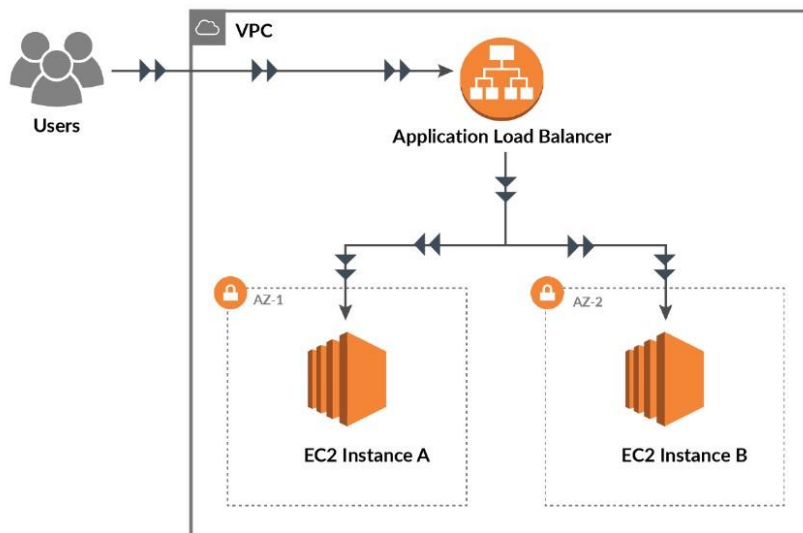
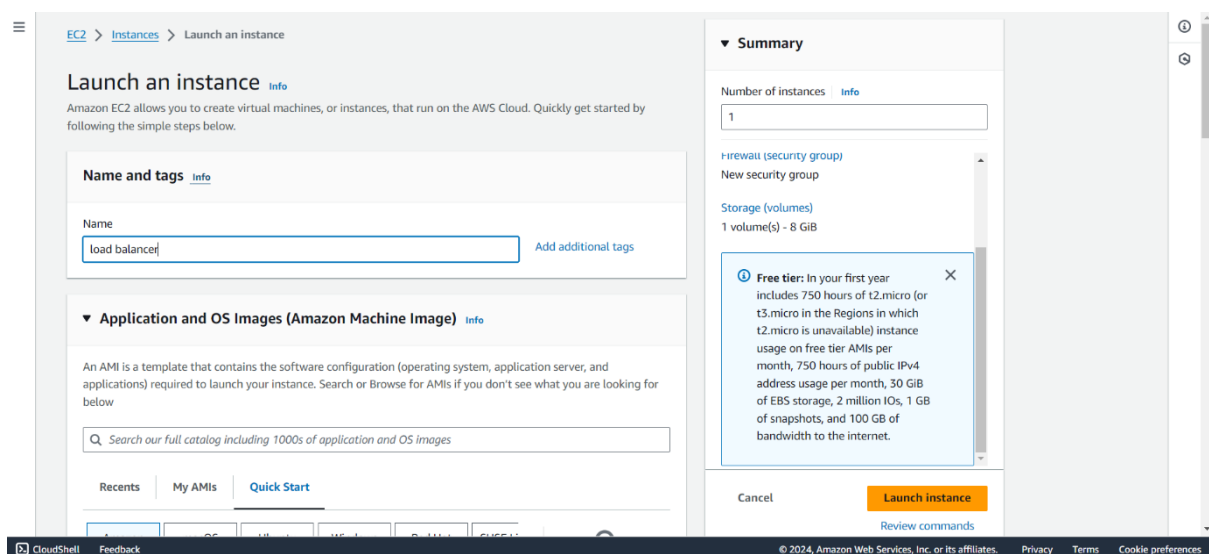


