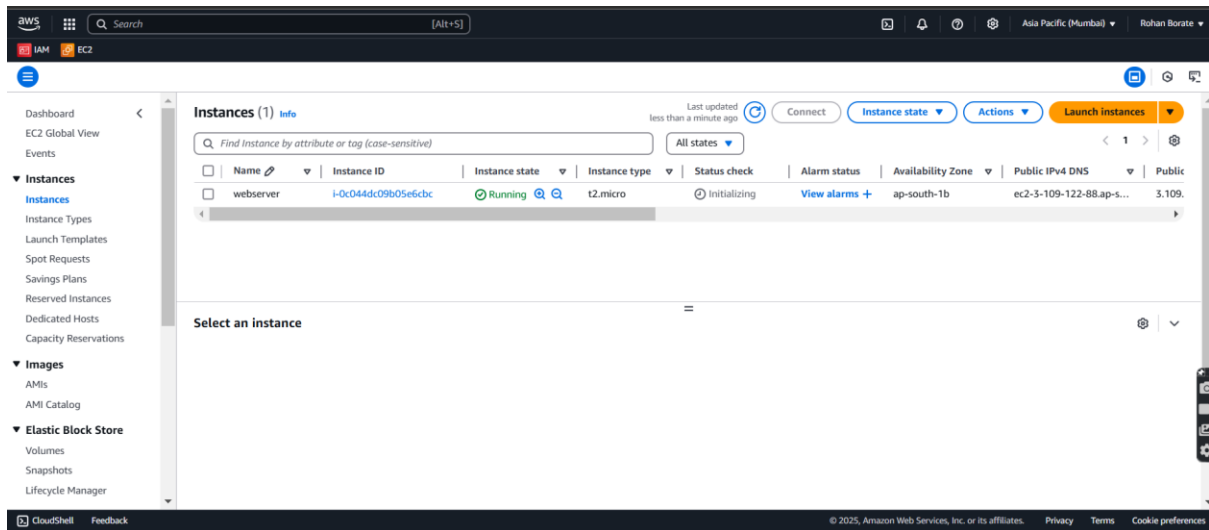


AMAZON MACHINE IMAGE (AMI)

Create EC2 instance



Connect to instance and install apache2 in it , for aws linux instance it termed as httpd

```
[ec2-user@ip-172-31-11-109 ~]$ sudo yum update
Last metadata expiration check: 0:09:05 ago on Sat Jan  4 13:14:48 2025.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-11-109 ~]$ yum install httpd
Error: This command has to be run with superuser privileges (under the root user on most systems).
[ec2-user@ip-172-31-11-109 ~]$ sudo -i
[root@ip-172-31-11-109 ~]# yum install httpd
Last metadata expiration check: 0:09:36 ago on Sat Jan  4 13:14:48 2025.
Dependencies resolved.
=====
Package                        Architecture      Version           Size
=====
Installing:
httpd                          x86_64            2.4.62-1.amzn2023 48 k
Installing dependencies:
apr                            x86_64            1.7.5-1.amzn2023.0.2 139 k
apr-util                       x86_64            1.6.3-1.amzn2023.0.1 98 k
generic-logos-httpd           noarch            18.0.0-12.amzn2023.0.3 19 k
httpd-core                     x86_64            2.4.62-1.amzn2023 1.4 M
httpd-filesystem               noarch            2.4.62-1.amzn2023 14 k
httpd-tools                    x86_64            2.4.62-1.amzn2023 81 k
libbrotli                      x86_64            1.0.9-4.amzn2023.0.2 315 k
mailcap                        noarch            2.1.49-3.amzn2023.0.3 33 k
Installing weak dependencies:
apr-util-openssl              x86_64            1.6.3-1.amzn2023.0.1 17 k
mod_http2                     x86_64            2.0.27-1.amzn2023.0.3 166 k
mod_lua                        x86_64            2.4.62-1.amzn2023 61 k
=====
Transaction Summary
=====
Install 12 Packages
```

