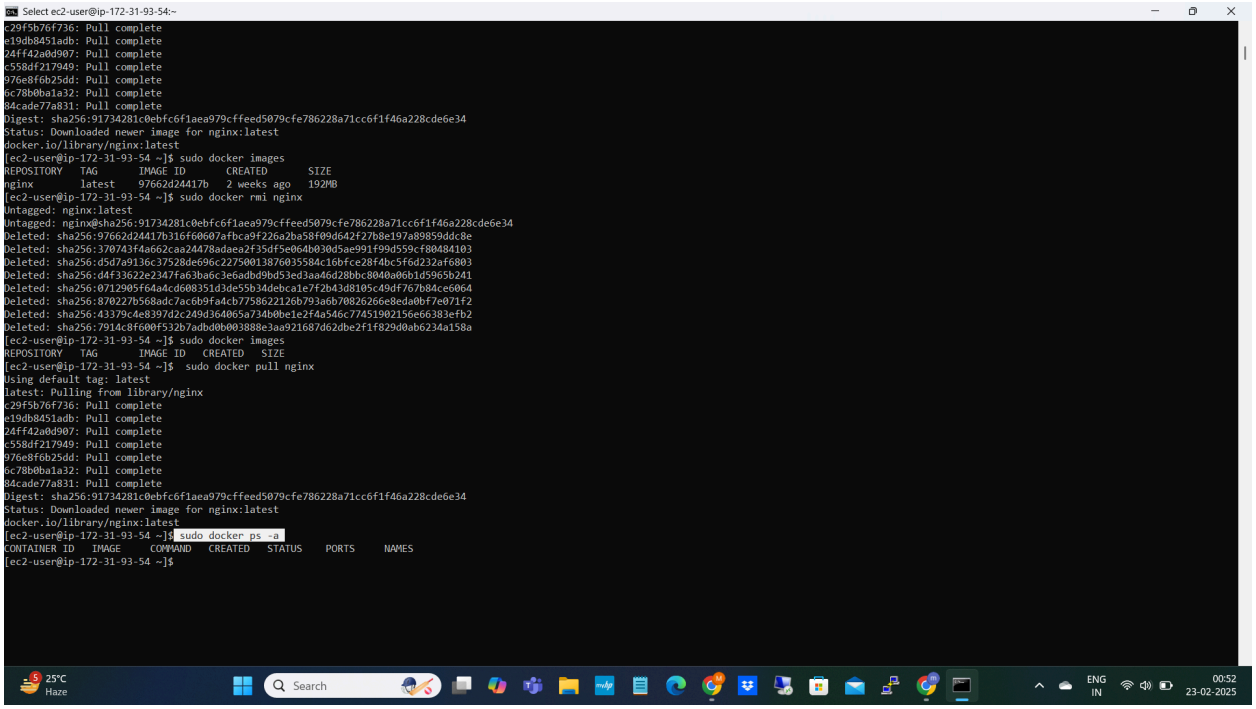


Docker Container Commands:

$ sudo docker pull nginx

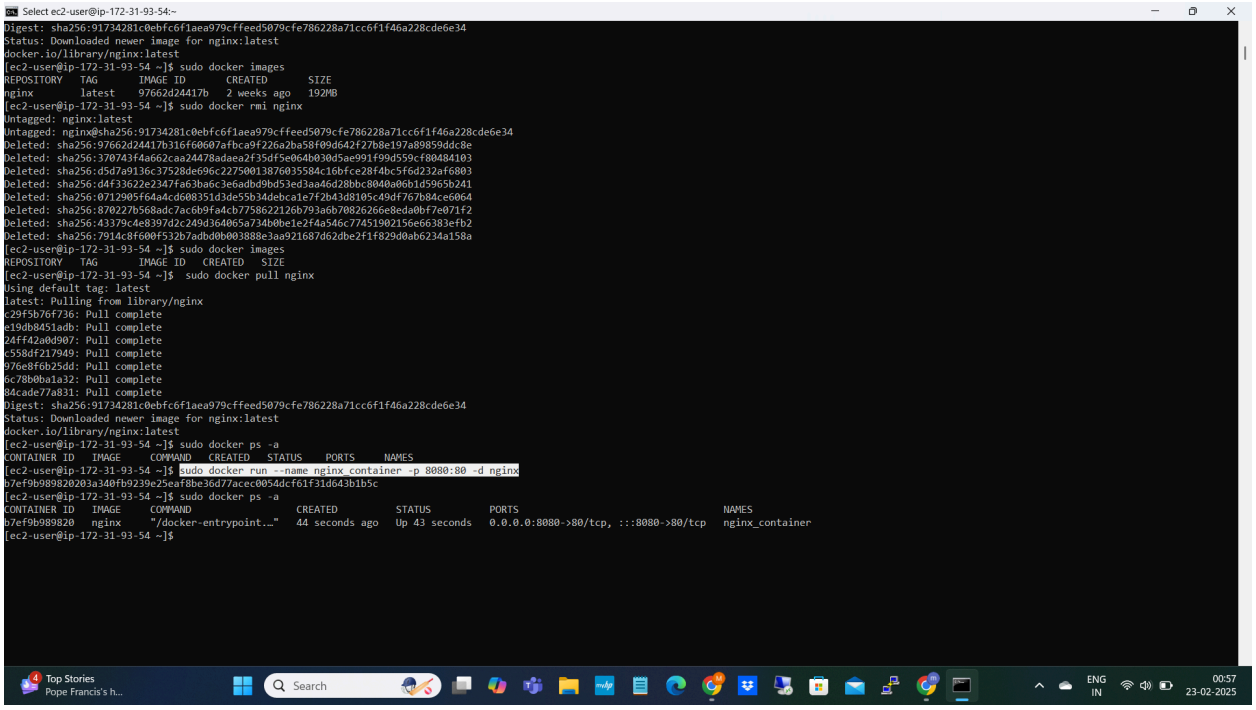


$ sudo docker ps -a –it will list all the containers there is no containers



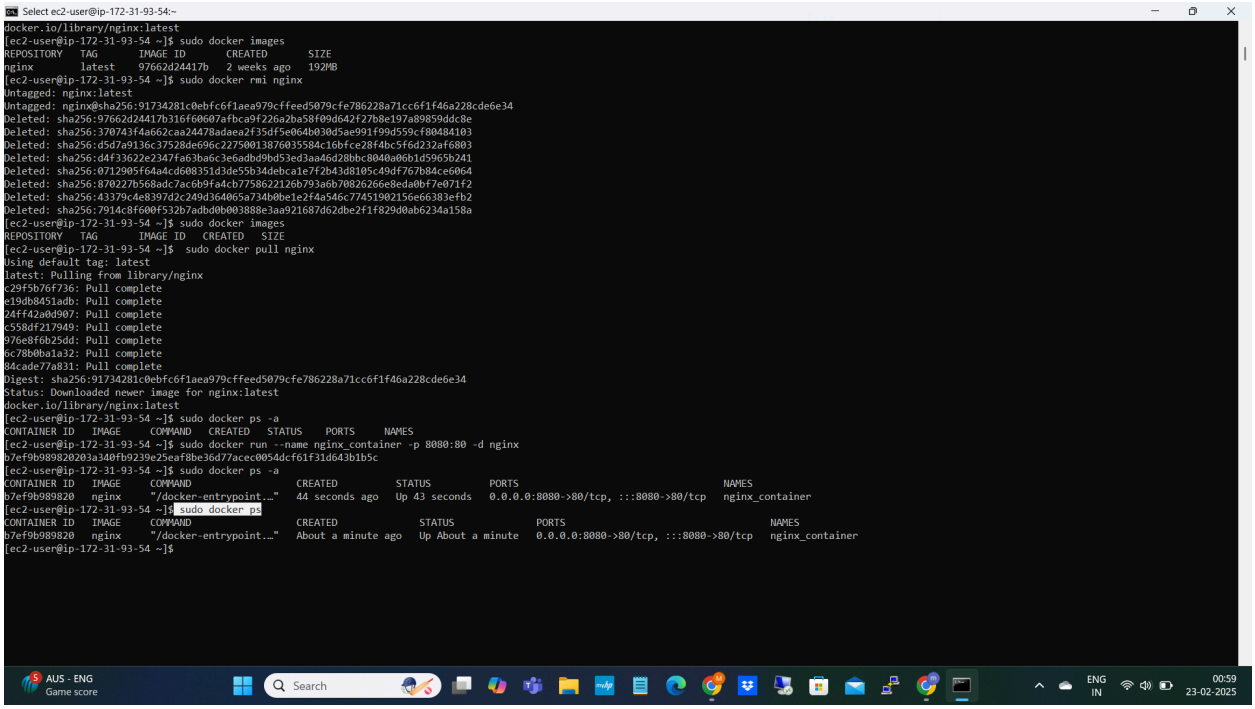
Create a container from the nginx image and run:

$ sudo docker run --name nginx_container -p 8080:80 -d nginx

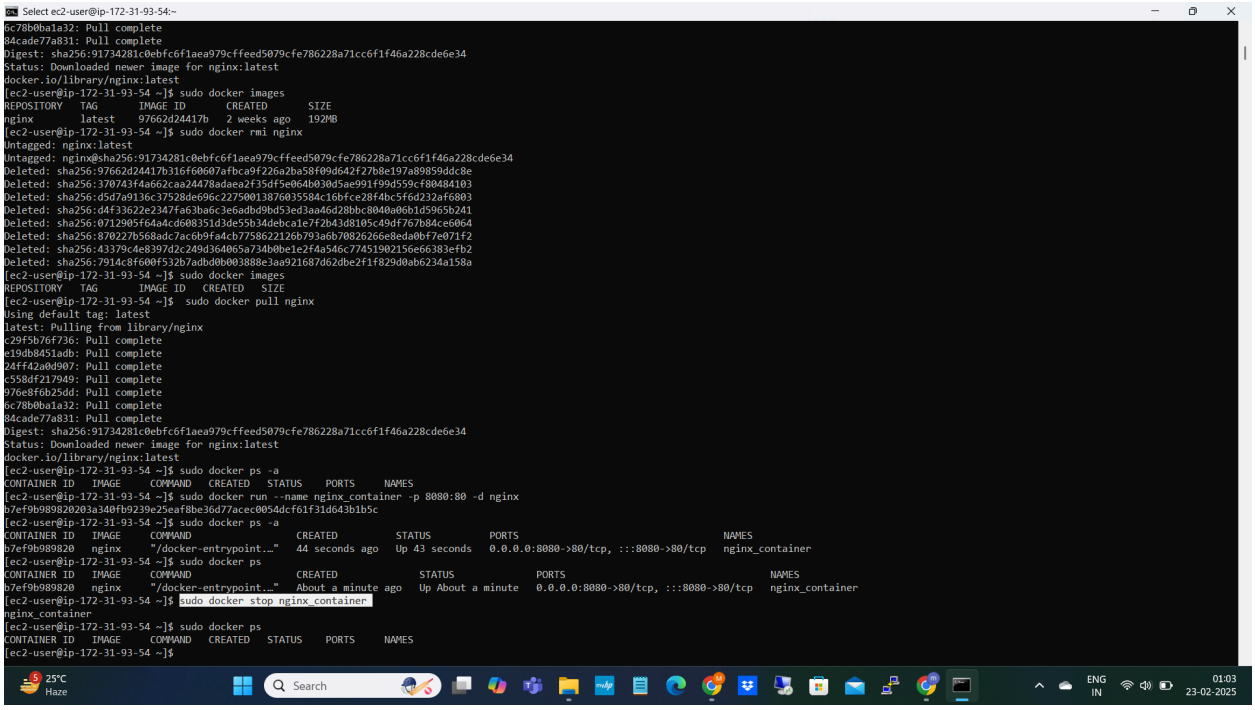


We can see the **nginx_container** created

$ sudo docker ps –it can display only the containers which are running



$ sudo docker stop nginx_container (or ‘b7ef9b989820’)
$ sudo docker ps

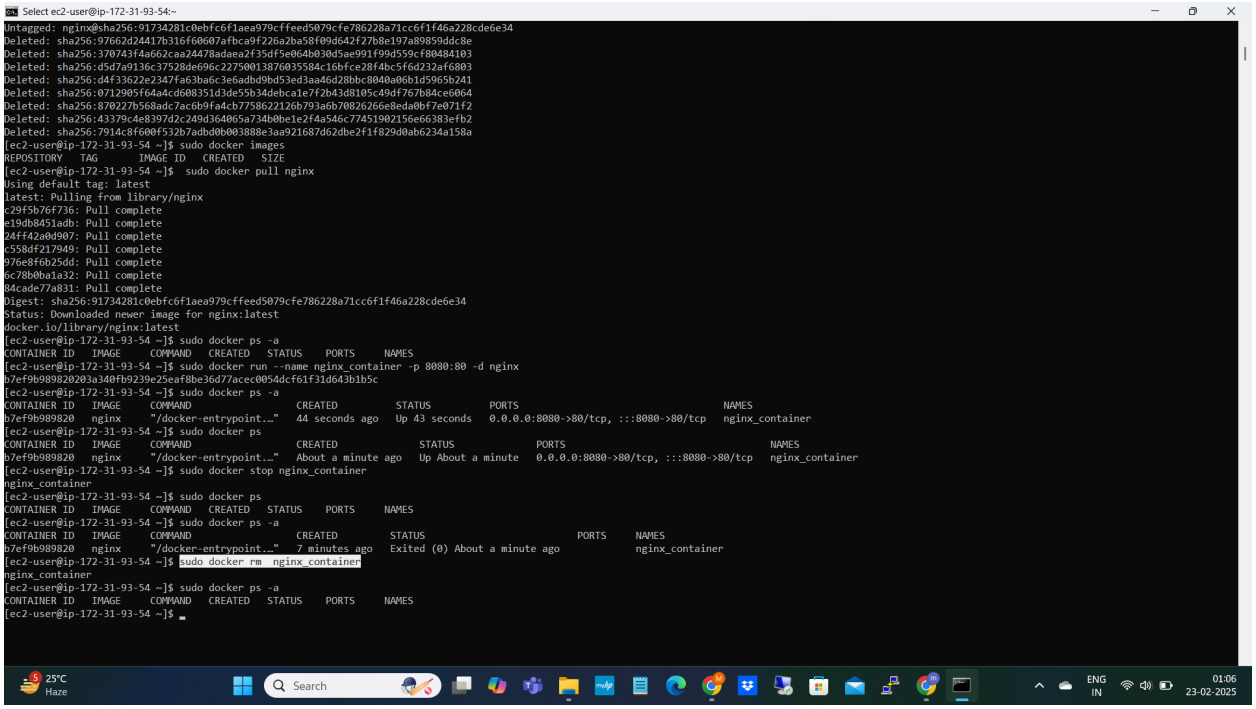


We can see container was stopped

Delete container:

$ sudo docker rm nginx_container (or b7ef9b989820)