# TITLE :CREATING CLASSIC LOAD BALANCER WITH EC2 INSTANCES

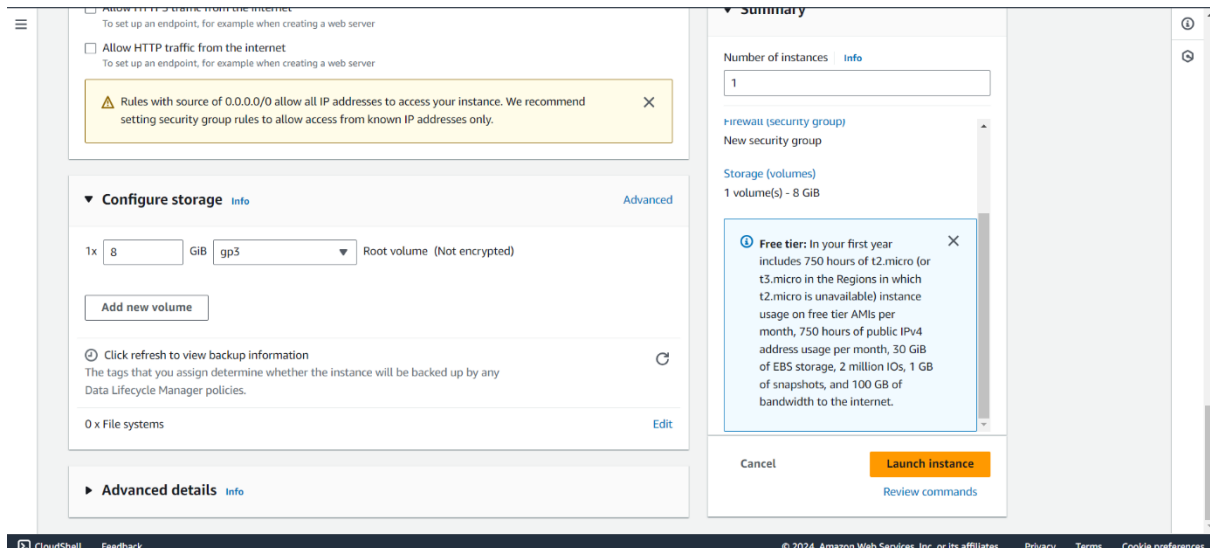
- ARCHITECTURE DIAGRAM



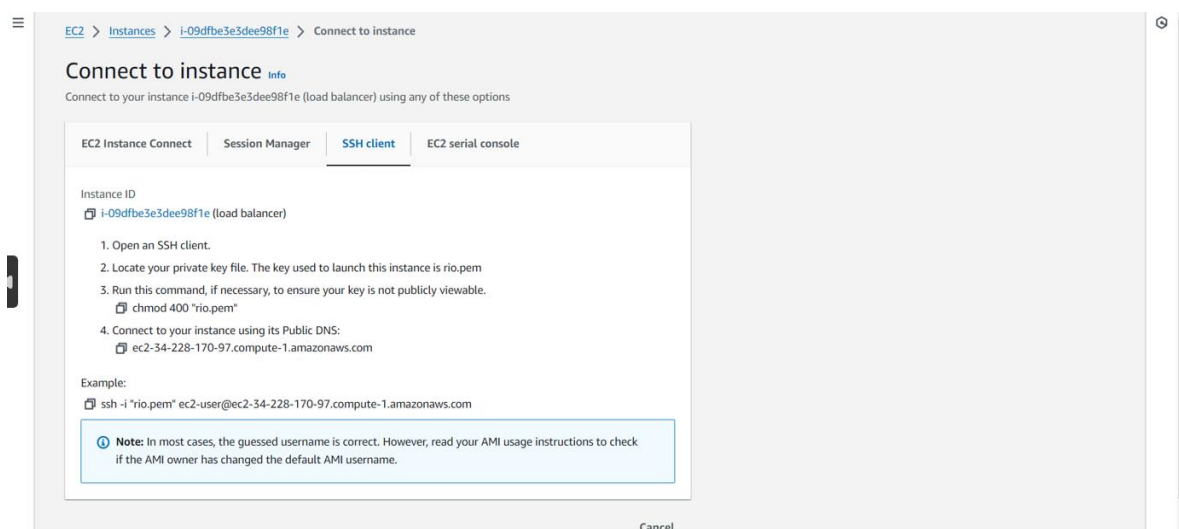
