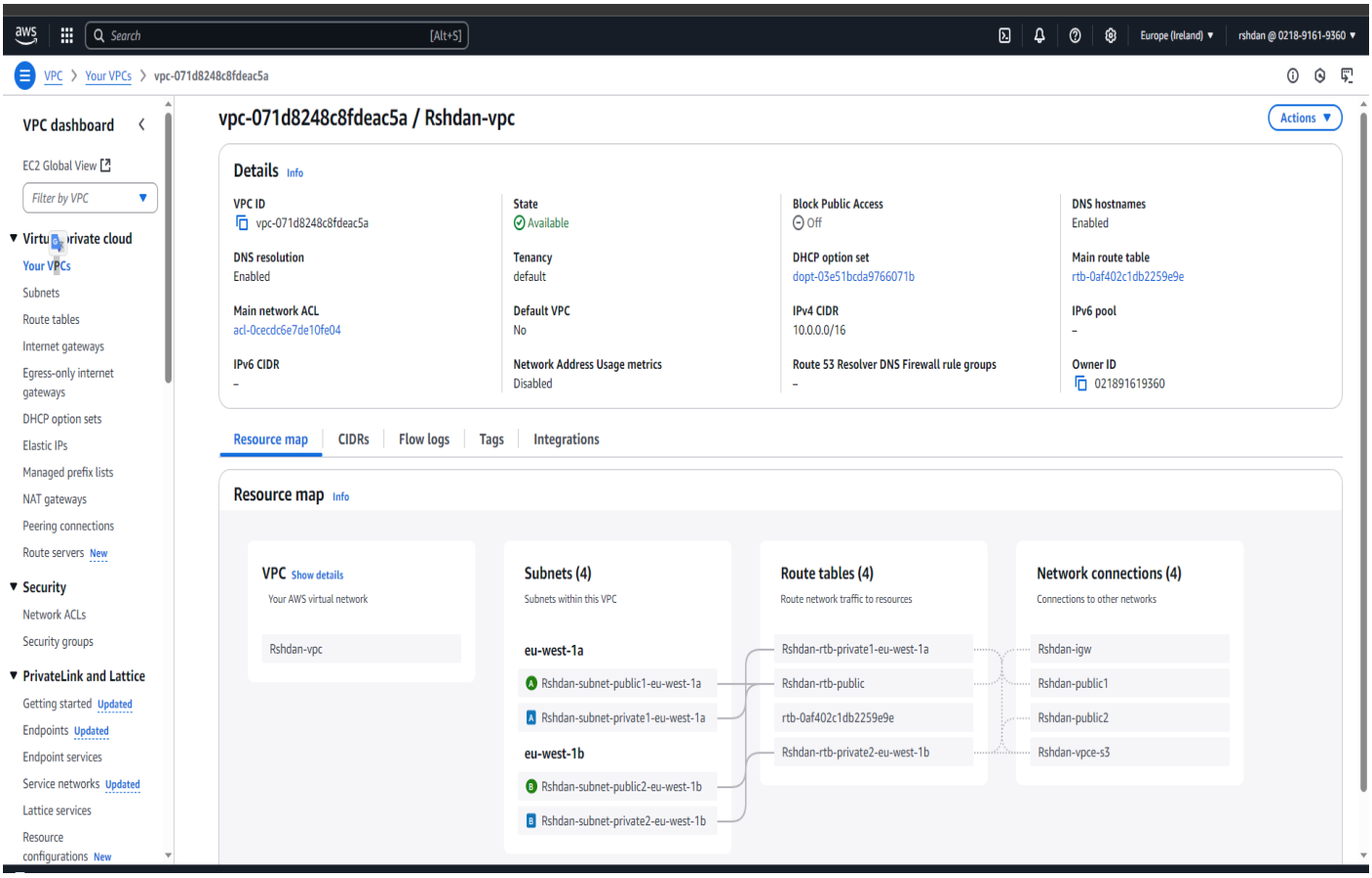
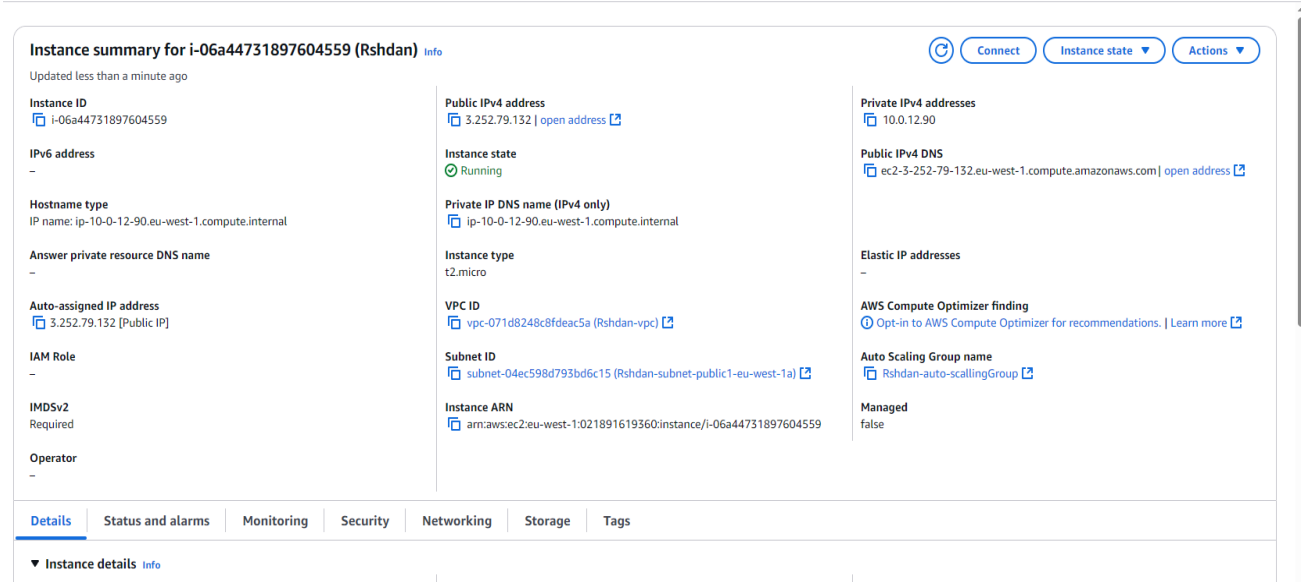
AWS LAB 3

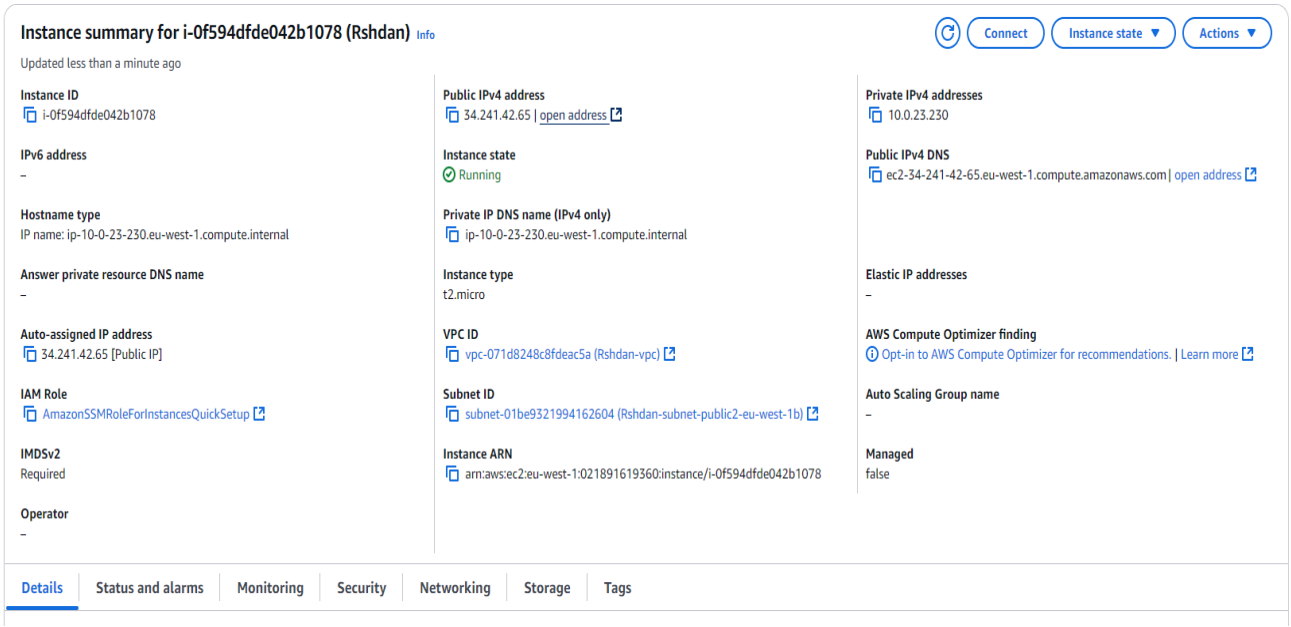
1- make architecture same as screenshot (where two ec2 in public are working as reverse proxy to point to internal load blanacer which forward traffic to two private instances having apache downloaded on it)

