Shaikh Ateeb Ahmed
17-09-2025

Deploy Docker on Amazon Linux 2 EC2 and Run a Web App in a Docker Container

Step1: Launch EC2 Instance

1. **Open the EC2 console**

   Sign in to the AWS Management Console.

   Services → **EC2** → **Instances** → **Launch instances**.

   Name: AmazonLinux2-Docker-Lab

2. **Choose an AMI**

   AMI: Select **Amazon Linux 2 (64-bit x86)** (Amazon Linux 2 AMI (HVM), SSD).

3. **Choose Instance Type**

   Instance type: t3.micro (Free Tier eligible).

4. **Key pair**

   When prompted **Select a key pair**: Choose an existing key if you have one

5. **Configure Instance Details**

   Number of instances: 1.

   Network (VPC): select default VPC.

   Subnet: choose a **public subnet** (one that auto-assigns a public IP).

   Auto-assign Public IP: **Enable** (so you get a Public IPv4).

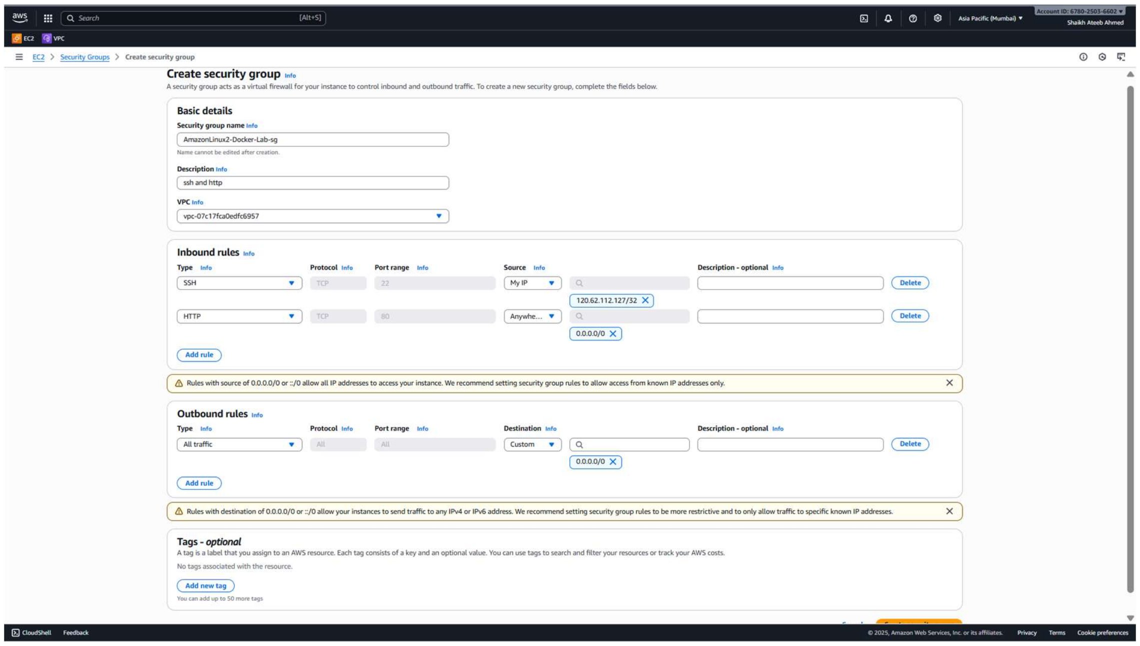
   Create new security group named e.g. AmazonLinux2-Docker-Lab-sg.

   Add inbound rules:

   SSH — Type: SSH, Port: 22, Source: **My IP** (recommended).

   HTTP — Type: HTTP, Port: 80, Source: 0.0.0.0/0.

Shaikh Ateeb Ahmed
17-09-2025



EC2 > Security Groups > Create security group

Create security group Info

A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. To create a new security group, complete the fields below.

Basic details

Security group name: AmazonLinux2-Docker-Lab-sg
Name cannot be edited after creation.

Description: ssh and http

VPC: vpc-07c17fca0edfc6957

Inbound rules

Type	Protocol	Port range	Source		Description - optional
SSH	TCP	22	My IP	120.62.112.127/32	
HTTP	TCP	80	Anywhe...	0.0.0.0/0	

Rules with source of 0.0.0.0/0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Outbound rules

Type	Protocol	Port range	Destination		Description - optional
All traffic	All	All	Custom	0.0.0.0/0	

Rules with destination of 0.0.0.0/0 or ::/0 allow your instances to send traffic to any IPv4 or IPv6 address. We recommend setting security group rules to be more restrictive and to only allow traffic to specific known IP addresses.

Tags - *optional*

A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

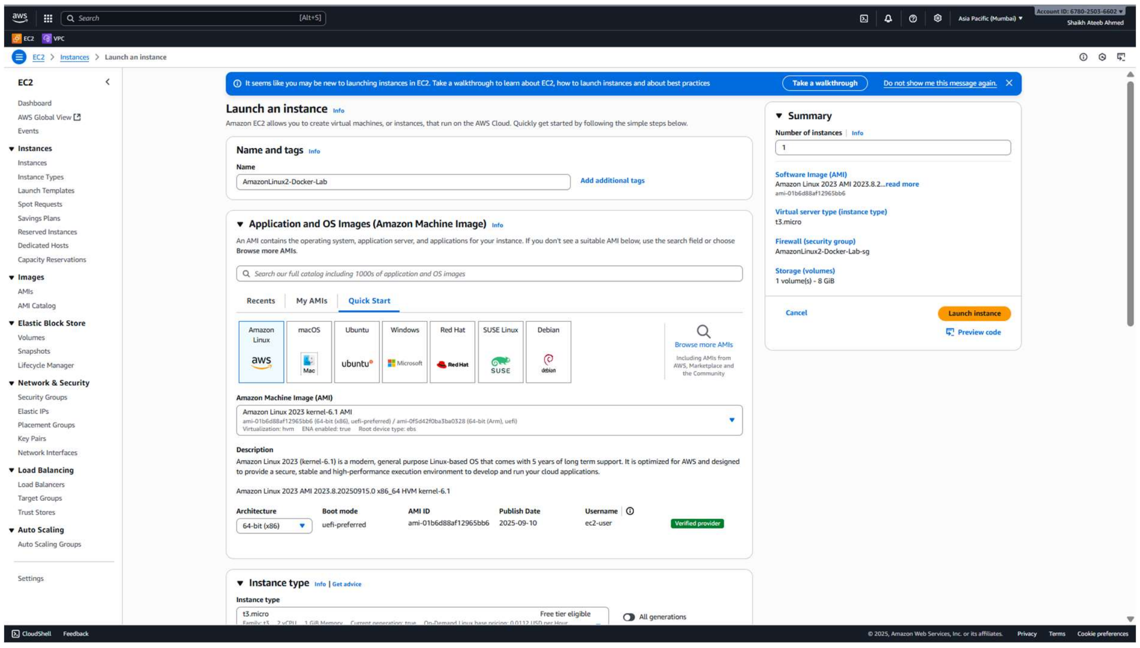
No tags associated with the resource.