## STEP 1 : Create EC2 with Linux



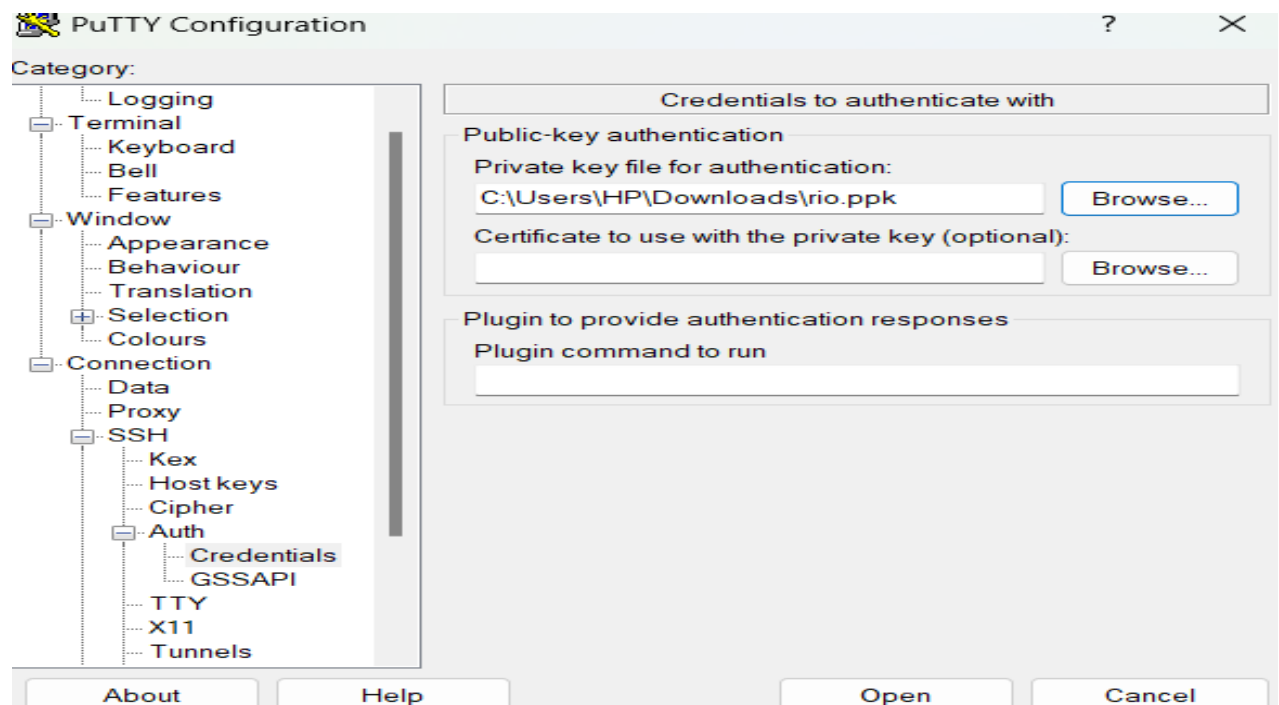
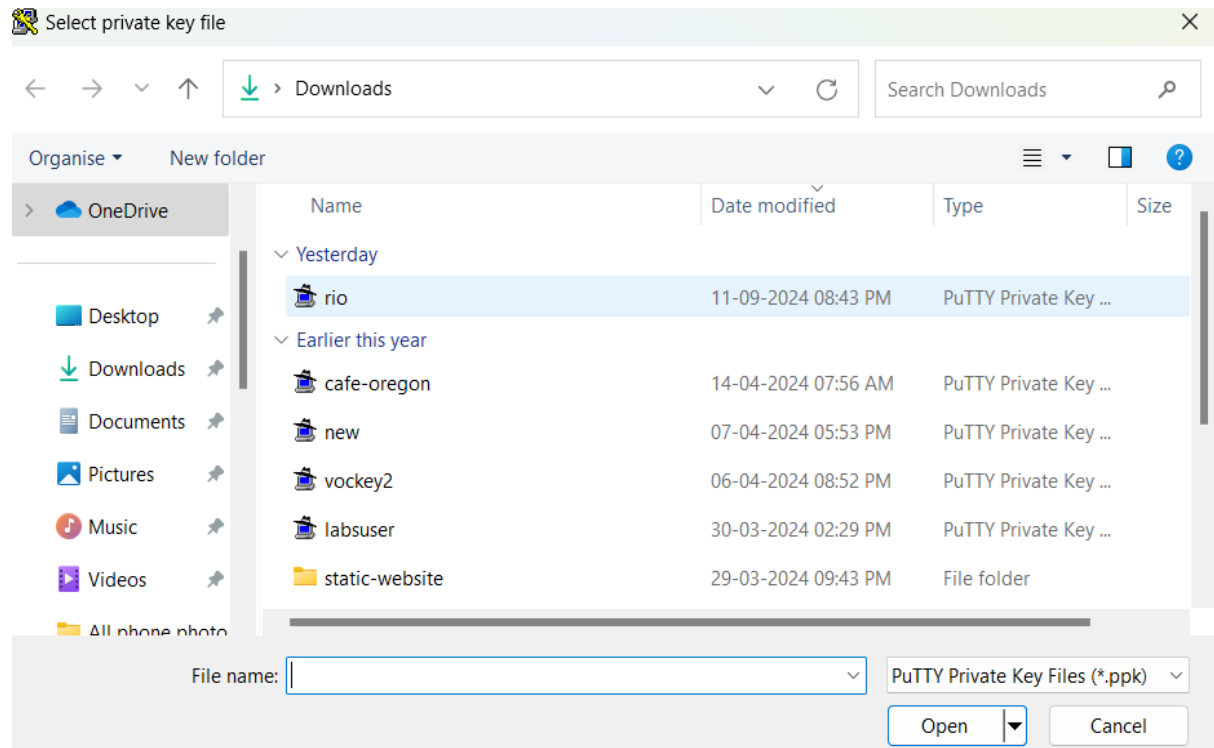
## STEP 2: Configure Storage and Launch Instances