1- Create VPC , Subnets and Route Tables



2- Create 2 Public EC2s and Private EC2s





Instance summary for i-07a200595cc9726b9 (Rshdan-private) Info

Connect | Instance state ▼ | Actions ▼

Updated less than a minute ago

Instance ID
i-07a200595cc9726b9

IPv6 address
–

Hostname type
IP name: ip-10-0-131-198.eu-west-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
AmazonSSMRoleForInstancesQuickSetup

IMDSv2
Required

Operator
–

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-131-198.eu-west-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-071d8248c8fdeac5a (Rshdan-vpc)

Subnet ID
subnet-0617d0bafaf4a36d5 (Rshdan-subnet-private1-eu-west-1a)

Instance ARN
arn:aws:ec2:eu-west-1:021891619360:instance/i-07a200595cc9726b9

Private IPv4 addresses
10.0.131.198

Public IPv4 DNS
–

Elastic IP addresses
–

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

Auto Scaling Group name
–

Managed
false

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

Instance summary for i-0224e5602cc9a70c8 (Rshdan-private2) Info

Connect | Instance state ▼ | Actions ▼

Updated less than a minute ago

Instance ID
i-0224e5602cc9a70c8

IPv6 address
–

Hostname type
IP name: ip-10-0-147-18.eu-west-1.compute.internal

Answer private resource DNS name
–

Auto-assigned IP address
–

IAM Role
–

IMDSv2
Required

Operator
–

Public IPv4 address
–

Instance state
Running

Private IP DNS name (IPv4 only)
ip-10-0-147-18.eu-west-1.compute.internal

Instance type
t2.micro

VPC ID
vpc-071d8248c8fdeac5a (Rshdan-vpc)

Subnet ID
subnet-0b8d235d32c0578c9 (Rshdan-subnet-private2-eu-west-1b)

Instance ARN
arn:aws:ec2:eu-west-1:021891619360:instance/i-0224e5602cc9a70c8

Private IPv4 addresses
10.0.147.18

Public IPv4 DNS
–

Elastic IP addresses
–

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

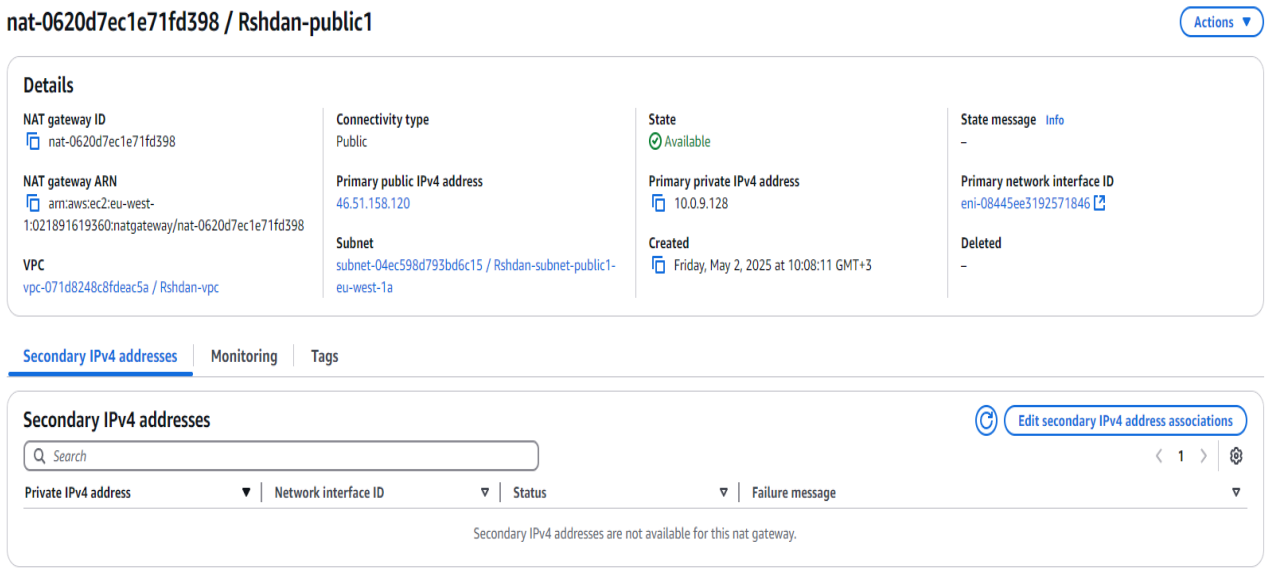
Auto Scaling Group name
–

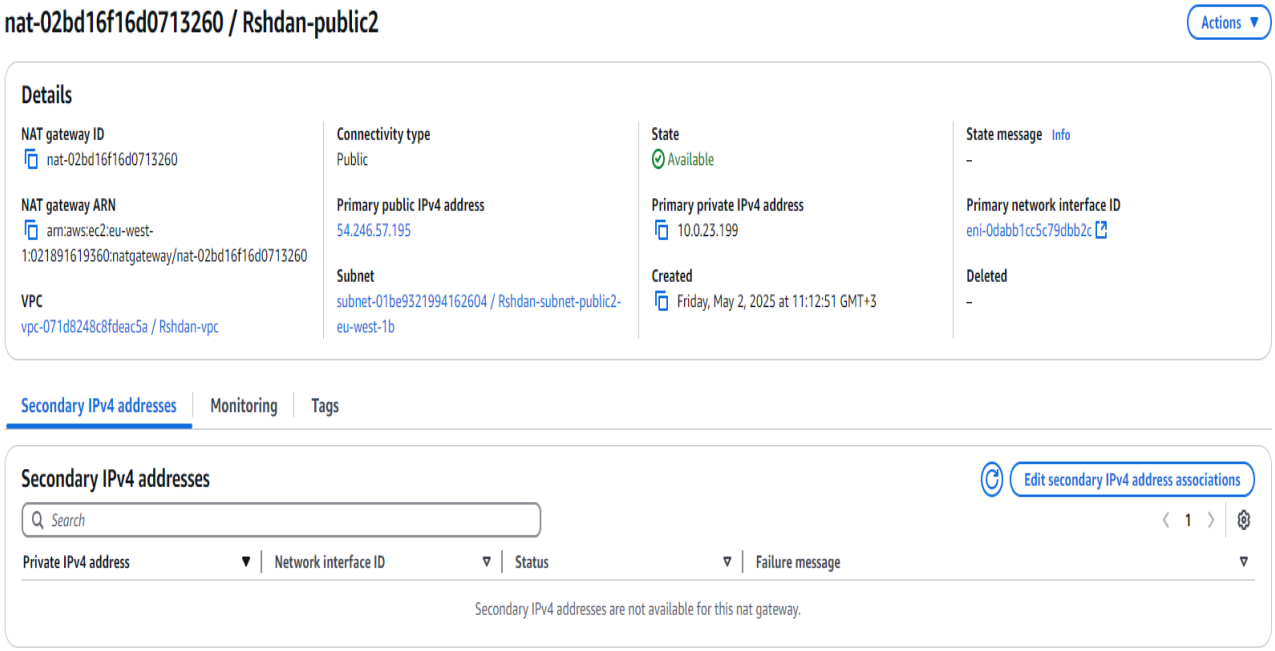
Managed
false

Details | Status and alarms | Monitoring | Security | Networking | Storage | Tags