You can add up to 50 more tags

6. Add Storage

Default root (e.g. 8 GiB gp2/gp3) is usually fine for a Docker lab. Change size if you need more disk.

7. Click **Review and Launch**.



EC2 > Instances > Launch an instance

It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices

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

Name: AmazonLinux2-Docker-Lab

Application and OS Images (Amazon Machine Image)

An AMI contains the operating system, application server, and applications for your instance. If you don't see a suitable AMI below, use the search field or choose Browse more AMIs.

Amazon Machine Image (AMI): Amazon Linux 2023 kernel-6.1 AMI
ami-01b6d88af12965bb6 (64-bit (x86), uefi-preferred) / ami-0f5d42f0ba3ba0328 (64-bit (Arm), uefi)
Virtualization: hvm ENA enabled: true Root device type: ebs

Description
Amazon Linux 2023 (kernel-6.1) is a modern, general purpose Linux-based OS that comes with 5 years of long term support. It is optimized for AWS and designed to provide a secure, stable and high-performance execution environment to develop and run your cloud applications.

Amazon Linux 2023 AMI 2023.8.20250915.0 x86_64 HVM kernel-6.1

Architecture	Boot mode	AMI ID	Publish Date	Username	
64-bit (x86)	uefi-preferred	ami-01b6d88af12965bb6	2025-09-10	ec2-user	Verified provider

Instance type

Instance type: t3.micro Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0112 USD per Hour

Summary

Number of instances: 1

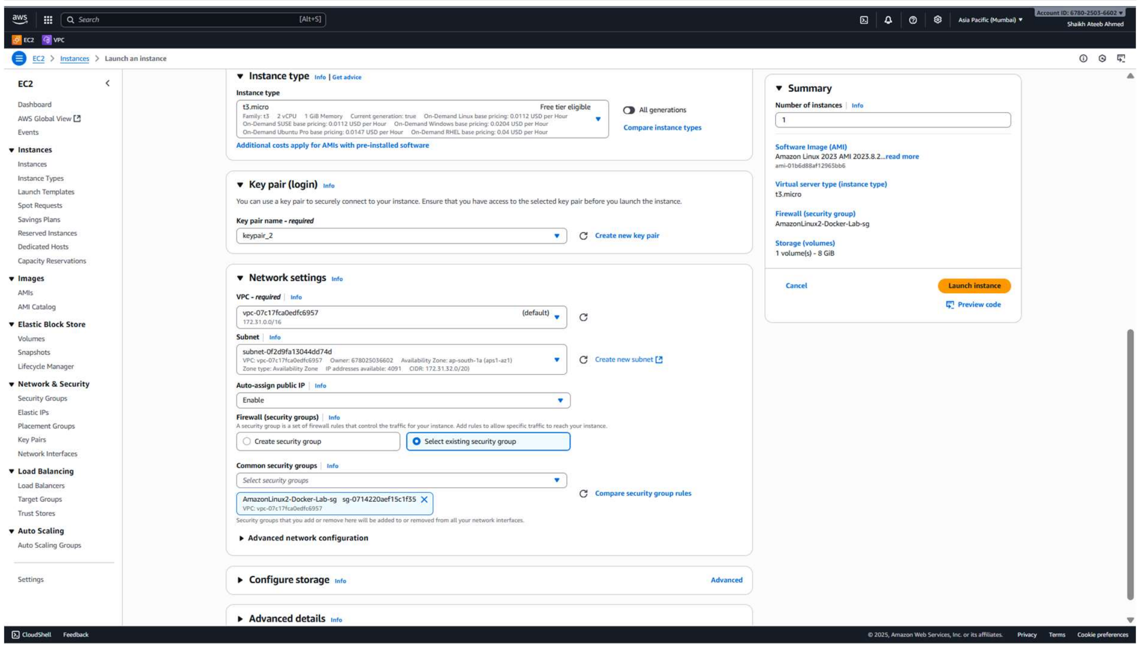
Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more
ami-01b6d88af12965bb6

Virtual server type (instance type): t3.micro

Firewall (security group): AmazonLinux2-Docker-Lab-sg

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

Shaikh Ateeb Ahmed
17-09-2025



Instance type
t3.micro — Free tier eligible
Family: t3 2 vCPU 1 GiB Memory Current generation: true On-Demand Linux base pricing: 0.0112 USD per Hour
On-Demand SUSE base pricing: 0.0112 USD per Hour On-Demand Windows base pricing: 0.0204 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0147 USD per Hour On-Demand RHEL base pricing: 0.04 USD per Hour
Additional costs apply for AMIs with pre-installed software

Key pair (login)
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*: keypair_2

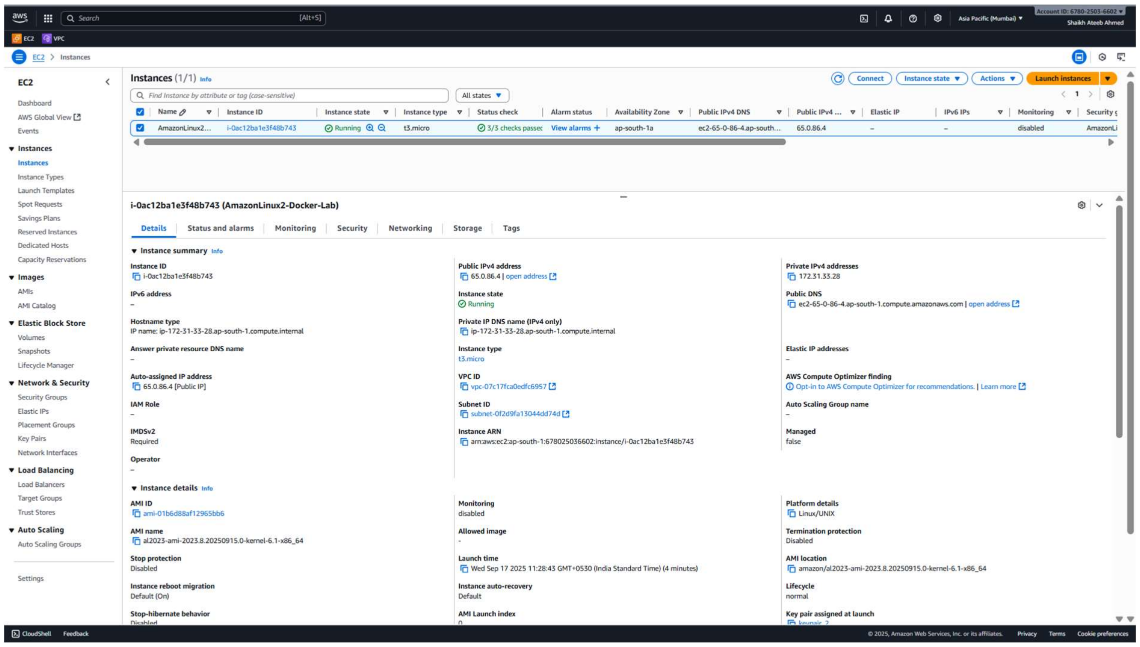
Network settings
VPC - *required*: vpc-07c17fca0edfc6957 (default) 172.31.0.0/16
Subnet: subnet-0f2d9fa13044dd74d
VPC: vpc-07c17fca0edfc6957 Owner: 678025036602 Availability Zone: ap-south-1a (aps1-az1)
Zone type: Availability Zone IP addresses available: 4091 CIDR: 172.31.32.0/20)
Auto-assign public IP: Enable
Firewall (security groups)
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: AmazonLinux2-Docker-Lab-sg sg-0714220aef15c1f35 VPC: vpc-07c17fca0edfc6957
Security groups that you add or remove here will be added to or removed from all your network interfaces.