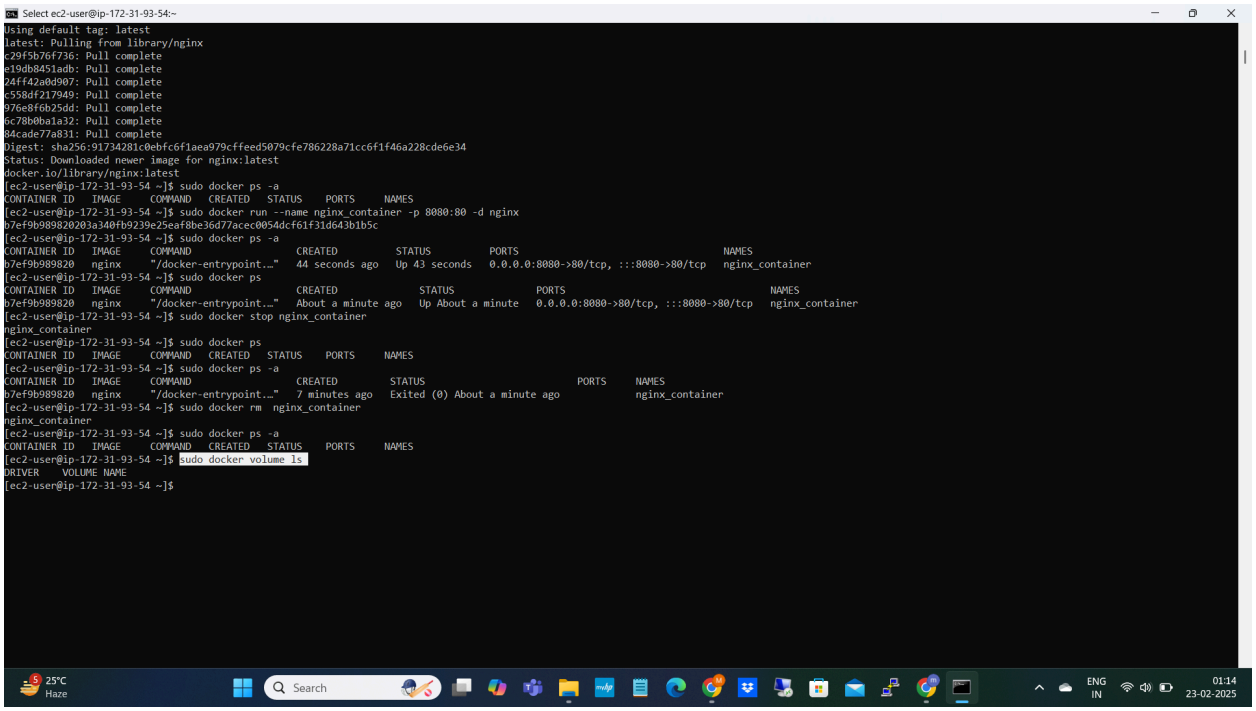
$ sudo docker ps -a



Docker Volume commands:

—----------------------------------

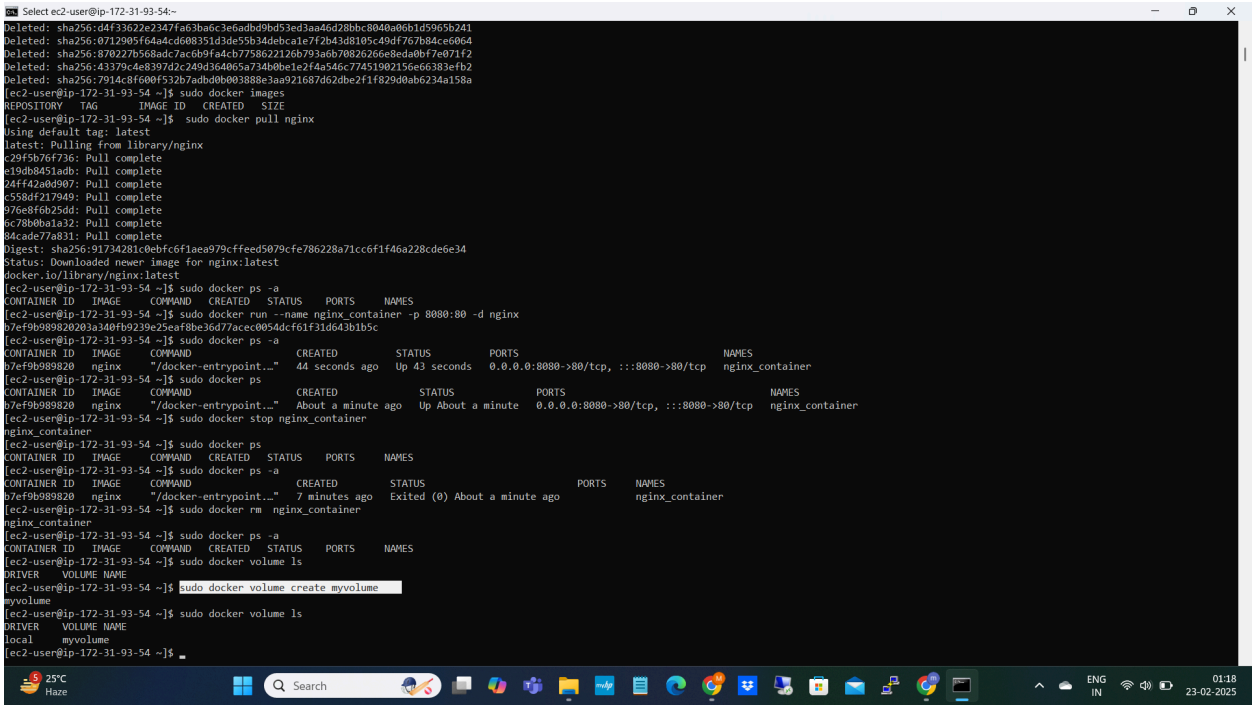
$ sudo docker volume ls →list the docker volumes here no volumes created yet



Create a docker volume:

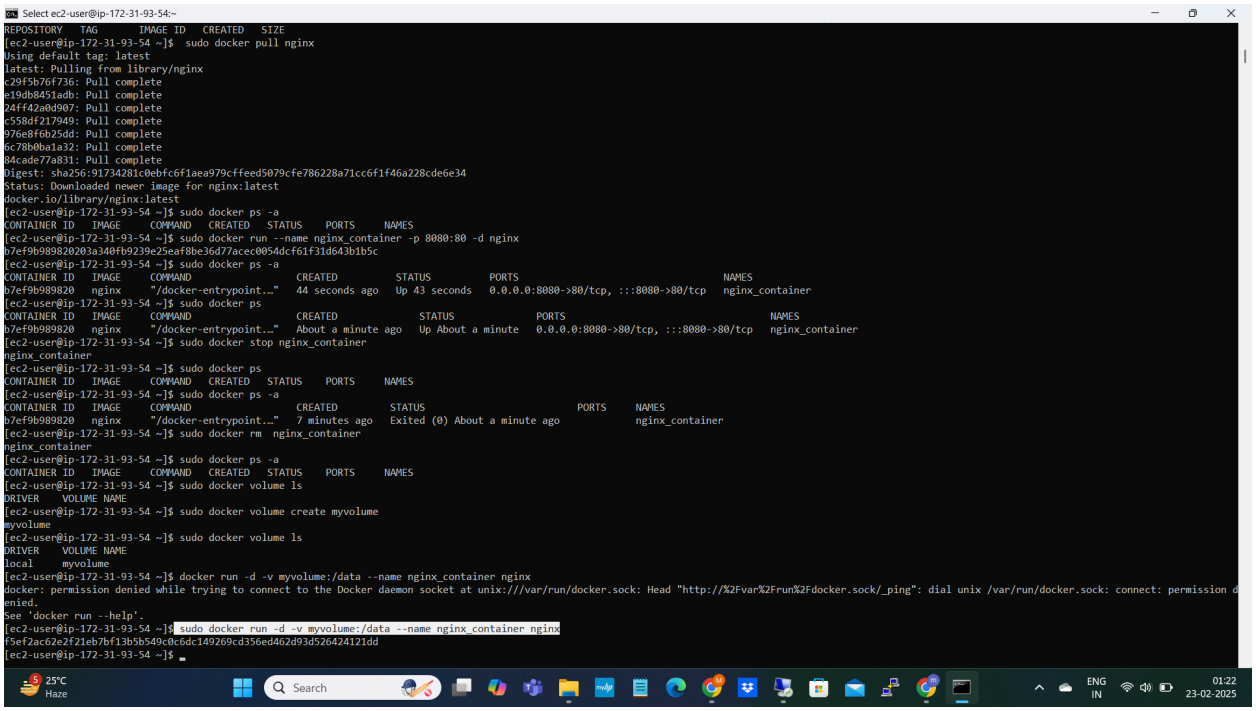
$ sudo docker volume create myvolume

$ sudo docker volume ls



Mount the volume to a container and image

$ docker run -d -v myvolume:/data --name nginx_container nginx



Inspect volume:

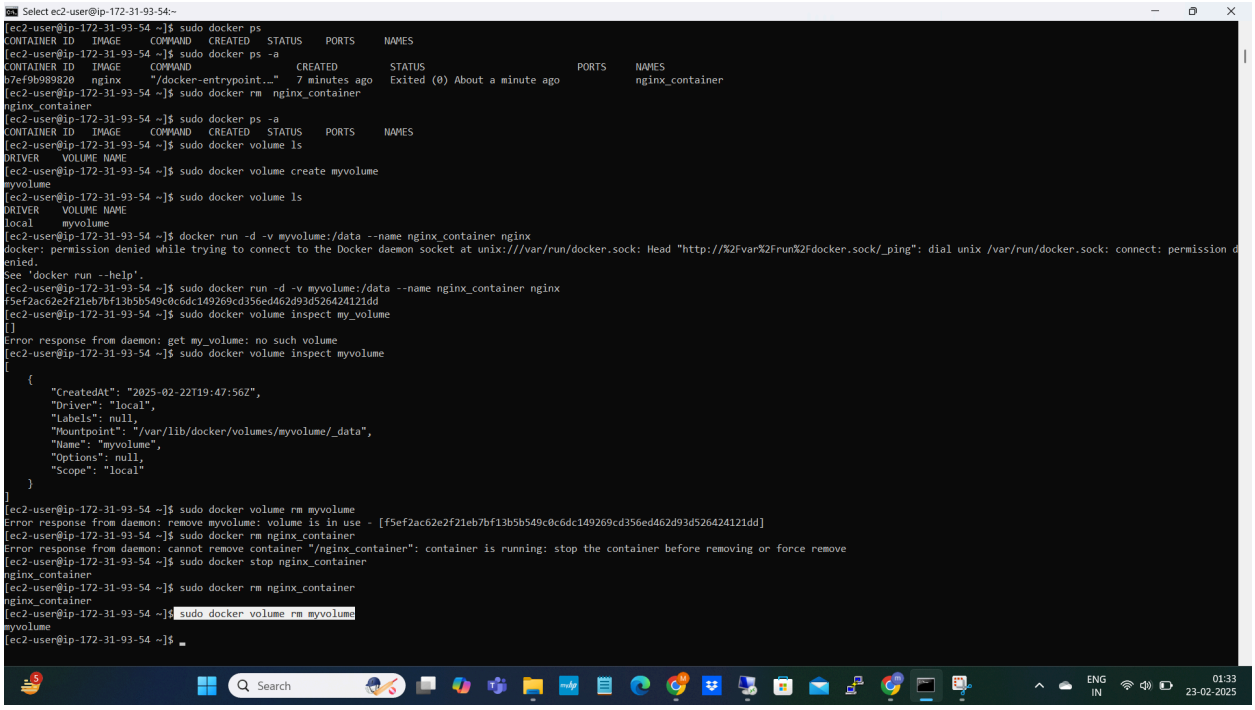
$ sudo docker volume inspect myvolume

```
[ec2-user@ip-172-31-93-54 ~]$ sudo docker volume inspect myvolume
[
    {
        "CreatedAt": "2025-02-22T19:47:56Z",
        "Driver": "local",
        "Labels": null,
        "Mountpoint": "/var/lib/docker/volumes/myvolume/_data",
        "Name": "myvolume",
        "Options": null,
        "Scope": "local"
    }
]
[ec2-user@ip-172-31-93-54 ~]$
```