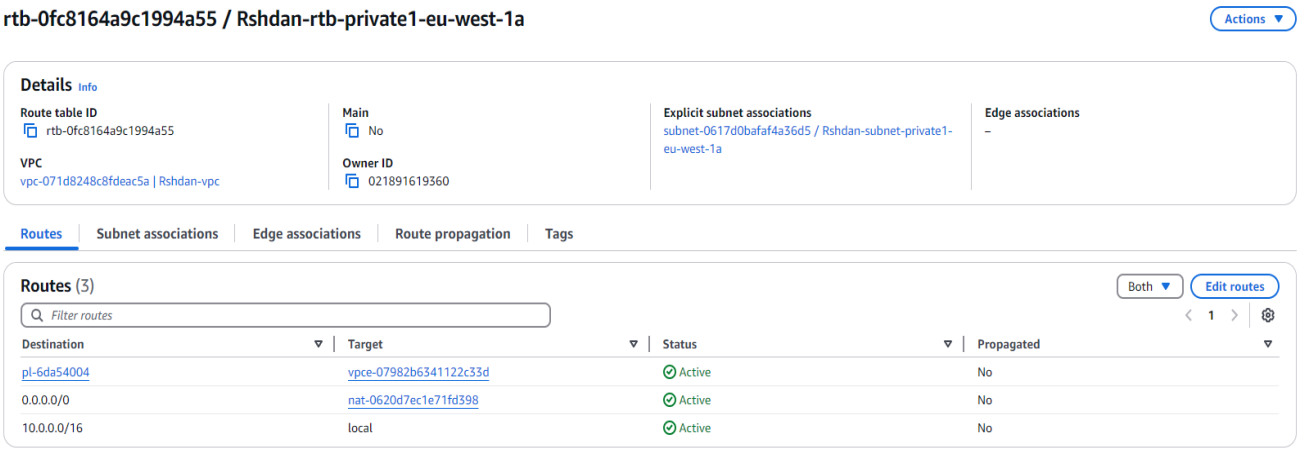
▼ Instance details Info

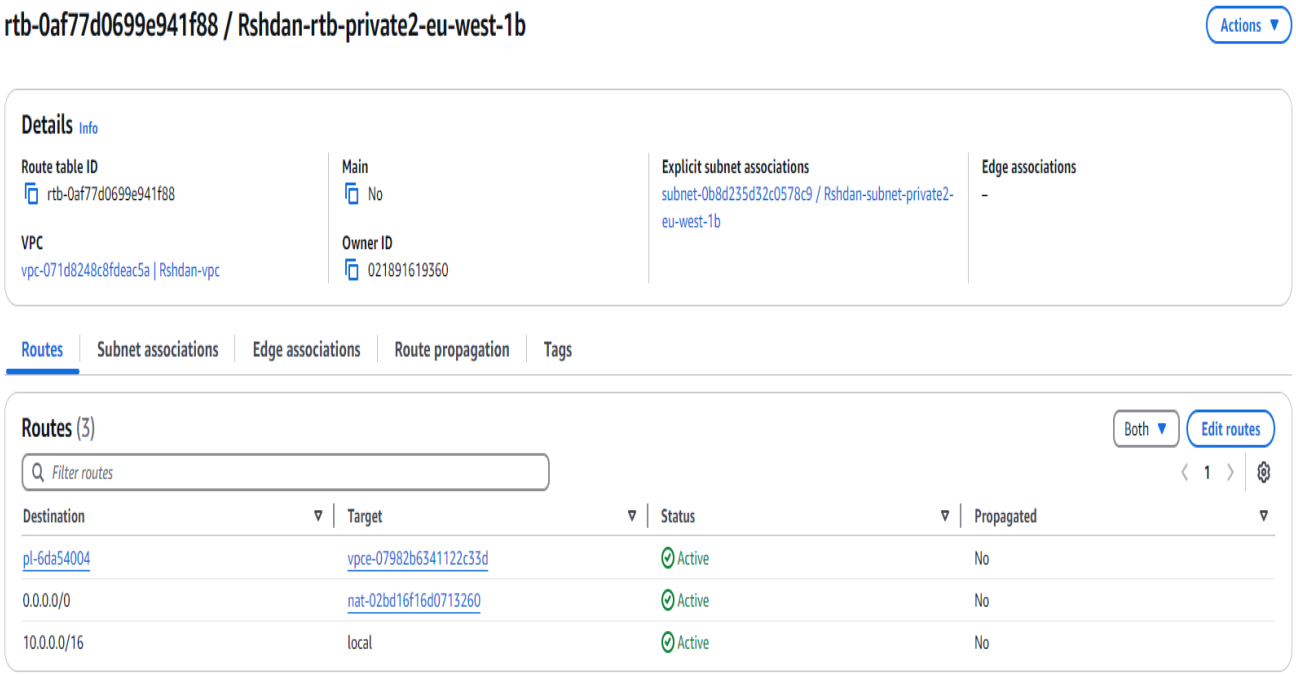
3- Create 2 NAT Gateways and an Elastic IP