Summary
Number of instances: 1
Software Image (AMI): Amazon Linux 2023 AMI 2023.8.2...read more ami-01b6d88af12965bb6
Virtual server type (instance type): t3.micro
Firewall (security group): AmazonLinux2-Docker-Lab-sg
Storage (volumes): 1 volume(s) - 8 GiB

8. Verify the instance

Go to **Instances** page. Select your AmazonLinux2-Docker-Lab instance.

Wait until **Instance state: running** and **Status checks: 3/3 checks passed** (the console shows progress).



Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4 ...	Elastic IP	IPv6 IPs	Monitoring	Security g
AmazonLinux2...	i-0ac12ba1e3f48b743	Running	t3.micro	3/3 checks passec	View alarms +	ap-south-1a	ec2-65-0-86-4.ap-south...	65.0.86.4	–	–	disabled	AmazonLi

i-0ac12ba1e3f48b743 (AmazonLinux2-Docker-Lab)

Instance summary

Instance ID: i-0ac12ba1e3f48b743	Public IPv4 address: 65.0.86.4 | open address	Private IPv4 addresses: 172.31.33.28
IPv6 address: –	Instance state: Running	Public DNS: ec2-65-0-86-4.ap-south-1.compute.amazonaws.com | open address
Hostname type: IP name: ip-172-31-33-28.ap-south-1.compute.internal	Private IP DNS name (IPv4 only): ip-172-31-33-28.ap-south-1.compute.internal	
Answer private resource DNS name: –	Instance type: t3.micro	Elastic IP addresses: –
Auto-assigned IP address: 65.0.86.4 [Public IP]	VPC ID: vpc-07c17fca0edfc6957	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role: –	Subnet ID: subnet-0f2d9fa13044dd74d	Auto Scaling Group name: –
IMDSv2: Required	Instance ARN: arn:aws:ec2:ap-south-1:678025036602:instance/i-0ac12ba1e3f48b743	Managed: false
Operator: –		

Instance details

AMI ID: ami-01b6d88af12965bb6	Monitoring: disabled	Platform details: Linux/UNIX
AMI name: al2023-ami-2023.8.20250915.0-kernel-6.1-x86_64	Allowed image: -	Termination protection: Disabled
Stop protection: Disabled	Launch time: Wed Sep 17 2025 11:28:43 GMT+0530 (India Standard Time) (4 minutes)	AMI location: amazon/al2023-ami-2023.8.20250915.0-kernel-6.1-x86_64
Instance reboot migration: Default (On)	Instance auto-recovery: Default	Lifecycle: normal
Stop-hibernate behavior: Disabled	AMI Launch index: 0	Key pair assigned at launch: keypair_2

Step 2: Connect to EC2 via SSH

Shaikh Ateeb Ahmed
17-09-2025

---

1. **Open your terminal**

   Navigate to the folder where you downloaded your **key pair (.pem)**. (or wherever your .pem file is saved)

   SSH into your EC2 instance

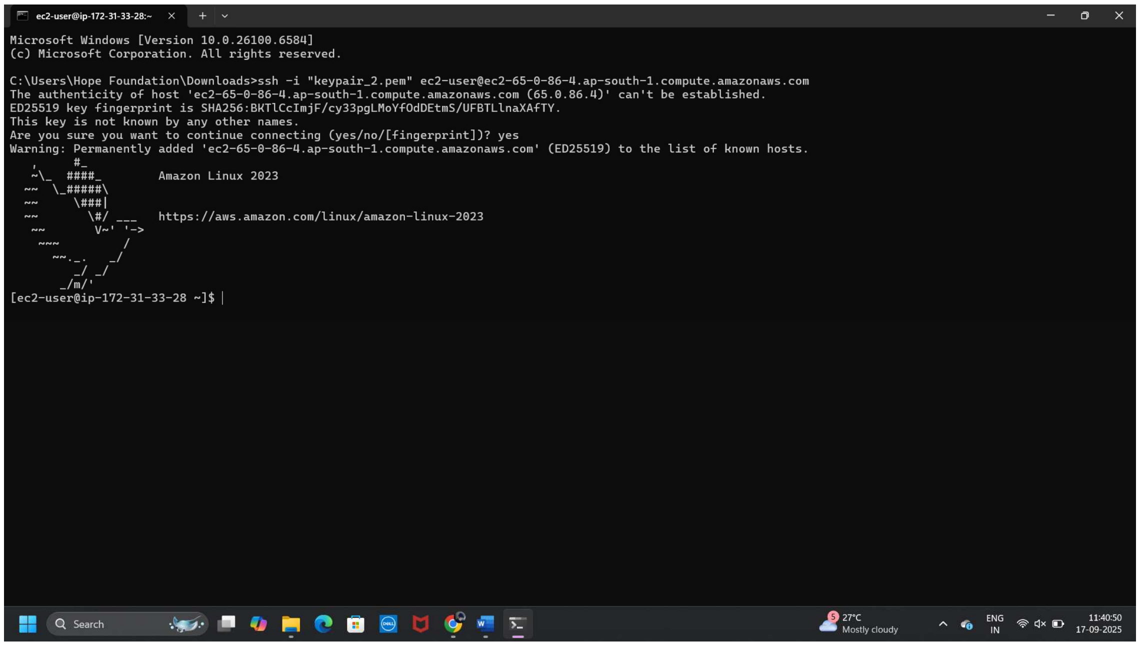
   Replace:

   your-key.pem → with your key file name

   <EC2-Public-IP> → with your instance's **Public IPv4 address** (from the AWS console)

   ssh -i "your-key.pem" ec2-user@<EC2-Public-IP>

   ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com



```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-0-86-4.ap-south-1.compute.amazonaws.com (65.0.86.4)' can't be established.
ED25519 key fingerprint is SHA256:BKTlCcImjF/cy33pgLMoYfOdDEtmS/UFBTLlnaXAfTY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-0-86-4.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-33-28 ~]$ 
```

