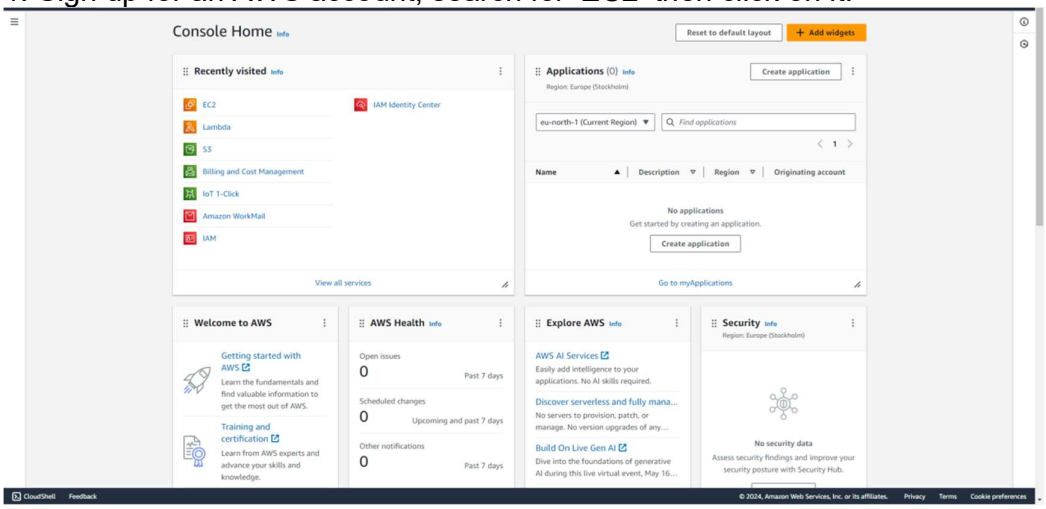
PROBLEM STATEMENT :

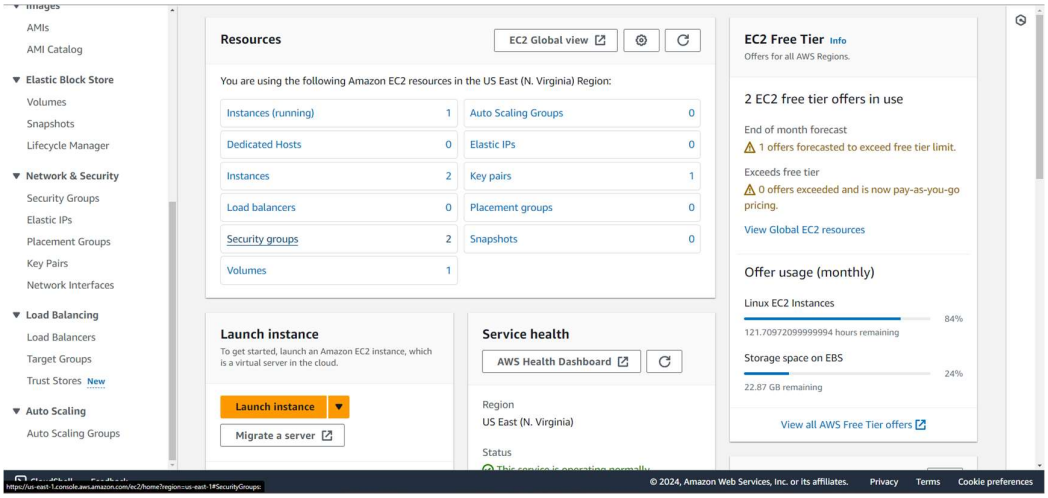
11) Build scaling plans in AWS that balance the load on different EC2 instances.