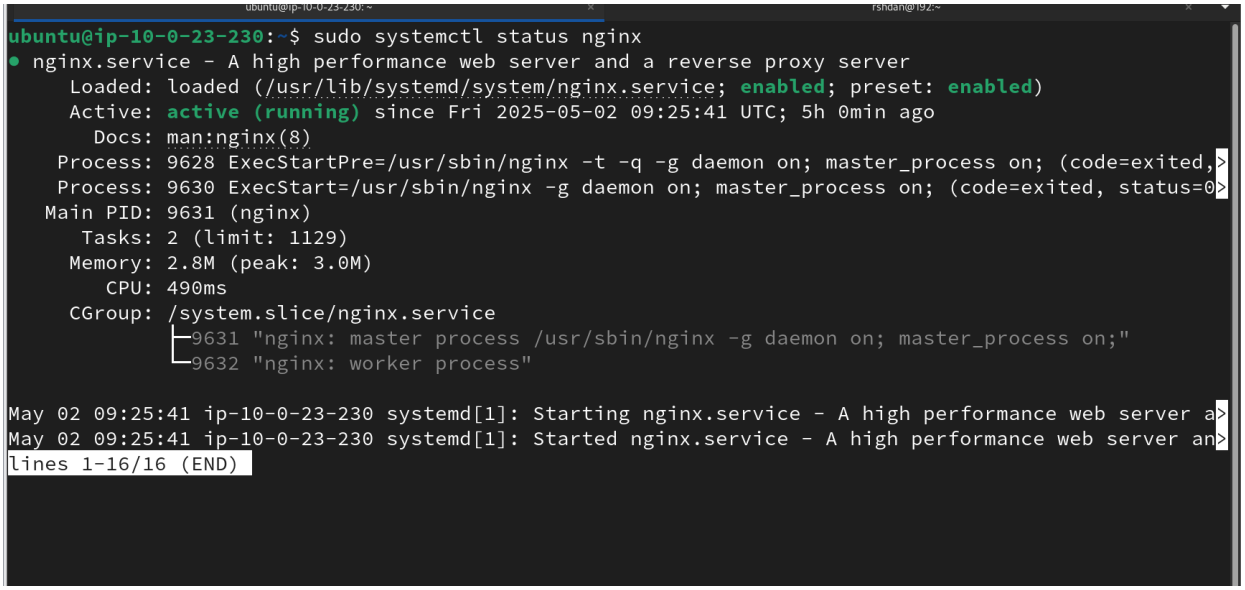


4- Modify Route Tables for Private Subnets

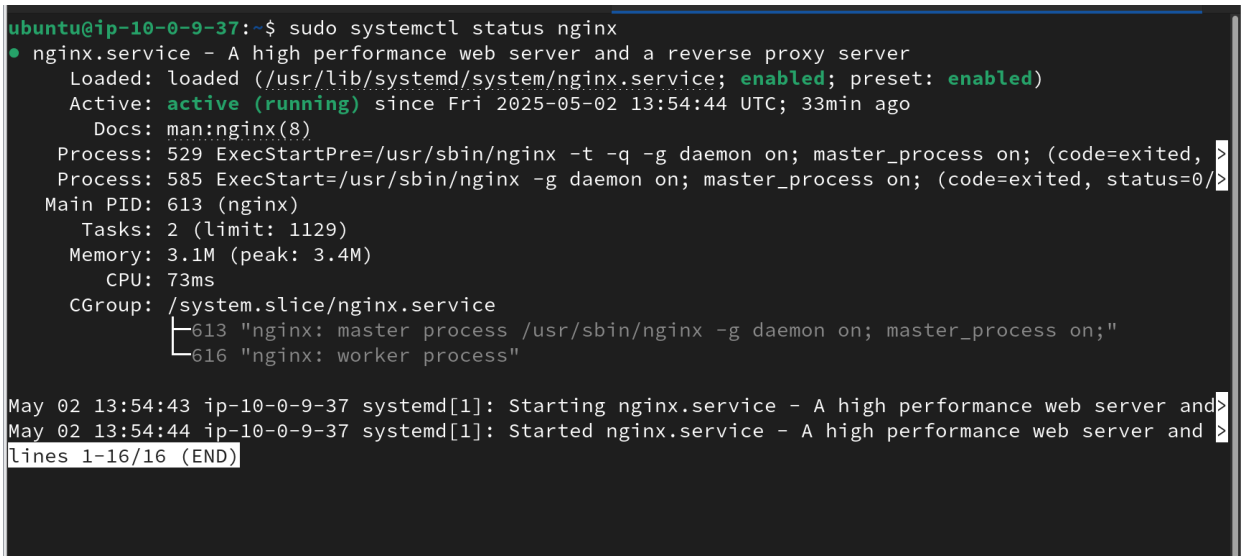




5- Access Public Instances and Install NGINX

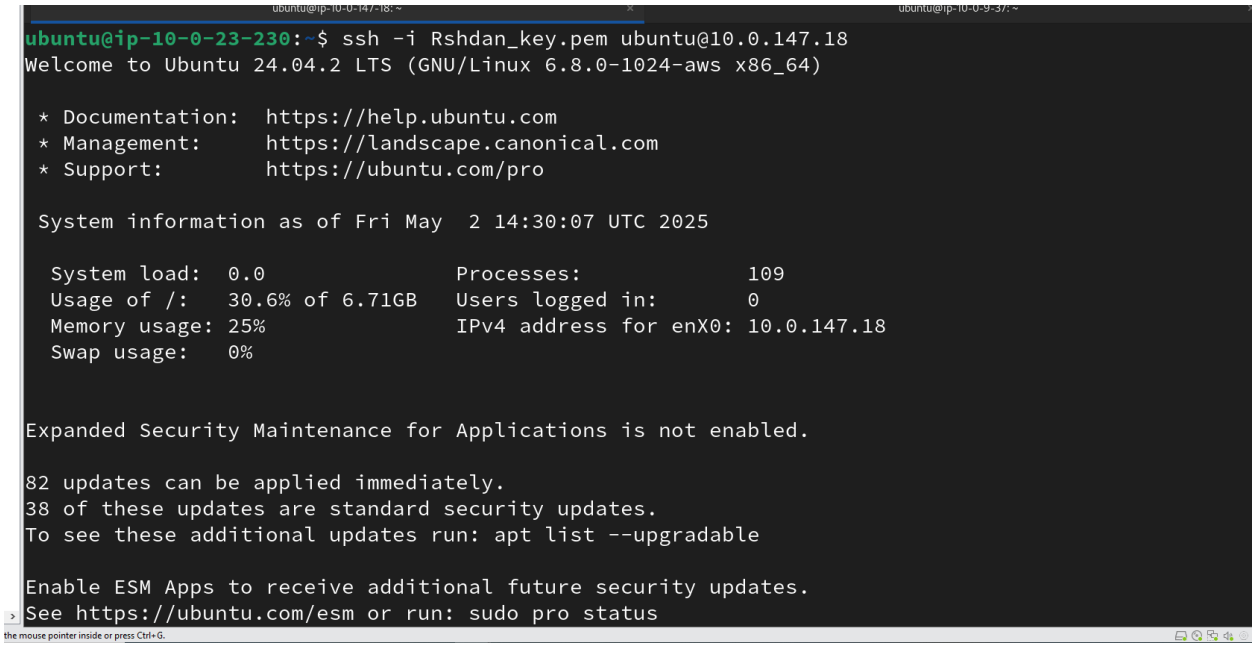


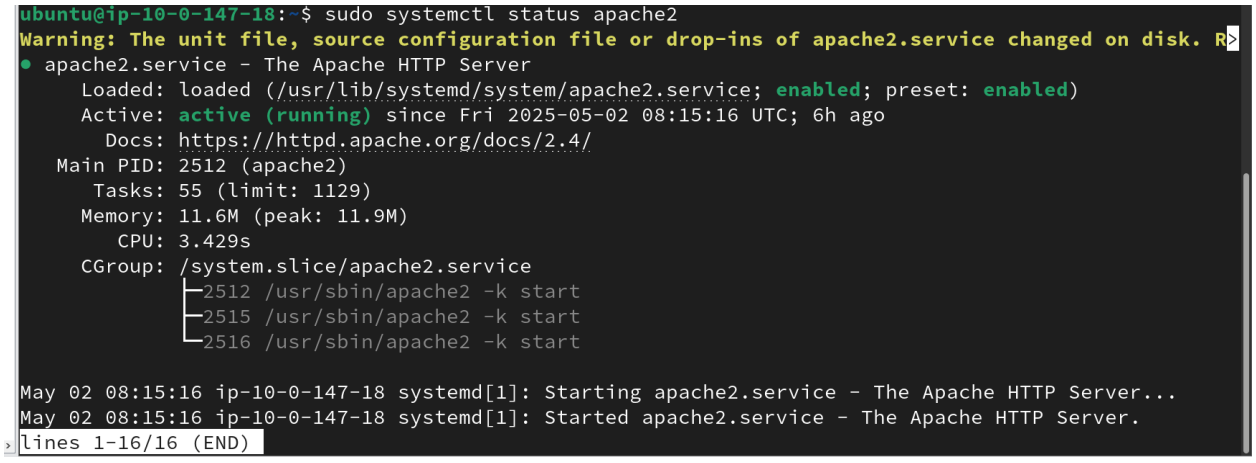
```
ubuntu@ip-10-0-23-230:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-05-02 09:25:41 UTC; 5h 0min ago
       Docs: man:nginx(8)
    Process: 9628 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited,>
    Process: 9630 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0>
   Main PID: 9631 (nginx)
      Tasks: 2 (limit: 1129)
     Memory: 2.8M (peak: 3.0M)
        CPU: 490ms
     CGroup: /system.slice/nginx.service
             ├─9631 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─9632 "nginx: worker process"

May 02 09:25:41 ip-10-0-23-230 systemd[1]: Starting nginx.service - A high performance web server a>
May 02 09:25:41 ip-10-0-23-230 systemd[1]: Started nginx.service - A high performance web server an>
lines 1-16/16 (END)
```



```
ubuntu@ip-10-0-9-37:~$ sudo systemctl status nginx
● nginx.service - A high performance web server and a reverse proxy server
     Loaded: loaded (/usr/lib/systemd/system/nginx.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-05-02 13:54:44 UTC; 33min ago
       Docs: man:nginx(8)
    Process: 529 ExecStartPre=/usr/sbin/nginx -t -q -g daemon on; master_process on; (code=exited, >
    Process: 585 ExecStart=/usr/sbin/nginx -g daemon on; master_process on; (code=exited, status=0/>
   Main PID: 613 (nginx)
      Tasks: 2 (limit: 1129)
     Memory: 3.1M (peak: 3.4M)
        CPU: 73ms
     CGroup: /system.slice/nginx.service
             ├─613 "nginx: master process /usr/sbin/nginx -g daemon on; master_process on;"
             └─616 "nginx: worker process"

May 02 13:54:43 ip-10-0-9-37 systemd[1]: Starting nginx.service - A high performance web server and>
May 02 13:54:44 ip-10-0-9-37 systemd[1]: Started nginx.service - A high performance web server and >
lines 1-16/16 (END)
```

6- Access Public Instances and copy my key to access Private Instances inside Public Instances, then install Apache2