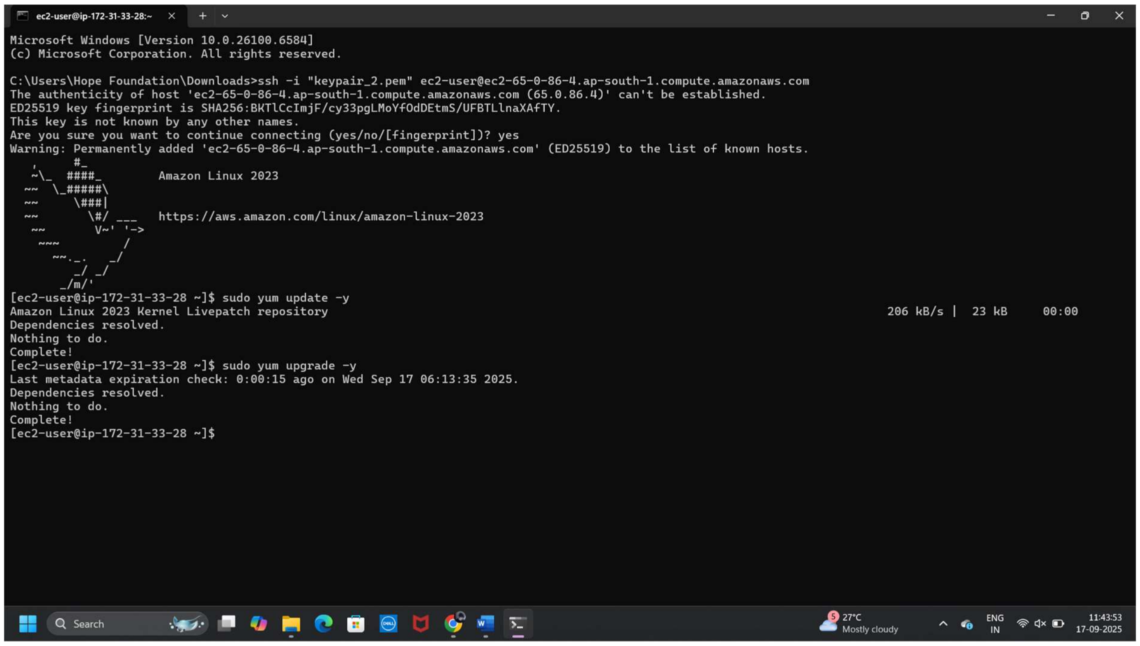
Step 3: Install Docker on Amazon Linux 2

1. **Update and upgrade packages**

   sudo yum update -y

   sudo yum upgrade -y

Shaikh Ateeb Ahmed
17-09-2025



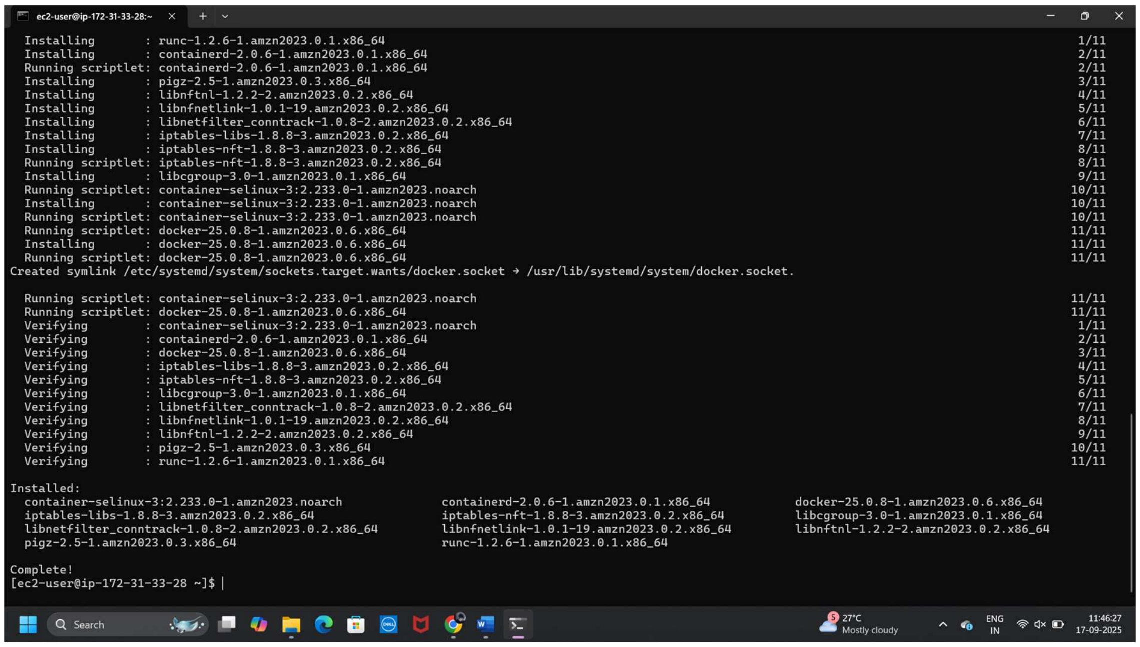
```
Microsoft Windows [Version 10.0.26100.6584]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com
The authenticity of host 'ec2-65-0-86-4.ap-south-1.compute.amazonaws.com (65.0.86.4)' can't be established.
ED25519 key fingerprint is SHA256:BKTlCcImjF/cy33pgLMoYfOdDEtmS/UFBTLlnaXAfTY.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-65-0-86-4.ap-south-1.compute.amazonaws.com' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-172-31-33-28 ~]$ sudo yum update -y
Amazon Linux 2023 Kernel Livepatch repository                                                206 kB/s |  23 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-33-28 ~]$ sudo yum upgrade -y
Last metadata expiration check: 0:00:15 ago on Wed Sep 17 06:13:35 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-33-28 ~]$
```