Start the apache2 server

```
[root@ip-172-31-11-109 ~]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@ip-172-31-11-109 ~]# systemctl enable httpd
Created symlink /etc/systemd/system/multi-user.target.wants/httpd.service → /usr/lib/systemd/system/httpd.service.
[root@ip-172-31-11-109 ~]# systemctl status httpd
○ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: inactive (dead)
     Docs: man:httpd.service(8)
[root@ip-172-31-11-109 ~]# systemctl start httpd
[root@ip-172-31-11-109 ~]# systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
   Active: active (running) since Sat 2025-01-04 13:26:23 UTC; 5s ago
     Docs: man:httpd.service(8)
  Main PID: 26287 (httpd)
    Status: "Started, listening on: port 80"
    Tasks: 177 (limit: 1111)
   Memory: 12.9M
      CPU: 51ms
   CGroup: /system.slice/httpd.service
           └─26287 /usr/sbin/httpd -DFOREGROUND
             └─26304 /usr/sbin/httpd -DFOREGROUND
               └─26306 /usr/sbin/httpd -DFOREGROUND
                 └─26307 /usr/sbin/httpd -DFOREGROUND
                   └─26308 /usr/sbin/httpd -DFOREGROUND

Jan 04 13:26:22 ip-172-31-11-109.ap-south-1.compute.internal systemd[1]: Starting httpd.service - The Apache HTTP Server...
Jan 04 13:26:23 ip-172-31-11-109.ap-south-1.compute.internal systemd[1]: Started httpd.service - The Apache HTTP Server.
Jan 04 13:26:23 ip-172-31-11-109.ap-south-1.compute.internal httpd[26287]: Server configured, listening on: port 80
```

Check in the webserver is accessible through browser , apache works on port no 80 ,so we have to enable this port In security groups inbound rules

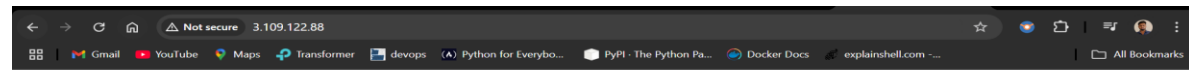
The screenshot shows the AWS Management Console interface for editing inbound rules on a security group. The breadcrumb navigation indicates the path: EC2 > Security Groups > sg-03f0bb35ad4f34be0 - launch-wizard-3 > Edit inbound rules. The main heading is 'Edit inbound rules' with an 'info' icon. Below this, a note states: 'Inbound rules control the incoming traffic that's allowed to reach the instance.'

The 'Inbound rules' section contains a table with the following columns: Security group rule ID, Type, Protocol, Port range, Source, and Description - optional. There are two rules listed:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sg-0622c8e81ccf4db4	SSH	TCP	22	Custom	
-	HTTP	TCP	80	Anywh...	

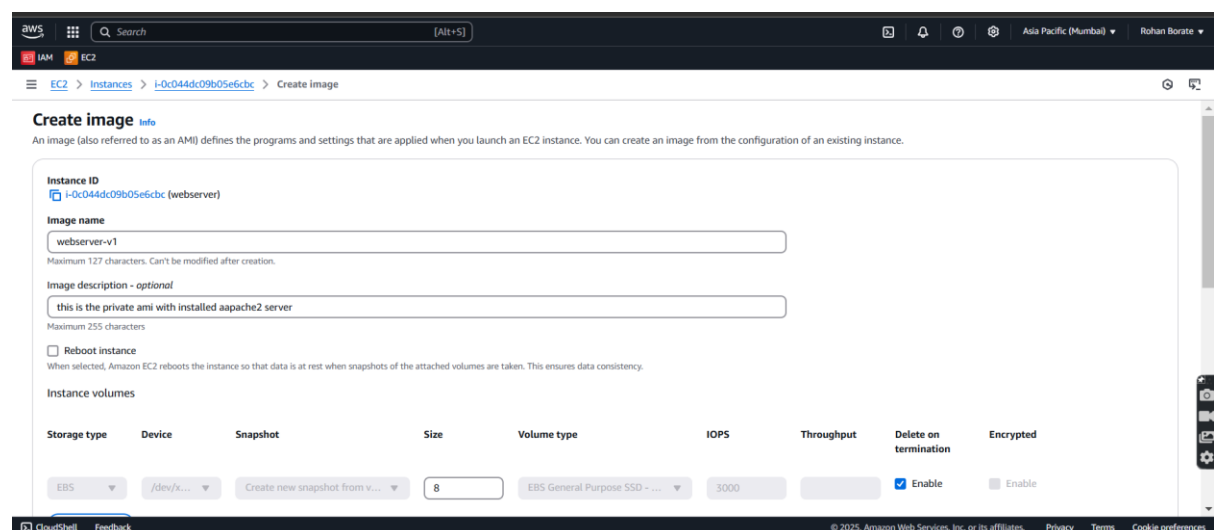
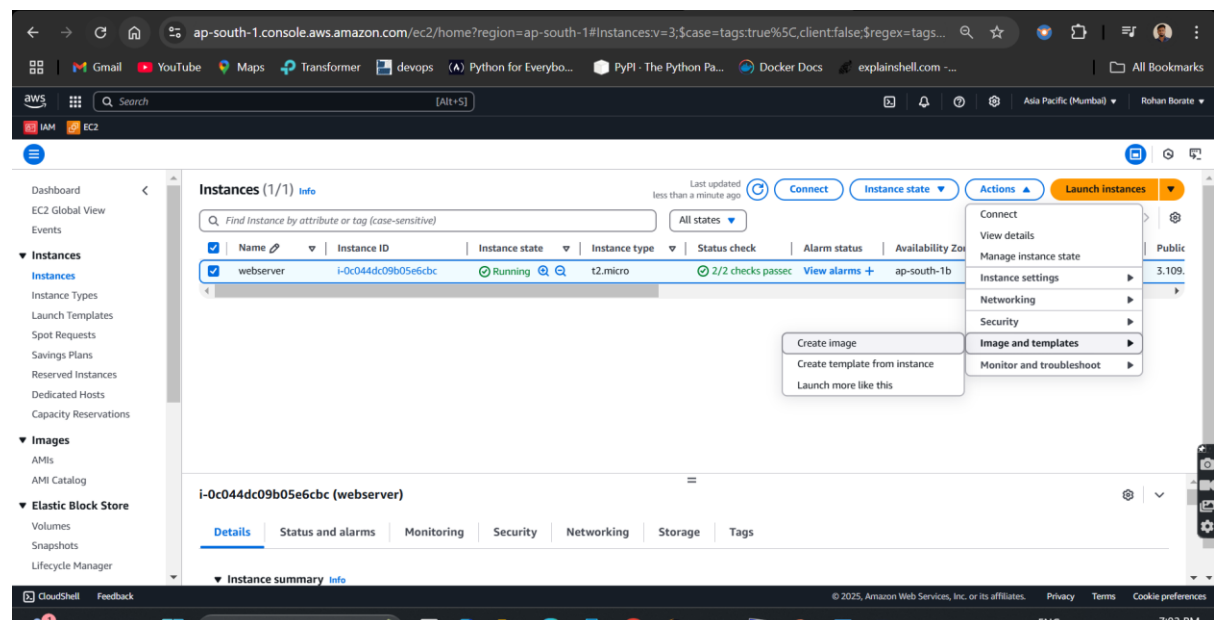
Below the table, there is an 'Add rule' button. At the bottom of the console, a yellow warning banner reads: 'Rules with source of 0.0.0.0/0 or ::0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' Navigation buttons at the bottom right include 'Cancel', 'Preview changes', and 'Save rules'.