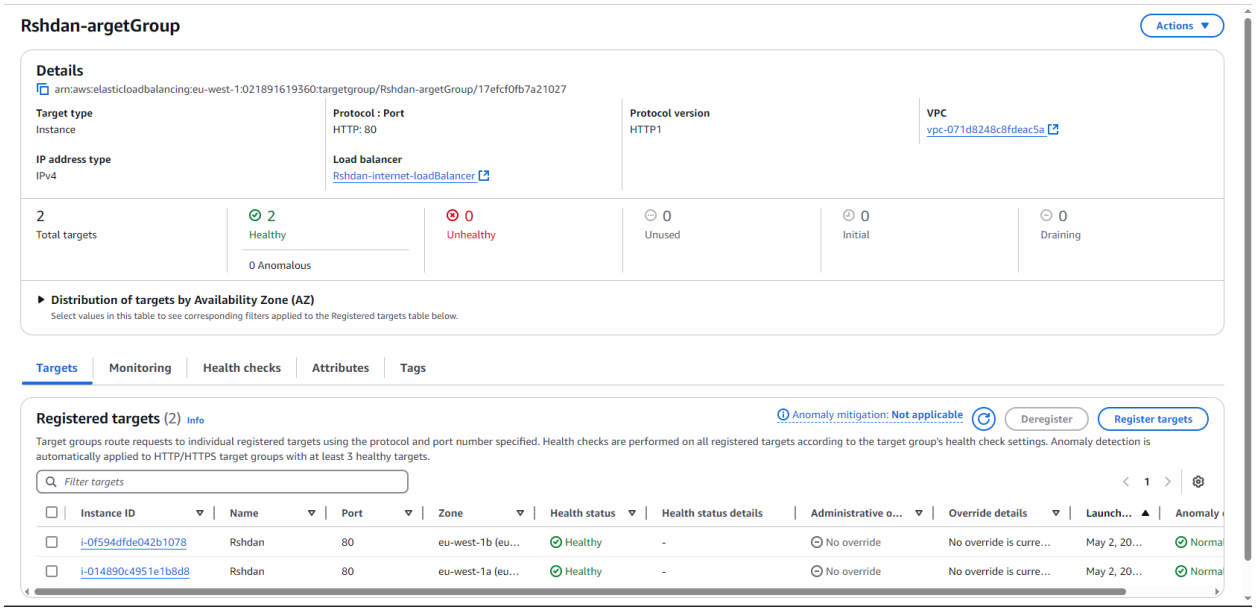


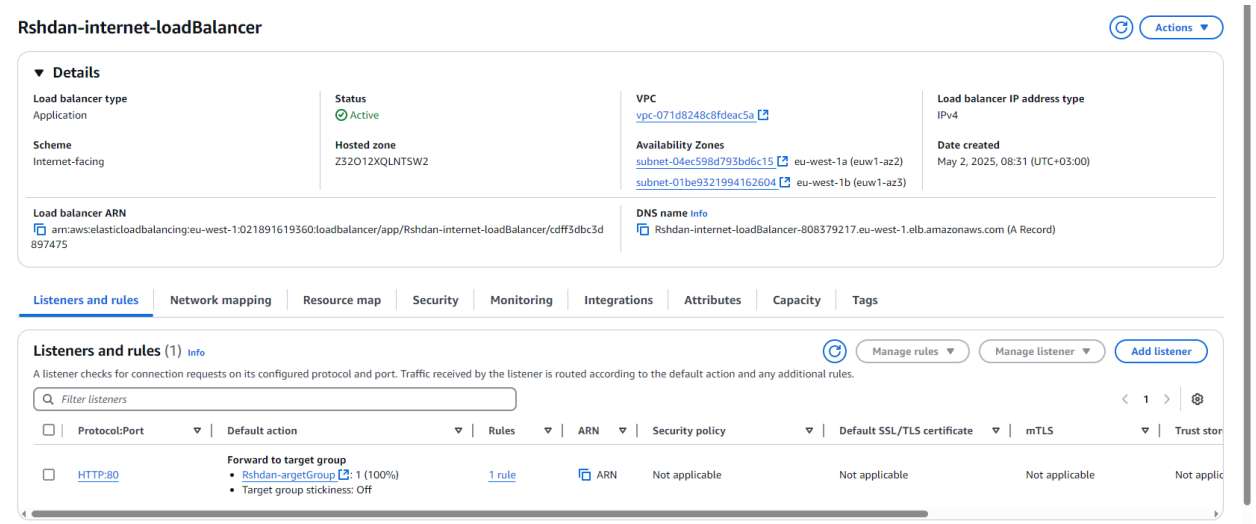
For Second Private Instance

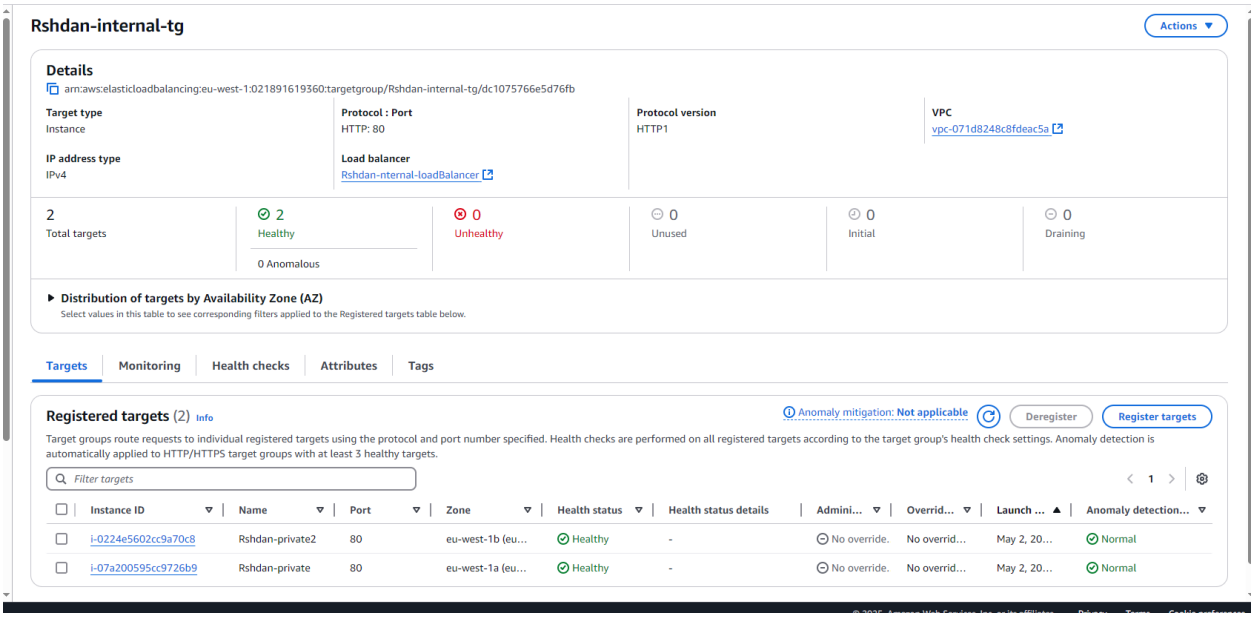
```
ubuntu@ip-10-0-9-37:~$ ssh -i Rshdan_key.pem ubuntu@10.0.131.198
Welcome to Ubuntu 24.04.2 LTS (GNU/Linux 6.8.0-1027-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

 System information as of Fri May  2 14:32:35 UTC 2025

  System load:  0.0               Processes:             107
  Usage of /:   37.7% of 6.71GB   Users logged in:       0
  Memory usage: 24%               IPv4 address for enX0: 10.0.131.198
  Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

   https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.
```

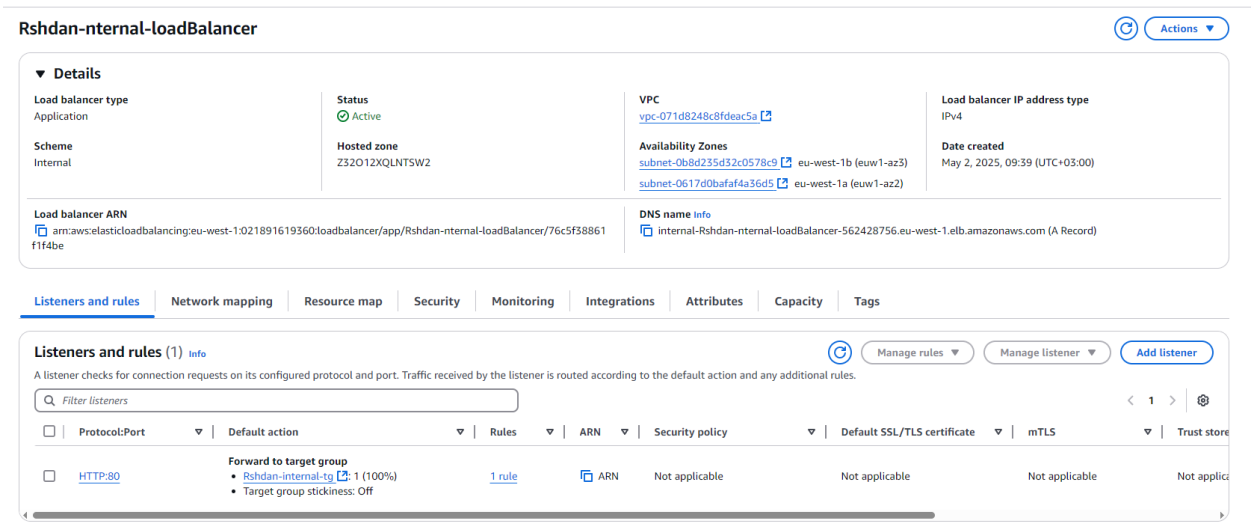
```
ubuntu@ip-10-0-131-198:~$ sudo systemctl status apache2
Warning: The unit file, source configuration file or drop-ins of apache2.service changed on disk. R>
● apache2.service - The Apache HTTP Server
     Loaded: loaded (/usr/lib/systemd/system/apache2.service; enabled; preset: enabled)
     Active: active (running) since Fri 2025-05-02 13:59:23 UTC; 35min ago
       Docs: https://httpd.apache.org/docs/2.4/
   Main PID: 542 (apache2)
      Tasks: 55 (limit: 1129)
     Memory: 13.6M (peak: 13.9M)
        CPU: 409ms
     CGroup: /system.slice/apache2.service
             ├─542 /usr/sbin/apache2 -k start
             ├─544 /usr/sbin/apache2 -k start
             └─545 /usr/sbin/apache2 -k start

May 02 13:59:23 ip-10-0-131-198 systemd[1]: Starting apache2.service - The Apache HTTP Server...
May 02 13:59:23 ip-10-0-131-198 systemd[1]: Started apache2.service - The Apache HTTP Server.
lines 1-16/16 (END)
```