2. **Install required utilities**

   sudo yum install -y yum-utils

3. **Install Docker**

   sudo yum install -y docker

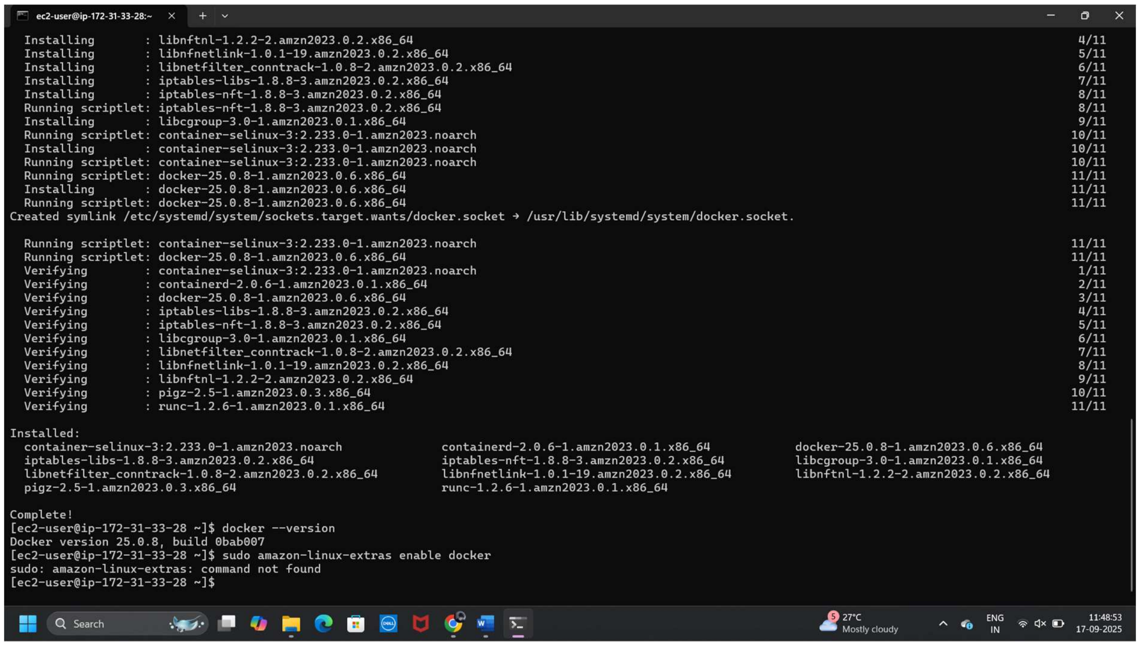


```
  Installing       : runc-1.2.6-1.amzn2023.0.1.x86_64                                                          1/11
  Installing       : containerd-2.0.6-1.amzn2023.0.1.x86_64                                                    2/11
  Running scriptlet: containerd-2.0.6-1.amzn2023.0.1.x86_64                                                    2/11
  Installing       : pigz-2.5-1.amzn2023.0.3.x86_64                                                            3/11
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                      4/11
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                 5/11
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                        6/11
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                 7/11
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                  8/11
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                  8/11
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                       9/11
  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                            10/11
  Installing       : container-selinux-3:2.233.0-1.amzn2023.noarch                                            10/11
  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                            10/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                      11/11
  Installing       : docker-25.0.8-1.amzn2023.0.6.x86_64                                                      11/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                      11/11
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                            11/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                      11/11
  Verifying        : container-selinux-3:2.233.0-1.amzn2023.noarch                                             1/11
  Verifying        : containerd-2.0.6-1.amzn2023.0.1.x86_64                                                    2/11
  Verifying        : docker-25.0.8-1.amzn2023.0.6.x86_64                                                       3/11
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                 4/11
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                  5/11
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                       6/11
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                        7/11
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                 8/11
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                      9/11
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                           10/11
  Verifying        : runc-1.2.6-1.amzn2023.0.1.x86_64                                                         11/11

Installed:
  container-selinux-3:2.233.0-1.amzn2023.noarch           containerd-2.0.6-1.amzn2023.0.1.x86_64         docker-25.0.8-1.amzn2023.0.6.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-1.2.6-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-33-28 ~]$ |
```

4. **Verify Docker installation**

   docker –version

You should see something like Docker version 20.x.x.



```
  Installing       : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                          4/11
  Installing       : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                     5/11
  Installing       : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                            6/11
  Installing       : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                     7/11
  Installing       : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                      8/11
  Running scriptlet: iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                      8/11
  Installing       : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                           9/11
  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                                10/11
  Installing       : container-selinux-3:2.233.0-1.amzn2023.noarch                                                10/11
  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                                10/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                          11/11
  Installing       : docker-25.0.8-1.amzn2023.0.6.x86_64                                                          11/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                          11/11
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /usr/lib/systemd/system/docker.socket.

  Running scriptlet: container-selinux-3:2.233.0-1.amzn2023.noarch                                                11/11
  Running scriptlet: docker-25.0.8-1.amzn2023.0.6.x86_64                                                          11/11
  Verifying        : container-selinux-3:2.233.0-1.amzn2023.noarch                                                 1/11
  Verifying        : containerd-2.0.6-1.amzn2023.0.1.x86_64                                                        2/11
  Verifying        : docker-25.0.8-1.amzn2023.0.6.x86_64                                                           3/11
  Verifying        : iptables-libs-1.8.8-3.amzn2023.0.2.x86_64                                                     4/11
  Verifying        : iptables-nft-1.8.8-3.amzn2023.0.2.x86_64                                                      5/11
  Verifying        : libcgroup-3.0-1.amzn2023.0.1.x86_64                                                           6/11
  Verifying        : libnetfilter_conntrack-1.0.8-2.amzn2023.0.2.x86_64                                            7/11
  Verifying        : libnfnetlink-1.0.1-19.amzn2023.0.2.x86_64                                                     8/11
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                          9/11
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                               10/11
  Verifying        : runc-1.2.6-1.amzn2023.0.1.x86_64                                                             11/11

Installed:
  container-selinux-3:2.233.0-1.amzn2023.noarch           containerd-2.0.6-1.amzn2023.0.1.x86_64          docker-25.0.8-1.amzn2023.0.6.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-1.2.6-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-33-28 ~]$ docker --version
Docker version 25.0.8, build 0bab007
[ec2-user@ip-172-31-33-28 ~]$ sudo amazon-linux-extras enable docker
sudo: amazon-linux-extras: command not found
[ec2-user@ip-172-31-33-28 ~]$
```