It is working now

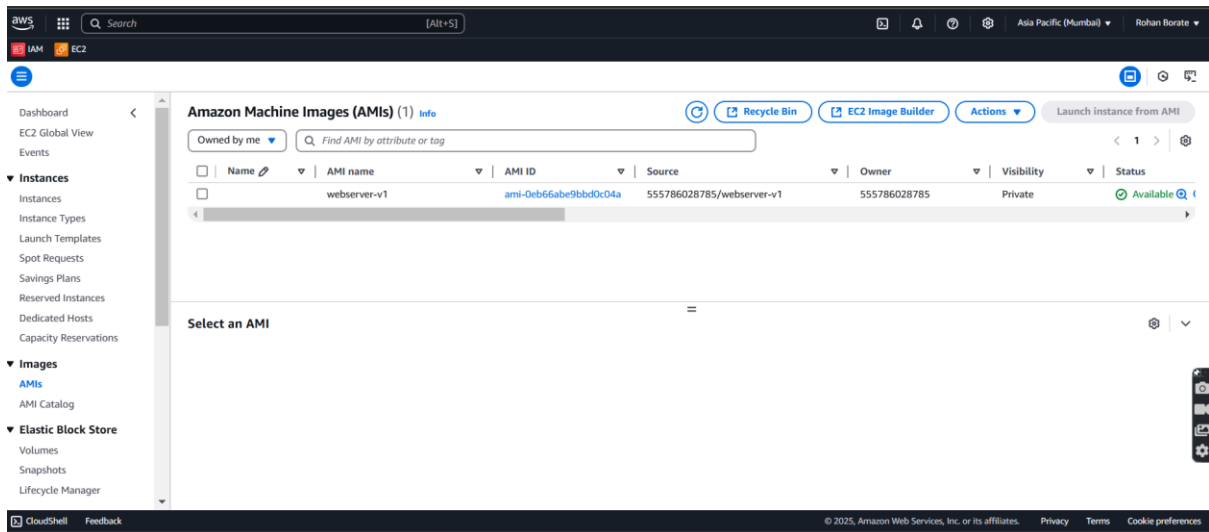


It works!