Create external and internal Load Balancer & Create Target Groups for Public Instances and Private Instances






Rshdan-internal-tg
Actions
Details
arn:aws:elasticloadbalancing:eu-west-1:021891619360:targetgroup/Rshdan-internal-tg/dc1075766e5d76fb
Target type
Instance
Protocol : Port
HTTP: 80
Protocol version
HTTP1
VPC
vpc-071d8248c8fdeac5a
IP address type
IPv4
Load balancer
Rshdan-nternal-loadBalancer
2
Total targets
2
Healthy
0 Anomalous
0
Unhealthy
0
Unused
0
Initial
0
Draining
Distribution of targets by Availability Zone (AZ)
Select values in this table to see corresponding filters applied to the Registered targets table below.
Targets
Monitoring
Health checks
Attributes
Tags
Registered targets (2) Info
Anomaly mitigation: Not applicable
Deregister
Register targets
Target groups route requests to individual registered targets using the protocol and port number specified. Health checks are performed on all registered targets according to the target group's health check settings. Anomaly detection is automatically applied to HTTP/HTTPS target groups with at least 3 healthy targets.
Filter targets
Instance ID
Name
Port
Zone
Health status
Health status details
Admini...
Overrid...
Launch ...
Anomaly detection...
i-0224e5602cc9a70c8
Rshdan-private2
80
eu-west-1b (eu...
Healthy
-
No override.
No overrid...
May 2, 20...
Normal
i-07a200595cc9726b9
Rshdan-private
80
eu-west-1a (eu...
Healthy
-
No override.
No overrid...
May 2, 20...
Normal

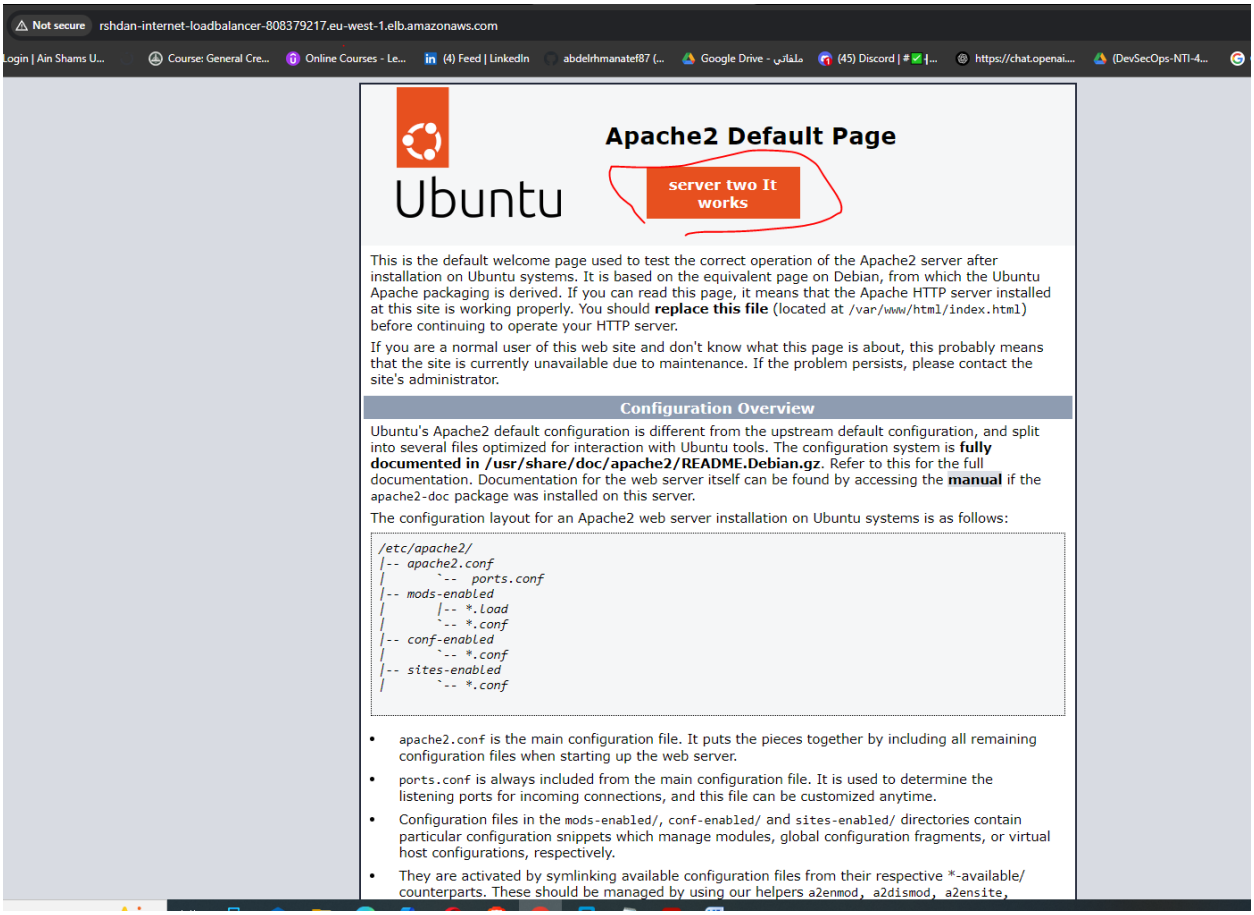

Rshdan-nternal-loadBalancer
Actions
Details
Load balancer type
Application
Status
Active
VPC
vpc-071d8248c8fdeac5a
Load balancer IP address type
IPv4
Scheme
Internal
Hosted zone
Z32O12XQLNTSW2
Availability Zones
subnet-0b8d235d32c0578c9 eu-west-1b (euw1-az3)
subnet-0617d0bafaf4a36d5 eu-west-1a (euw1-az2)
Date created
May 2, 2025, 09:39 (UTC+03:00)
Load balancer ARN
arn:aws:elasticloadbalancing:eu-west-1:021891619360:loadbalancer/app/Rshdan-nternal-loadBalancer/76c5f38861f1f4be
DNS name Info
internal-Rshdan-nternal-loadBalancer-562428756.eu-west-1.elb.amazonaws.com (A Record)
Listeners and rules
Network mapping
Resource map
Security
Monitoring
Integrations
Attributes
Capacity
Tags
Listeners and rules (1) Info
Manage rules
Manage listener
Add listener
A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.
Filter listeners
Protocol:Port
Default action
Rules
ARN
Security policy
Default SSL/TLS certificate
mTLS
Trust store
HTTP:80
Forward to target group
Rshdan-internal-tg: 1 (100%)
Target group stickiness: Off
1 rule
ARN
Not applicable
Not applicable
Not applicable
Not applica