STEP 3: Connect to instance with apache server in putty and copy the DNS paste in putty



STEP 4: Download the keypair in ec2 instance and paste with putty



STEP 5: Open putty login ec2 -user instances start the httpd service and check the Status of the server

1. Create the Apache HTTP server
2. Change the rootuser so it will run

```
login as: ec2-user
* Authenticating with public key "r10"

#####
  _   _          _   _
 | | | |        | | | |
 | |_| |        | |_| |
 |  __/ |        |  __/ |
 |_| |_|        |_| |_|
 |_| |_|        |_| |_|

https://aws.amazon.com/linux/amazon-linux-2023

Last login: Wed Sep 11 16:32:26 2024 from 223.178.86.33
ec2-user@ip-172-31-29-102 ~]$ sudo service httpd start
Redirecting to /bin/systemctl start httpd.service
ec2-user@ip-172-31-29-102 ~]$ sudo service httpd status
Redirecting to /bin/systemctl status httpd.service
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-09-11 16:12:35 UTC; 11h ago
     Docs: man:httd.service(8)
   Main PID: 3010 (httpd)
   Status: "Total requests: 3027; Idle/Busy workers 100/0;Requests/sec: 0.0707; Bytes serv
   Tasks: 230 (limit: 1112)
   Memory: 24.8M
   CPU: 32.210s
   CGroup: /system.slice/httpd.service
           └─3010 /usr/sbin/httpd -DFOREGROUND
           └─3039 /usr/sbin/httpd -DFOREGROUND
           └─3060 /usr/sbin/httpd -DFOREGROUND
           └─3061 /usr/sbin/httpd -DFOREGROUND
           └─3062 /usr/sbin/httpd -DFOREGROUND
           └─5438 /usr/sbin/httpd -DFOREGROUND

Sep 11 16:12:34 ip-172-31-29-102.ec2.internal systemd[1]: Starting httpd.service - The Apache
Sep 11 16:12:35 ip-172-31-29-102.ec2.internal systemd[1]: Started httpd.service - The Apache
Sep 11 16:12:35 ip-172-31-29-102.ec2.internal httpd[3010]: Server configured, listening on:
lines 1-20/20 (END) ...skipping...
* httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Wed 2024-09-11 16:12:35 UTC; 11h ago
     Docs: man:httd.service(8)
   Main PID: 3010 (httpd)
   Status: "Total requests: 3027; Idle/Busy workers 100/0;Requests/sec: 0.0707; Bytes served/sec: 37 B/sec"
   Tasks: 230 (limit: 1112)
   Memory: 24.8M
   CPU: 32.210s
   CGroup: /system.slice/httpd.service
           └─3010 /usr/sbin/httpd -DFOREGROUND
```