Steps to build scaling plans :

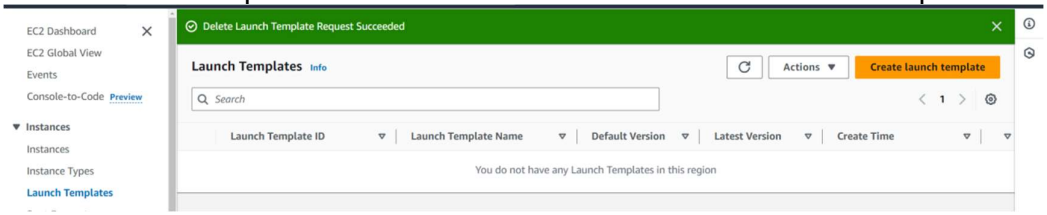
1. Sign up for an AWS account, search for ‘EC2’ then click on it.



2. If you have an existing Security group then no need to create it if not then create it.



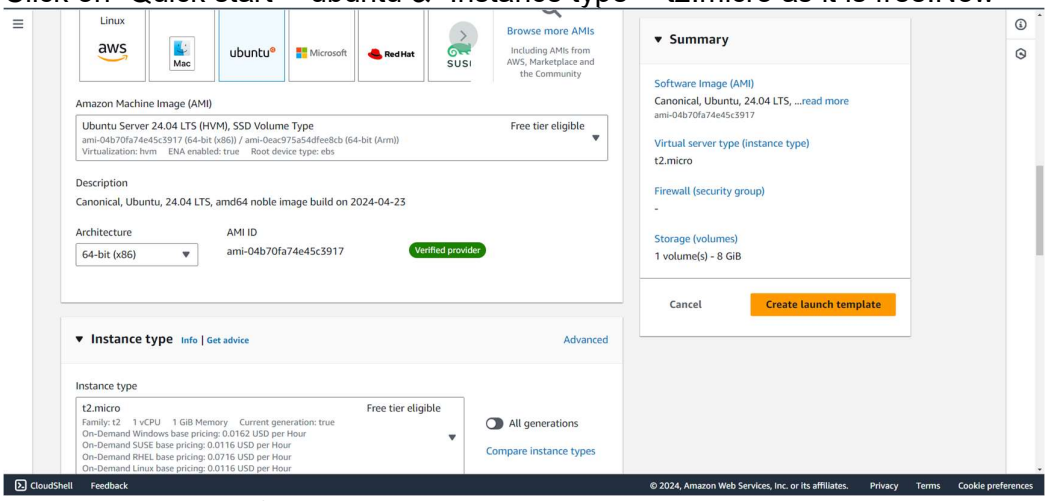
3. Create one template. Click Instances & there click on “Launch template”.



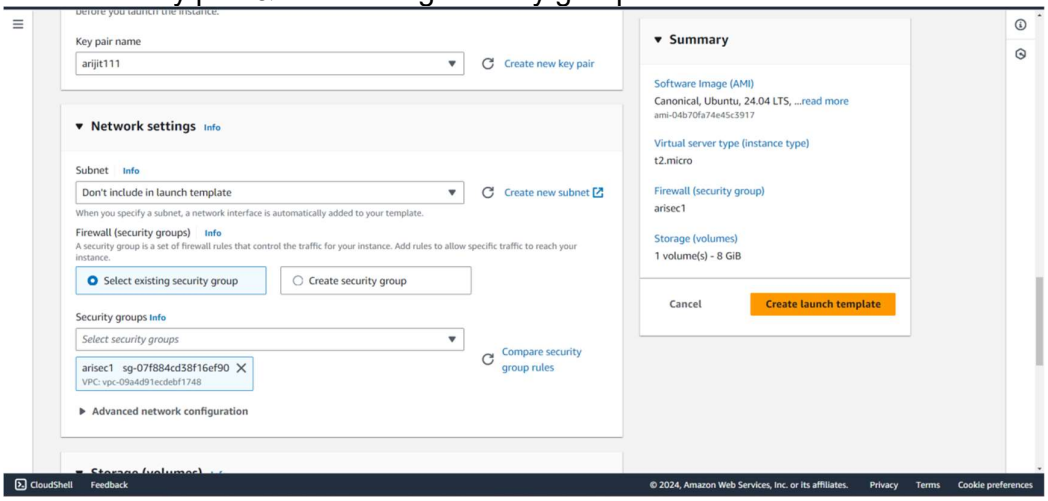
4. Now click on “Create launch template”.