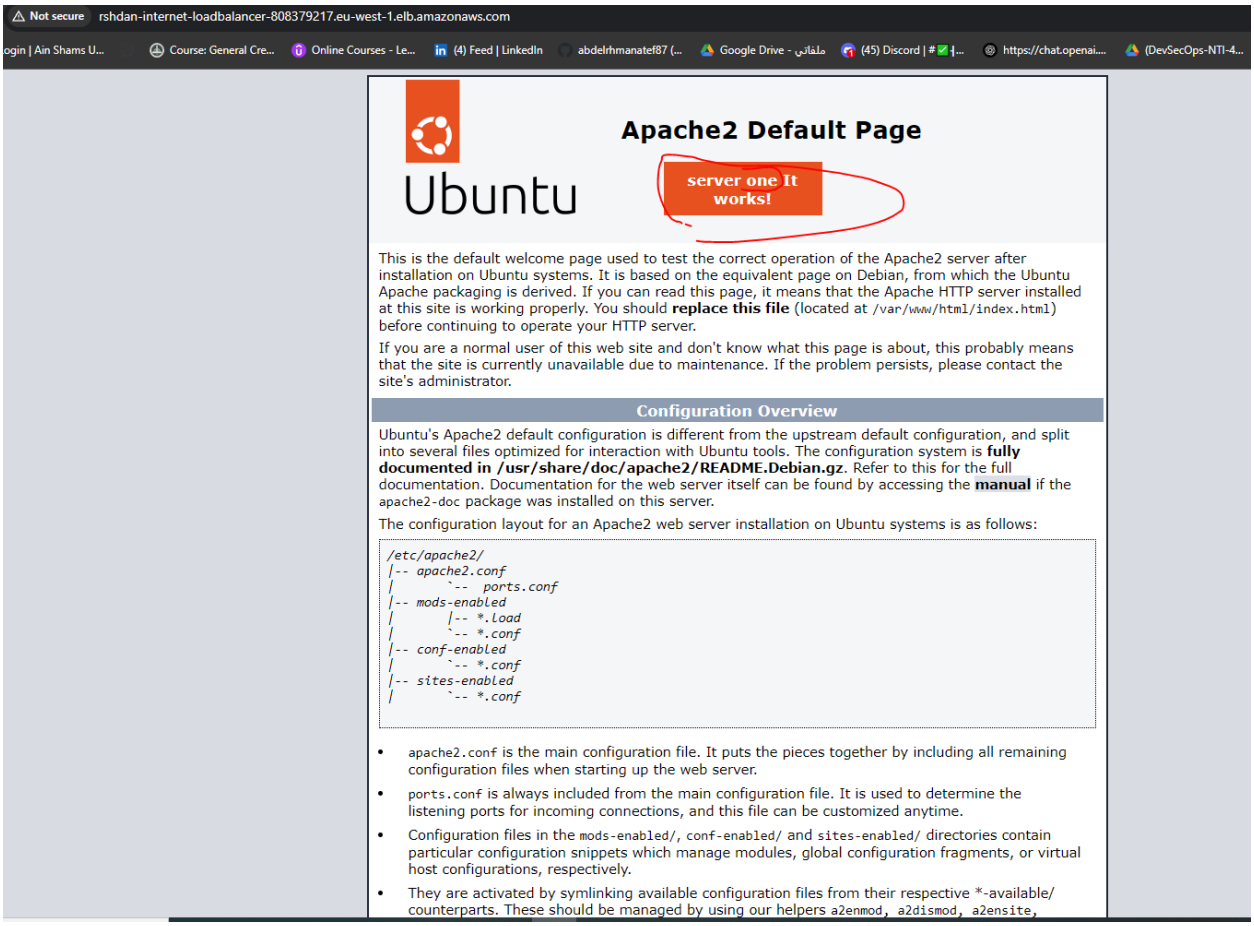
now I can access DNS for load balancer to show nginx page
but that I do not want. I want nginx as reverse proxy so we modify nginx configurations in Two Public Instances

Modify nginx configurations in Two Public Instances

My NGINX Configurations

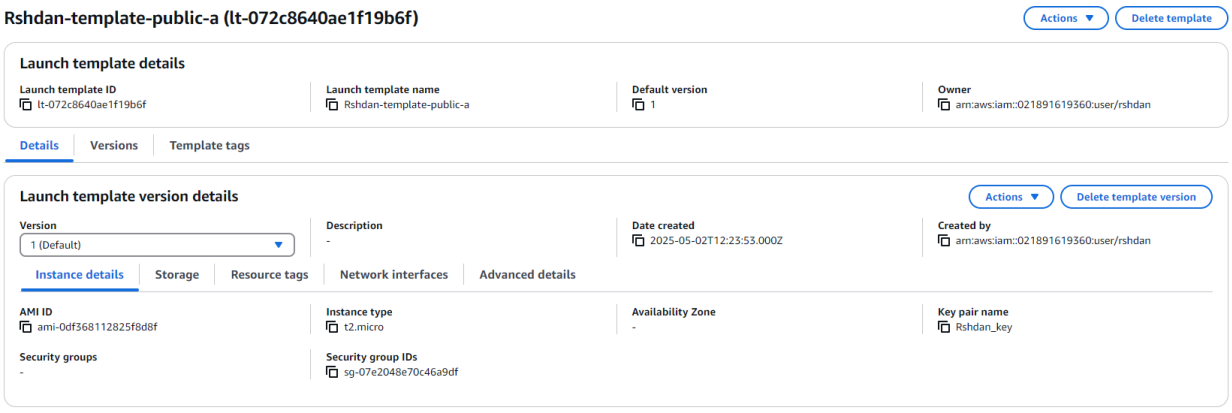
```
server {

   listen 80;

   server_name _;



   location / {

      proxy_pass http://internal-Rshdan-nternal-loadBalancer-562428756.eu-west-
1.elb.amazonaws.com;

      proxy_set_header Host $host;

      proxy_set_header X-Real-IP $remote_addr;

      proxy_set_header X-Forwarded-For $proxy_add_x_forwarded_for;

   }

}

~

~
```


Not secure
rshdan-internet-loadbalancer-808379217.eu-west-1.elb.amazonaws.com
Ubuntu
Apache2 Default Page
server two It works
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite,

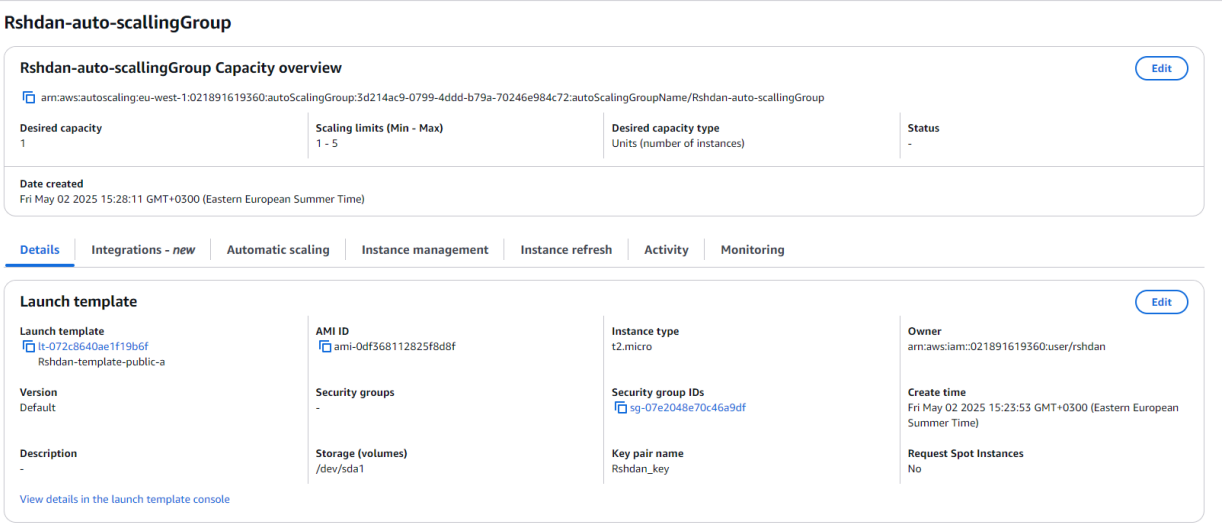

Not secure
rshdan-internet-loadbalancer-808379217.eu-west-1.elb.amazonaws.com
Apache2 Default Page
Ubuntu
server one It works!
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace this file (located at /var/www/html/index.html) before continuing to operate your HTTP server.
If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.
Configuration Overview
Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/apache2/README.Debian.gz. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.
The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:
/etc/apache2/
|-- apache2.conf
| `-- ports.conf
|-- mods-enabled
| |-- *.load
| `-- *.conf
|-- conf-enabled
| `-- *.conf
|-- sites-enabled
| `-- *.conf
apache2.conf is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
ports.conf is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
Configuration files in the mods-enabled/, conf-enabled/ and sites-enabled/ directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
They are activated by symlinking available configuration files from their respective *-available/ counterparts. These should be managed by using our helpers a2enmod, a2dismod, a2ensite,

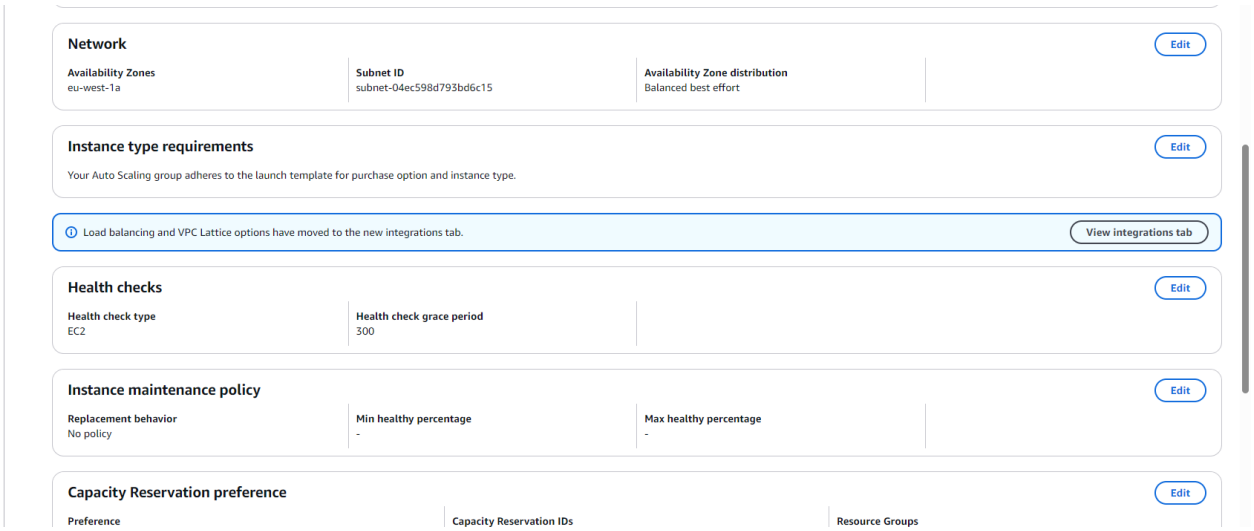
2- make autoscaling group and scale out based on any policy and prove it's working