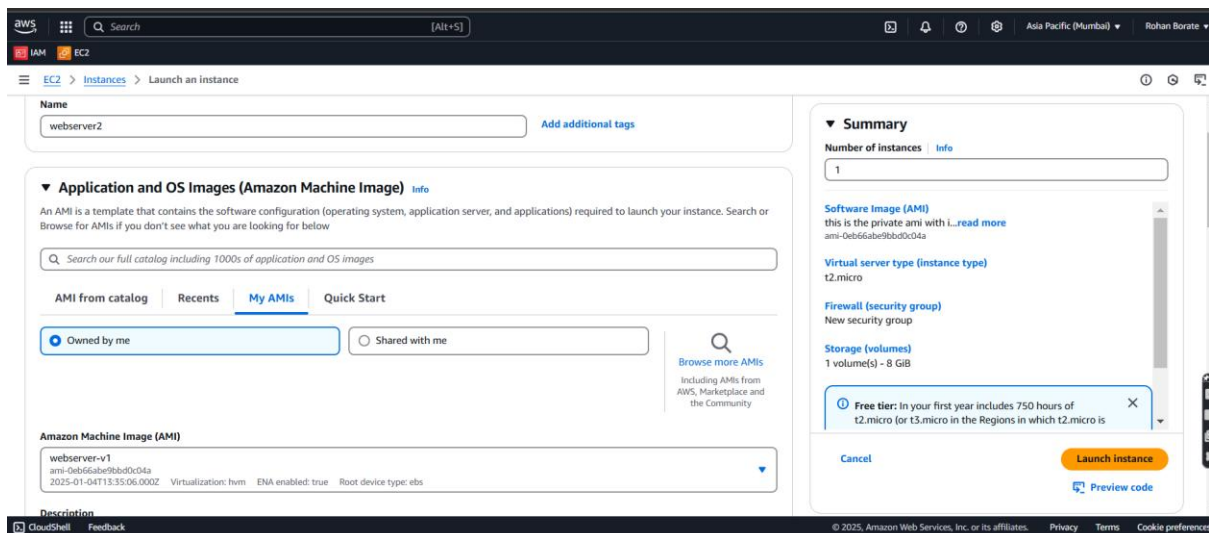
Now create the AMI for this instance ,



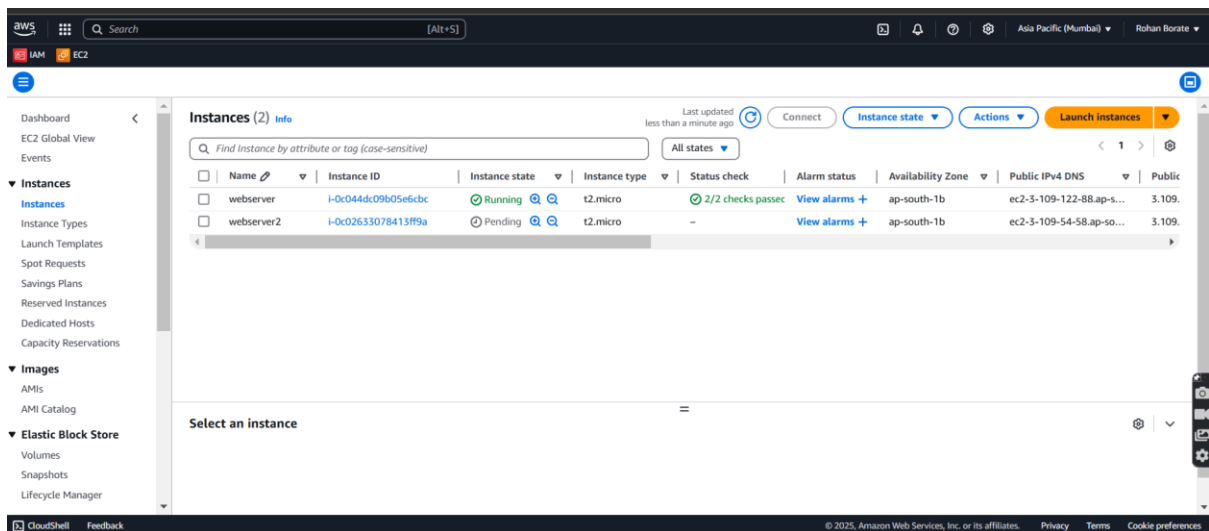
Check the image → AMI , that ami is available



Now create the another instance using this AMI , named as webserver2 , select the created AMI (webserver-v1) and launch the instance



Instance is created



Then set the inbound rules and then check that the apache server is running or not

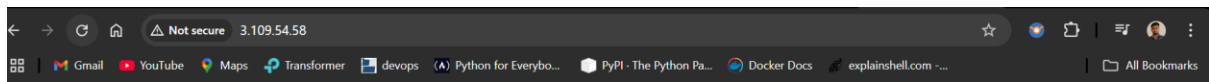
URL → public ip of instance : 80 (http port)

apach2 is running , because we are using the AMI we have created this instance ,

In AMI we already have an apache2 server installed.

Basically AMI is useful to create an multiple instances with the same configuration,

And also for the backup and recovery purpose



It works!

Note →

AMI's are the region dependent so if we create AMI in one region we cannot use this in another region , but it can be explicitly copied to any specific region

Steps to copy AMI → choose AMI want to copy → action → copy AMI → select region