**STEP 6:** Write basic Html code WELCOME TO SERVER 1 and save the code

```
ec2-user@ip-172-31-29-102:/var/www/html

html>
<head>
</head>
<body>
    <h1>WELCOME TO SERVER 1</h1>
</body>
/html>

index.html" [readonly] 121  85B  1  0-1  211
```

**STEP 7:** Check the Apache server running with given html code

## STEP 8 : Create another instance with AMI and launch instances

EC2 Dashboard × EC2 Global View Events Console-to-Code [Preview](#)

▼ Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations [New](#)

▼ Images AMIs AMI Catalog

▼ Elastic Block Store Volumes Snapshots

EC2 > Instances > i-06905930378011019 > Create image

### Create image [info](#)

An image (also referred to as an AMI) defines the programs and settings that are applied when you launch an EC2 instance. You can create an image from the configuration of an existing instance.

Instance ID  
i-06905930378011019 (AMI)

Image name  
AMI  
Maximum 127 characters. Can't be modified after creation.

Image description - optional  
Image description  
Maximum 255 characters

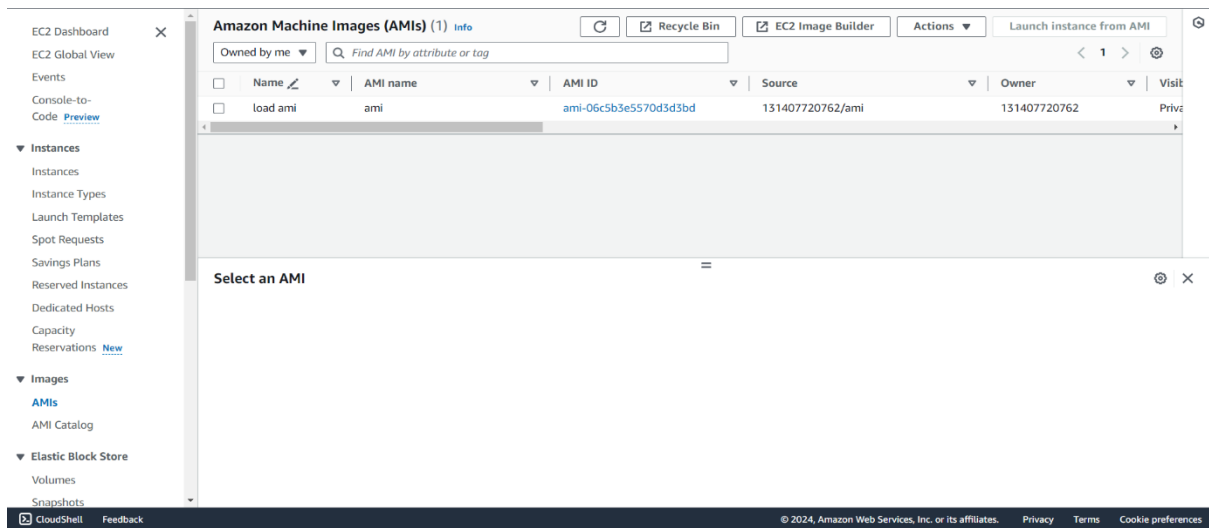
☒ Reboot instance  
When selected, Amazon EC2 reboots the instance so that data is at rest when snapshots of the attached volumes are taken. This ensures data consistency.