1- Create Launch Template

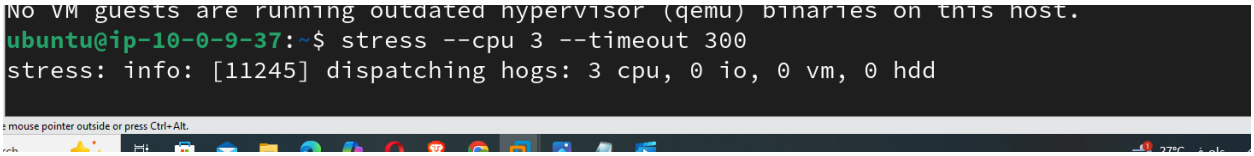


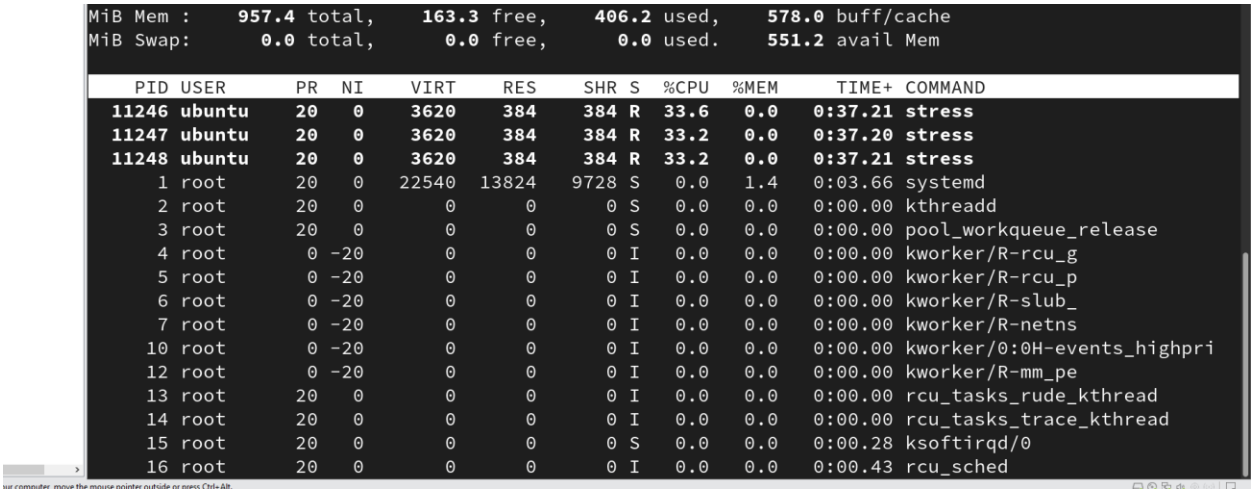
2- Create Auto Scaling Group





Test Auto Scaling in AWS Console

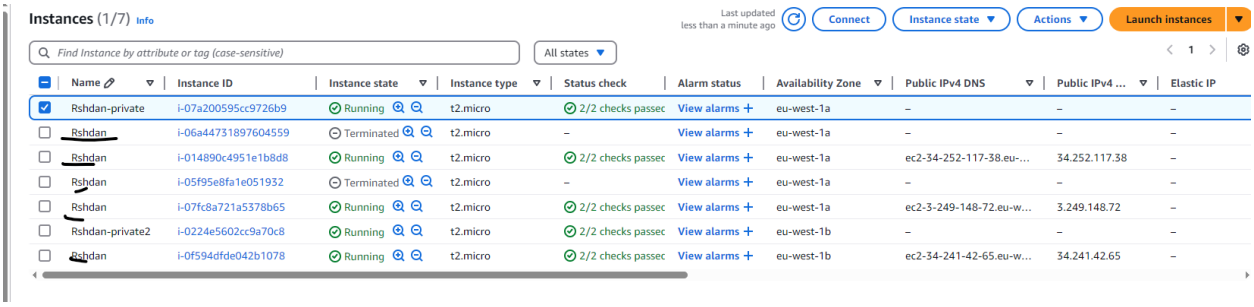




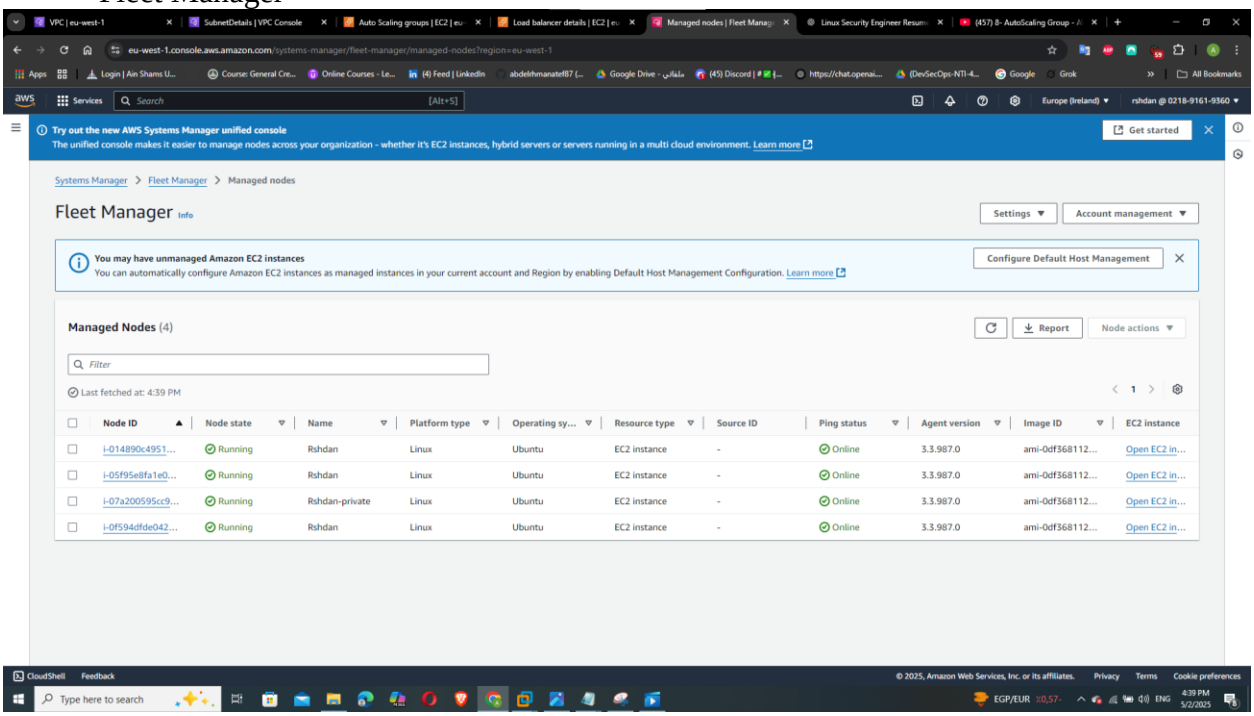
Activity history (2)

Filter activity history

Status	Description	Cause	Start time
Successful	Launching a new EC2 instance: i-01d232f48a0c3ca77	At 2025-05-02T12:28:11Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-05-02T12:28:13Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 May 02, 03:2 +03:00
Successful	Launching a new EC2 instance: i-05f95e8fa1e051932	At 2025-05-02T12:28:11Z a user request created an AutoScalingGroup changing the desired capacity from 0 to 2. At 2025-05-02T12:28:13Z an instance was started in response to a difference between desired and actual capacity, increasing the capacity from 0 to 2.	2025 May 02, 03:2 +03:00



- Fleet Manager



3- Run Patch Manager