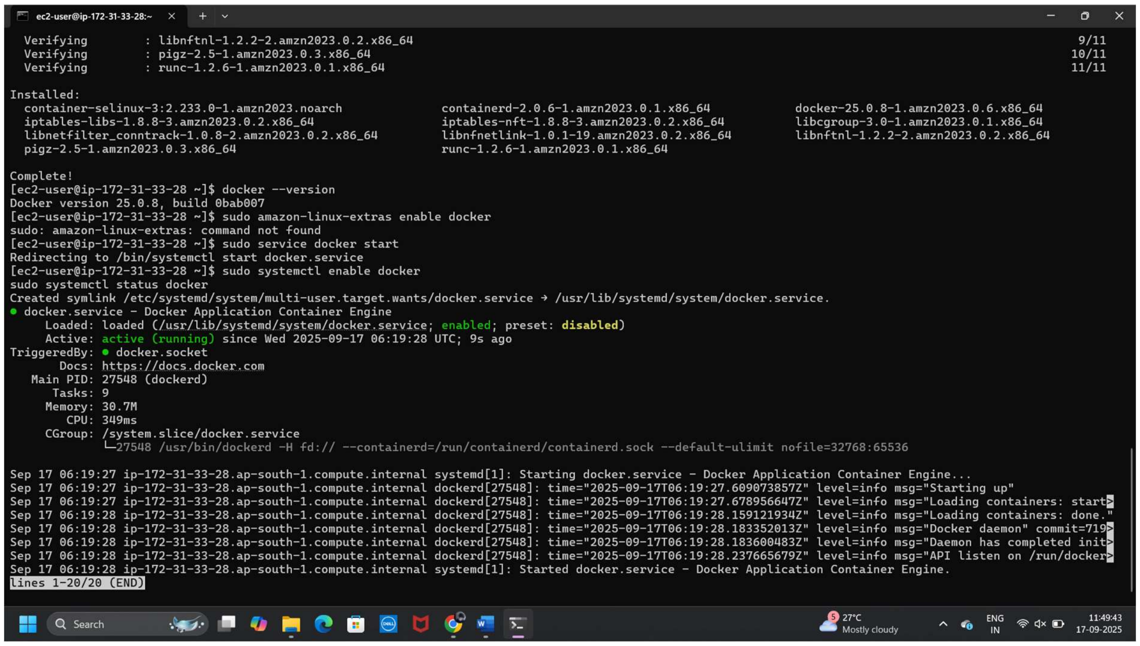
5. **Start and enable Docker service**

   sudo service docker start

   sudo systemctl enable docker

   sudo systemctl status docker

   status should show **active (running)**.



```
  Verifying        : libnftnl-1.2.2-2.amzn2023.0.2.x86_64                                                          9/11
  Verifying        : pigz-2.5-1.amzn2023.0.3.x86_64                                                               10/11
  Verifying        : runc-1.2.6-1.amzn2023.0.1.x86_64                                                             11/11

Installed:
  container-selinux-3:2.233.0-1.amzn2023.noarch           containerd-2.0.6-1.amzn2023.0.1.x86_64          docker-25.0.8-1.amzn2023.0.6.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-1.2.6-1.amzn2023.0.1.x86_64

Complete!
[ec2-user@ip-172-31-33-28 ~]$ docker --version
Docker version 25.0.8, build 0bab007
[ec2-user@ip-172-31-33-28 ~]$ sudo amazon-linux-extras enable docker
sudo: amazon-linux-extras: command not found
[ec2-user@ip-172-31-33-28 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-33-28 ~]$ sudo systemctl enable docker
sudo systemctl status docker
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /usr/lib/systemd/system/docker.service.
● docker.service - Docker Application Container Engine
     Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: disabled)
     Active: active (running) since Wed 2025-09-17 06:19:28 UTC; 9s ago
TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
   Main PID: 27548 (dockerd)
      Tasks: 9
     Memory: 30.7M
        CPU: 349ms
     CGroup: /system.slice/docker.service
             └─27548 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Sep 17 06:19:27 ip-172-31-33-28.ap-south-1.compute.internal systemd[1]: Starting docker.service - Docker Application Container Engine...
Sep 17 06:19:27 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:27.609073857Z" level=info msg="Starting up"
Sep 17 06:19:27 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:27.678956647Z" level=info msg="Loading containers: start>
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:28.159121934Z" level=info msg="Loading containers: done."
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:28.183352013Z" level=info msg="Docker daemon" commit=719>
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:28.183600483Z" level=info msg="Daemon has completed init>
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:28.237665679Z" level=info msg="API listen on /run/docker>
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal systemd[1]: Started docker.service - Docker Application Container Engine.
lines 1-20/20 (END)
```

6. **Apply group changes**

   Exit and reconnect to refresh groups:

   Exit

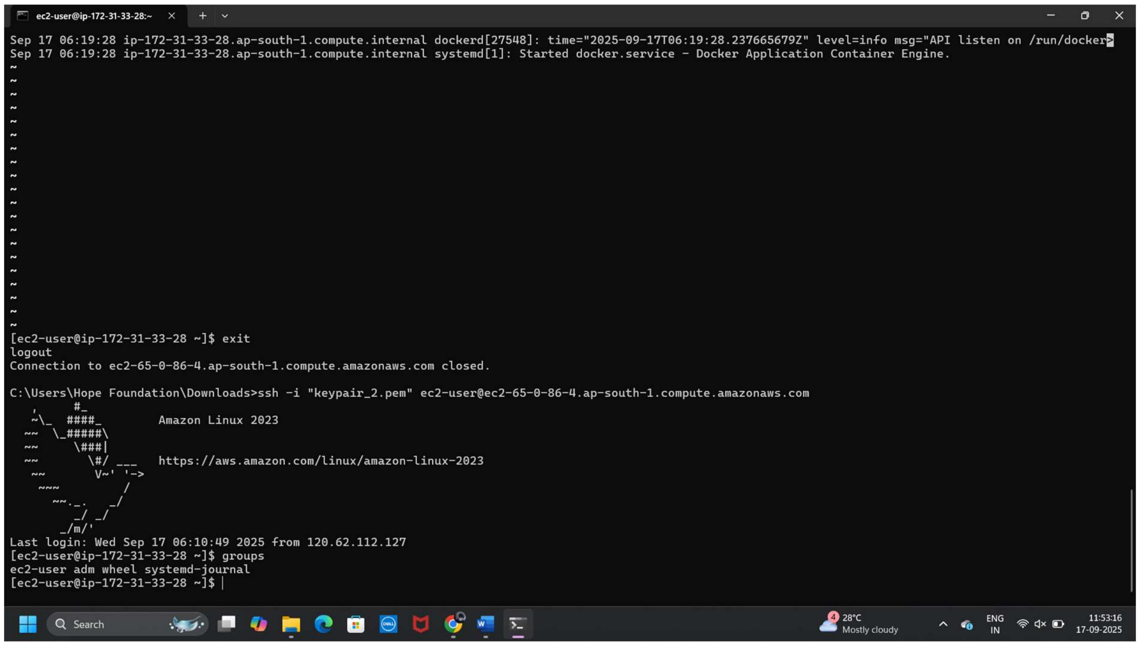
   Reconnect SSH:

   ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com

   Check groups:

   Groups

   You should now see docker in the list.



```
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal dockerd[27548]: time="2025-09-17T06:19:28.237665679Z" level=info msg="API listen on /run/docker>
Sep 17 06:19:28 ip-172-31-33-28.ap-south-1.compute.internal systemd[1]: Started docker.service - Docker Application Container Engine.
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
[ec2-user@ip-172-31-33-28 ~]$ exit
logout
Connection to ec2-65-0-86-4.ap-south-1.compute.amazonaws.com closed.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Sep 17 06:10:49 2025 from 120.62.112.127
[ec2-user@ip-172-31-33-28 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-33-28 ~]$ |
```

Step 4: Run Apache in Docker (on your EC2)

1. **Pull the official Apache HTTP Server image**

   sudo docker pull httpd:latest

2. **Run the Apache container on port 80**

   sudo docker run -d -p 80:80 --name my-apache-app httpd:latest

   -d → detached mode (runs in background)

   -p 80:80 → maps **container port 80** to **host EC2 port 80**

   --name my-apache-app → names the container

3. **Verify the container is running**

Shaikh Ateeb Ahmed
17-09-2025

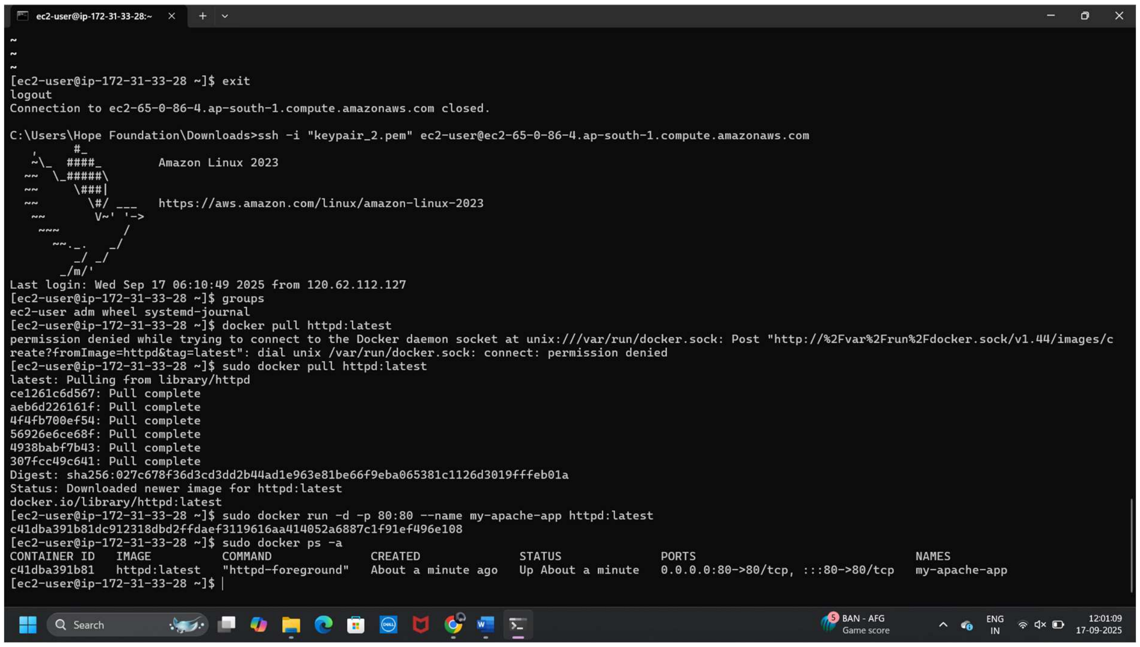
---

sudo docker ps -a

You should see something like:

CONTAINER ID   IMAGE   COMMAND   STATUS   PORTS   NAMES

abc123456789   httpd:latest   "httpd-foreground"   Up 2 minutes   0.0.0.0:80->80/tcp   my-apache-app



```
ec2-user@ip-172-31-33-28:~
~
~
~
[ec2-user@ip-172-31-33-28 ~]$ exit
logout
Connection to ec2-65-0-86-4.ap-south-1.compute.amazonaws.com closed.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Sep 17 06:10:49 2025 from 120.62.112.127
[ec2-user@ip-172-31-33-28 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-33-28 ~]$ docker pull httpd:latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/images/c
reate?fromImage=httpd&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-33-28 ~]$ sudo docker pull httpd:latest
latest: Pulling from library/httpd
ce1261c6d567: Pull complete
aeb6d226161f: Pull complete
4f4fb700ef54: Pull complete
56926e6ce68f: Pull complete
4938babf7b43: Pull complete
307fcc49c641: Pull complete
Digest: sha256:027c678f36d3cd3dd2b44ad1e963e81be66f9eba065381c1126d3019fffeb01a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
[ec2-user@ip-172-31-33-28 ~]$ sudo docker run -d -p 80:80 --name my-apache-app httpd:latest
c41dba391b81dc912318dbd2ffdaef3119616aa414052a6887c1f91ef496e108
[ec2-user@ip-172-31-33-28 ~]$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND              CREATED              STATUS               PORTS                                 NAMES
c41dba391b81   httpd:latest   "httpd-foreground"   About a minute ago   Up About a minute    0.0.0.0:80->80/tcp, :::80->80/tcp   my-apache-app
[ec2-user@ip-172-31-33-28 ~]$ |
```