Delete volume:

To delete container we should make volume unused(by stop and delete the container)

$ sudo docker volume rm myvolume



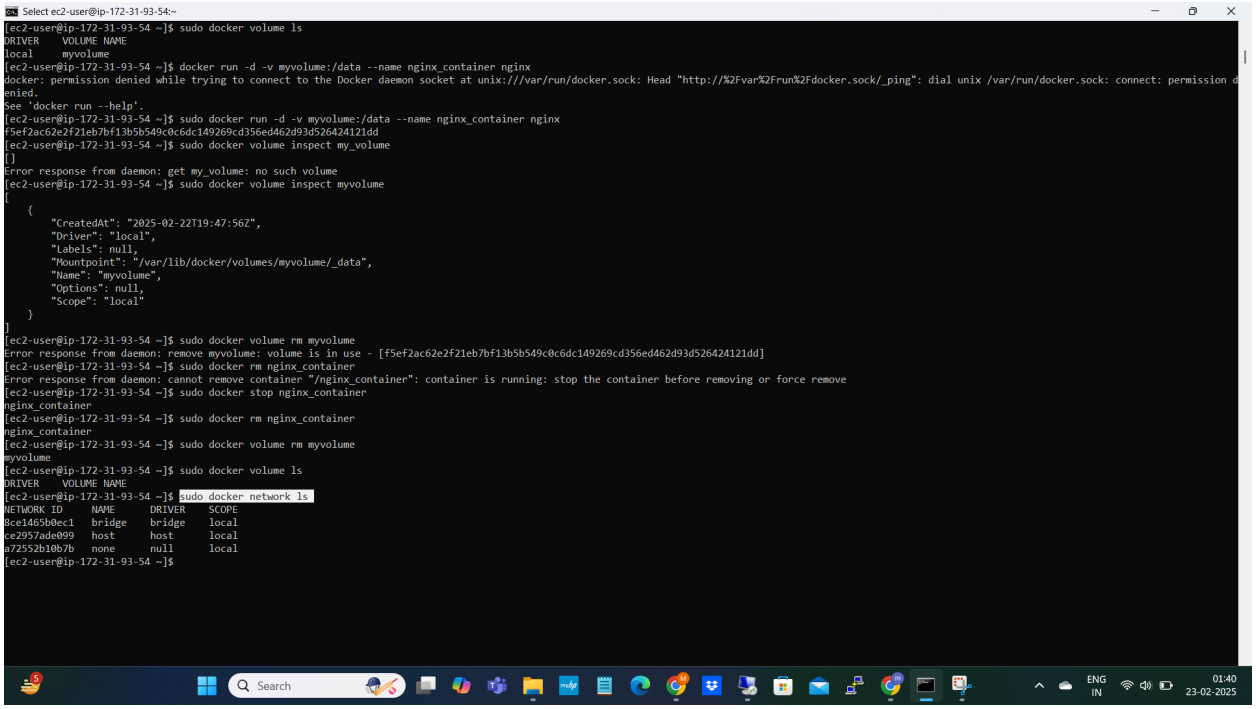
$ sudo docker volume ls —Volume deleted

```
nginx_container
[ec2-user@ip-172-31-93-54 ~]$ sudo docker volume rm myvolume
myvolume
[ec2-user@ip-172-31-93-54 ~]$ sudo docker volume ls
DRIVER    VOLUME NAME
[ec2-user@ip-172-31-93-54 ~]$
```

Docker network commands:

—----------

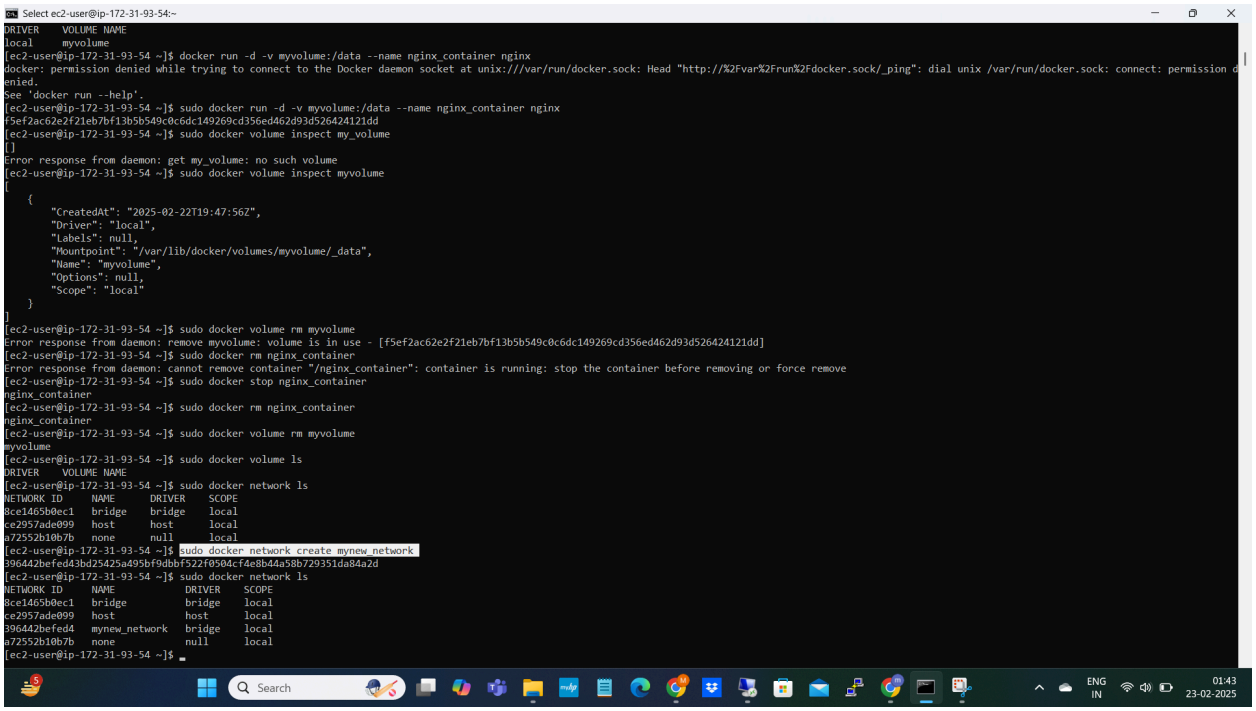
$ sudo docker network ls → it will list all the networks



Create docker network:

$ sudo docker network create **mynew_network** → created mynew_network

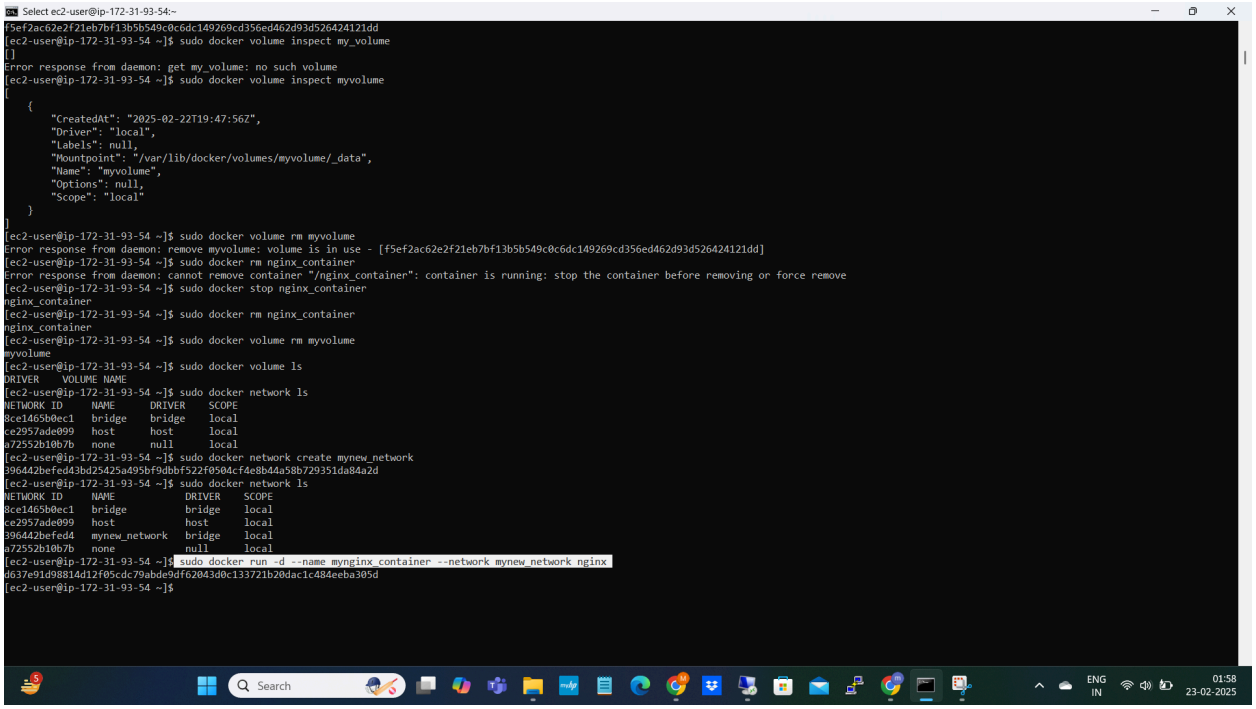
$ sudo docker network ls



Connect the container to the network

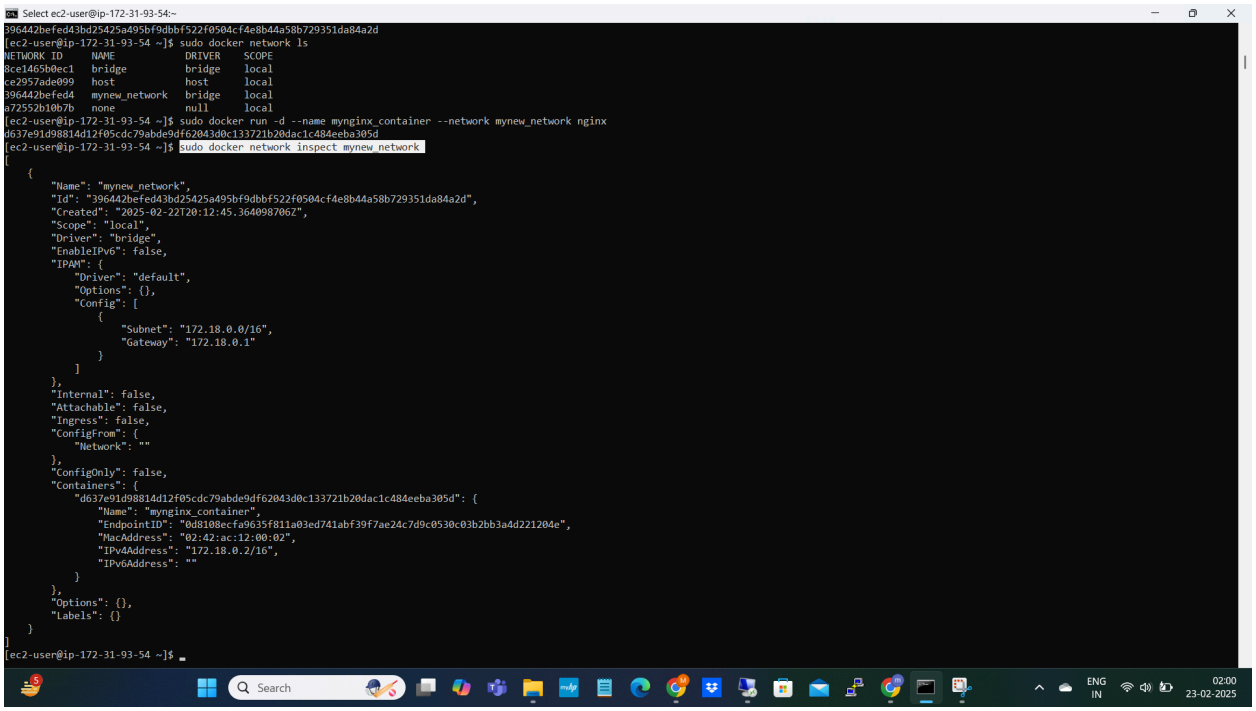
$ docker run -d --name mynginx_container --network mynew_network nginx

Above command can create a mynginx_contaner and connect to the mynew_network



To see the network details

$ **sudo docker network inspect mynew_network**

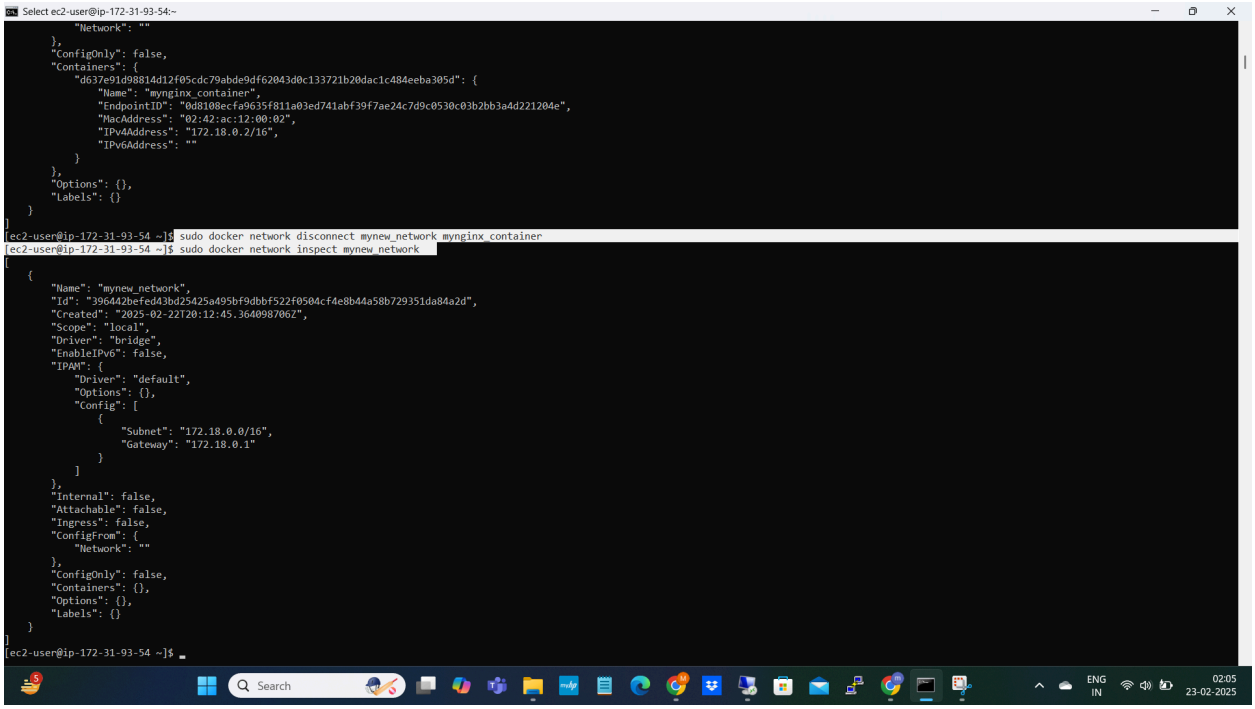


We can able to view the containers which are connected to this network

Disconnect the container from the network

$ sudo docker network disconnect mynew_network mynginx_container

$ sudo docker network inspect mynew_network



We can see no containers connected now

Delete the mynew_network:

$ sudo docker network rm mynew_network