Instance volumes

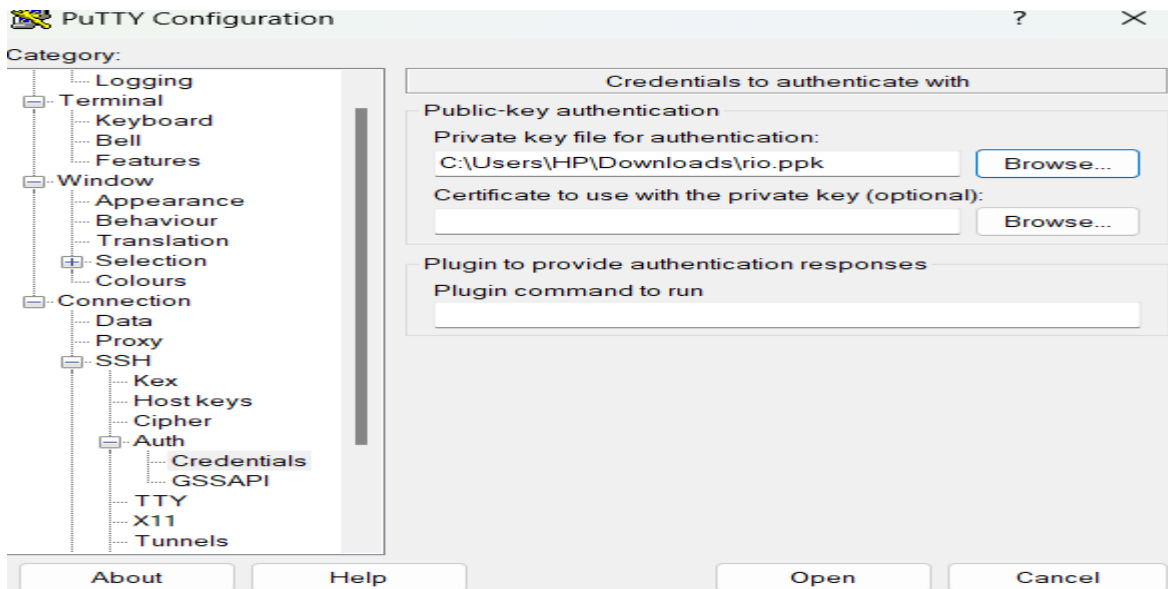
Storage type	Device	Snapshot	Size	Volume type	IOPS	Throughput	Delete on termination	Encrypted
--------------	--------	----------	------	-------------	------	------------	-----------------------	-----------

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

- Successfully create AMI AND LAUNCH WITH CONFIGURE



## STEP 9: CONNECT AMI INSTANCES WITH PUTTY



**STEP 10:** Open putty login ec2 -user instances start the httpd service and check the Status of the server

1. Create the Apache HTTP server
2. Change the rootuser so it will run



**STEP13:** Create Classic load balancer and to connect with two instances in load balancer

Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Create

Deployment, support for IPv6, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low latencies.

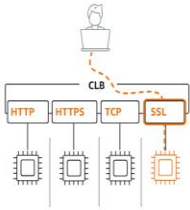
Create

SECURE: These appliances enable you to improve security, compliance, and policy controls.