Step 5: Customize the Web Page

1. **Enter the running container**

   sudo docker exec -it my-apache-app /bin/bash

2. **Go to Apache's web root**

   cd /usr/local/apache2/htdocs

3. **Create (or overwrite) index.html**

   echo "MyWebsite - Running on Apache inside Docker" > index.html

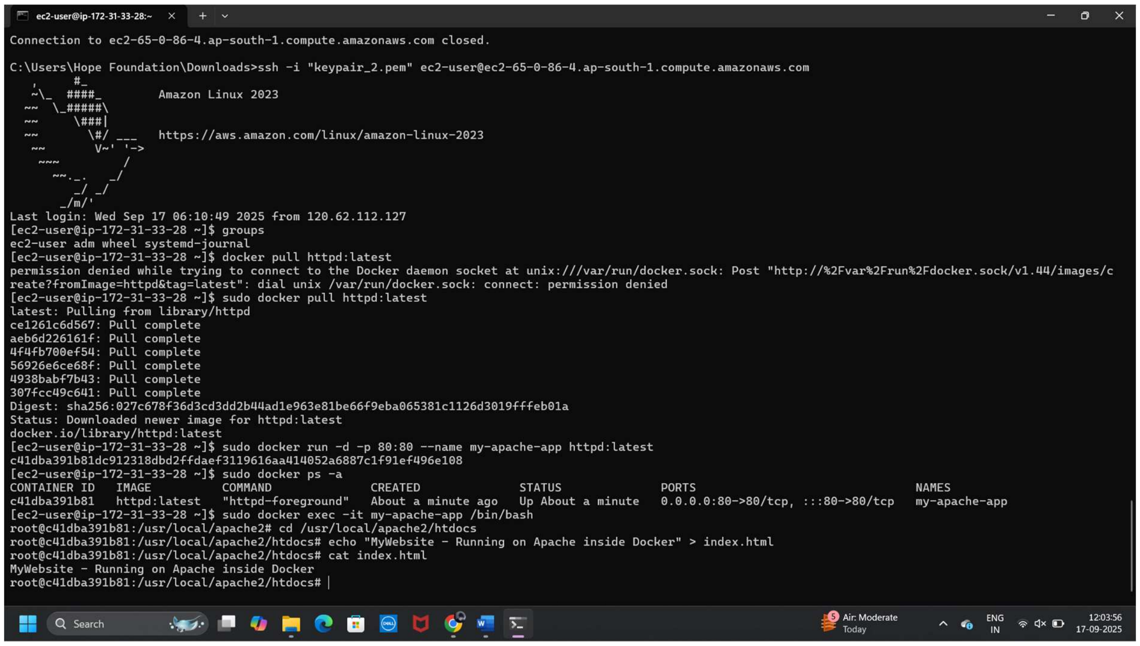
4. **Verify the file**

   cat index.html

   You should see:

   MyWebsite - Running on Apache inside Docker

Shaikh Ateeb Ahmed
17-09-2025



```
Connection to ec2-65-0-86-4.ap-south-1.compute.amazonaws.com closed.

C:\Users\Hope Foundation\Downloads>ssh -i "keypair_2.pem" ec2-user@ec2-65-0-86-4.ap-south-1.compute.amazonaws.com
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Wed Sep 17 06:10:49 2025 from 120.62.112.127
[ec2-user@ip-172-31-33-28 ~]$ groups
ec2-user adm wheel systemd-journal
[ec2-user@ip-172-31-33-28 ~]$ docker pull httpd:latest
permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://%2Fvar%2Frun%2Fdocker.sock/v1.44/images/c
reate?fromImage=httpd&tag=latest": dial unix /var/run/docker.sock: connect: permission denied
[ec2-user@ip-172-31-33-28 ~]$ sudo docker pull httpd:latest
latest: Pulling from library/httpd
ce1261c6d567: Pull complete
aeb6d226161f: Pull complete
4f4fb700ef54: Pull complete
56926e6ce68f: Pull complete
4938babf7b43: Pull complete
307fcc49c641: Pull complete
Digest: sha256:027c678f36d3cd3dd2b44ad1e963e81be66f9eba065381c1126d3019fffeb01a
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
[ec2-user@ip-172-31-33-28 ~]$ sudo docker run -d -p 80:80 --name my-apache-app httpd:latest
c41dba391b81dc912318dbd2ffdaef3119616aa414052a6887c1f91ef496e108
[ec2-user@ip-172-31-33-28 ~]$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND              CREATED              STATUS              PORTS                                NAMES
c41dba391b81   httpd:latest   "httpd-foreground"   About a minute ago   Up About a minute   0.0.0.0:80->80/tcp, :::80->80/tcp   my-apache-app
[ec2-user@ip-172-31-33-28 ~]$ sudo docker exec -it my-apache-app /bin/bash
root@c41dba391b81:/usr/local/apache2# cd /usr/local/apache2/htdocs
root@c41dba391b81:/usr/local/apache2/htdocs# echo "MyWebsite - Running on Apache inside Docker" > index.html
root@c41dba391b81:/usr/local/apache2/htdocs# cat index.html
MyWebsite - Running on Apache inside Docker
root@c41dba391b81:/usr/local/apache2/htdocs# |
```

5. **Exit the container**

   Exit

Step 6: Test the Web App

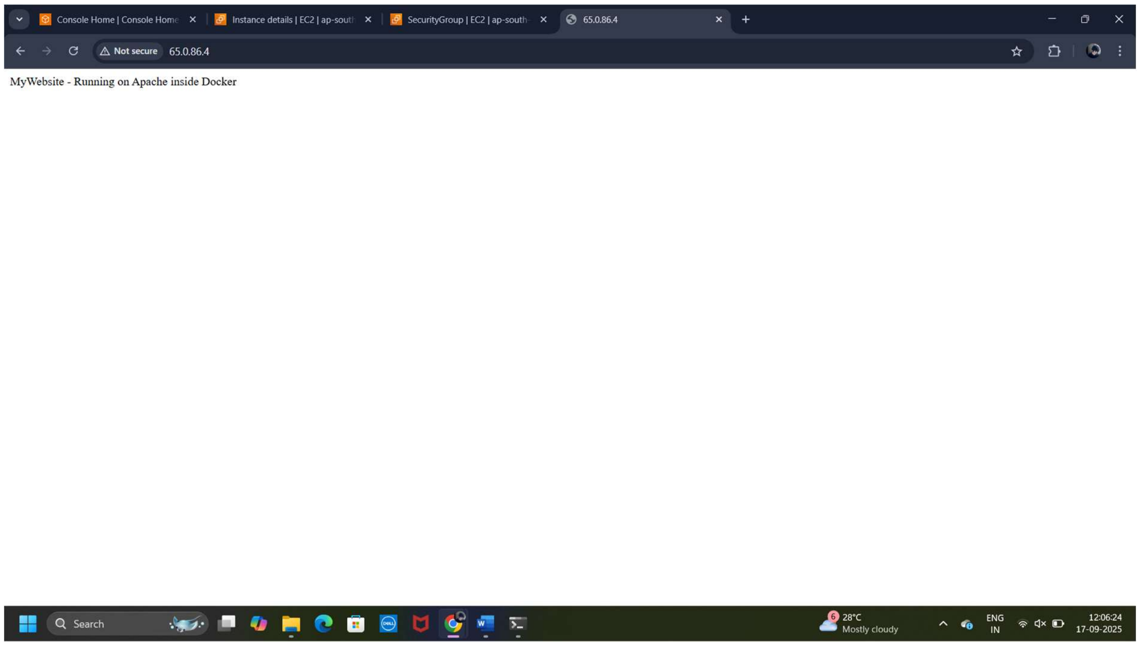
1. Open in browser:

   http://<public ip>

   http://65.0.86.4

   You should see: MyWebsite - Running on Apache inside Docker

Shaikh Ateeb Ahmed
17-09-2025


Console Home | Console Home
Instance details | EC2 | ap-south
SecurityGroup | EC2 | ap-south
65.0.86.4
Not secure
65.0.86.4
Search
28°C
Mostly cloudy
ENG
IN
12:06:24
17-09-2025

MyWebsite - Running on Apache inside Docker