Create

▼ Classic Load Balancer - previous generation

**Classic Load Balancer** [Info](#)



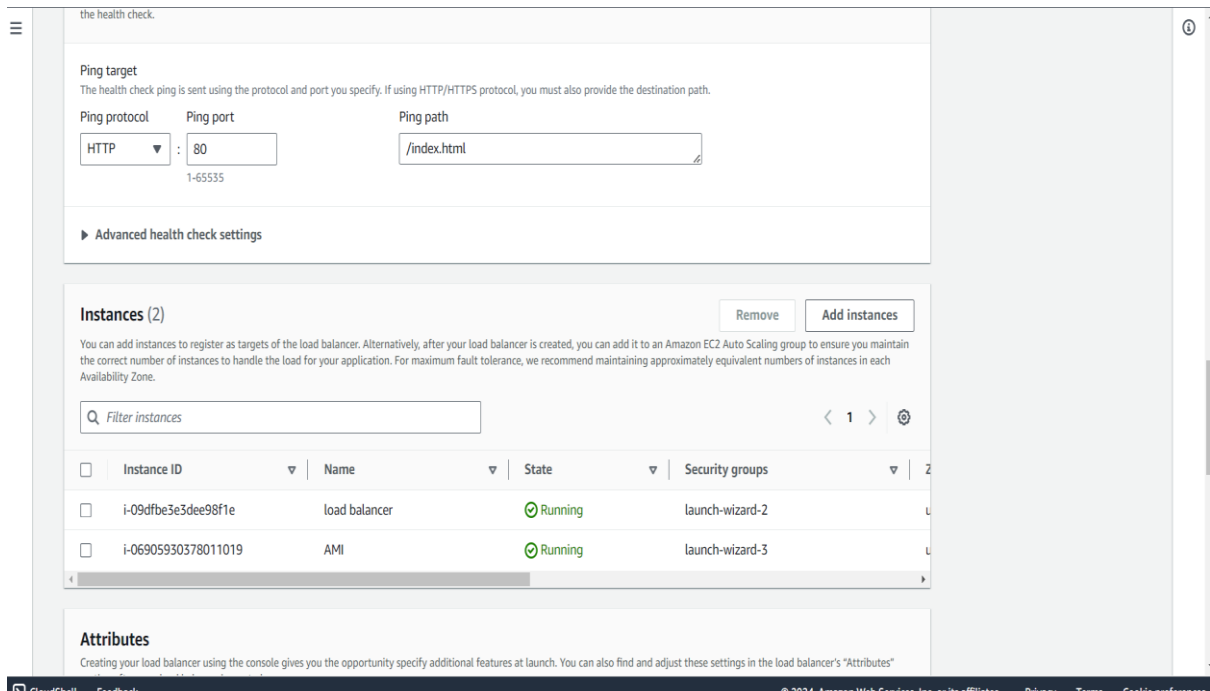
Choose a Classic Load Balancer when you have an existing application running in the EC2-Classical network.

Create

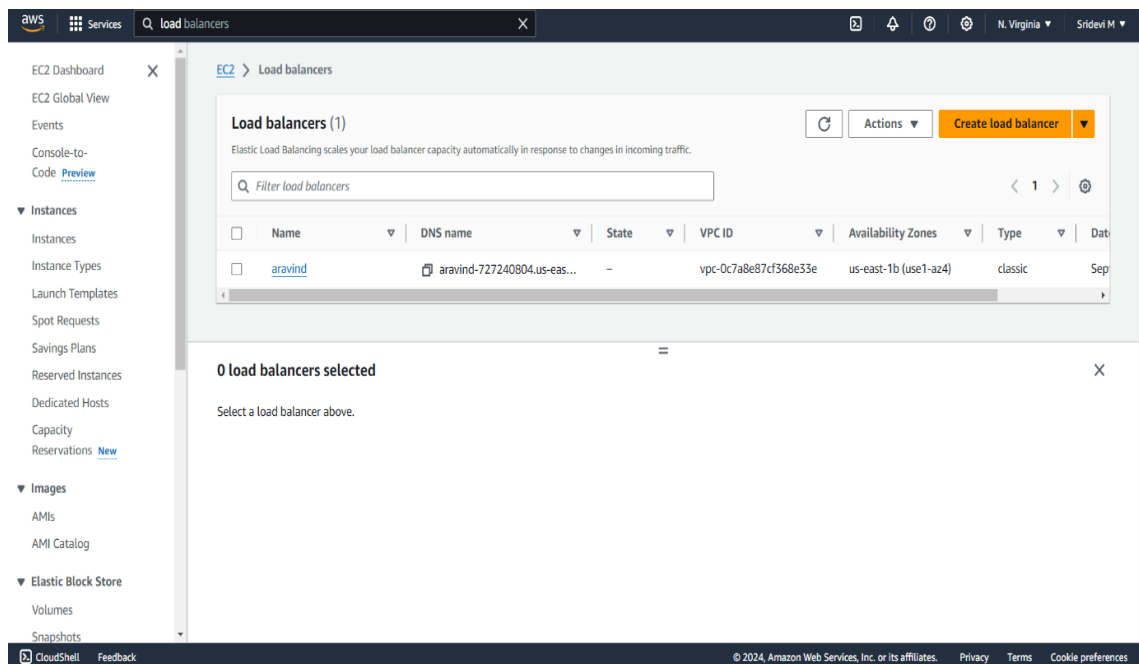
Close

- Create and launch with configure setting
- Add two instances which you create with ec2 instances

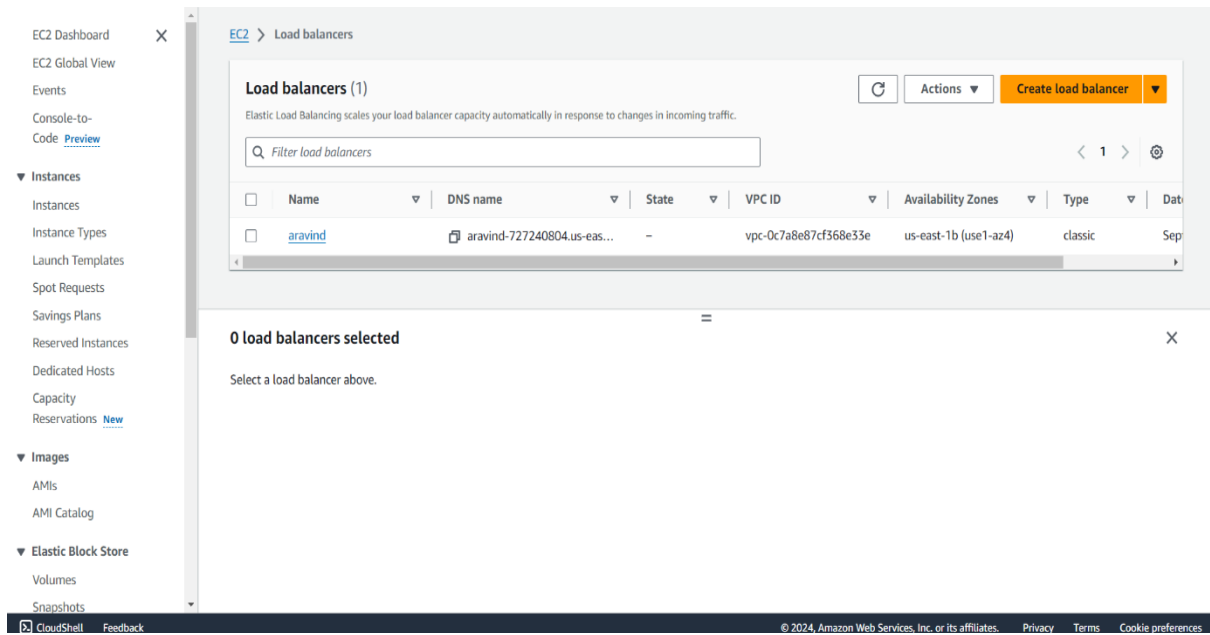




- **SUCCESSFULLY CREATE LOAD BALANCER WITH TWO INSTANCES**

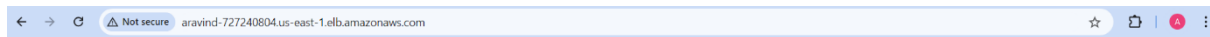


## STEP 14: COPY THE DNS AND RUN THE TWO INSTANCES



## OUTPUT

- SUCCESSFULLY RUN WITH TWO INSTANCES



WELCOME TO SERVER 1

**Welcome to SERVER 2**