5. Under “Create launch template”, give the following details.

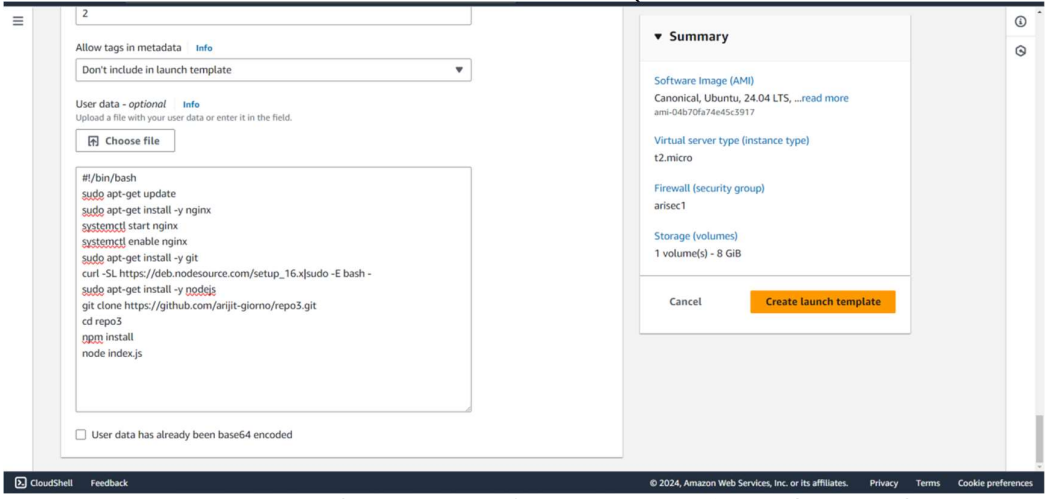
Click on “Quick start”->ubuntu & “Instance type”->t2.micro as it is free.Now



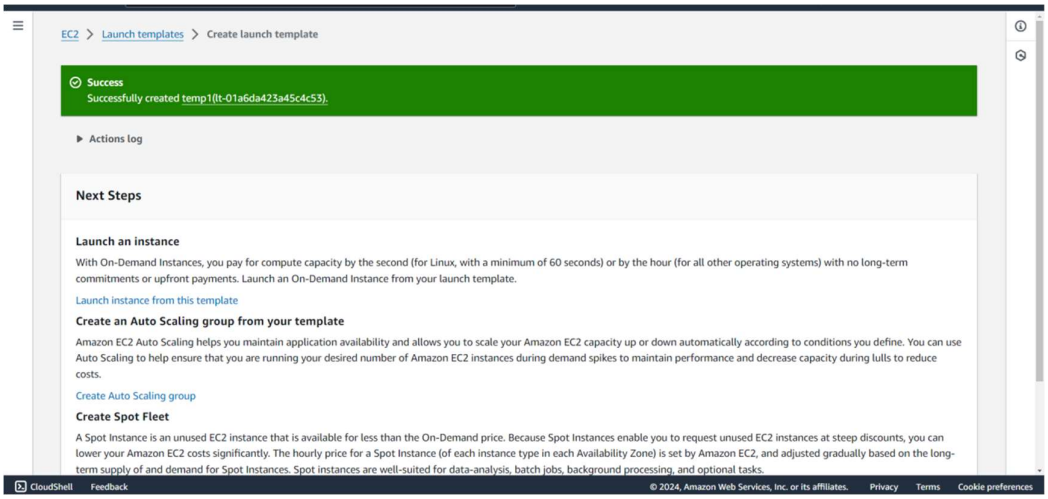
select the key pair & the existing security group.



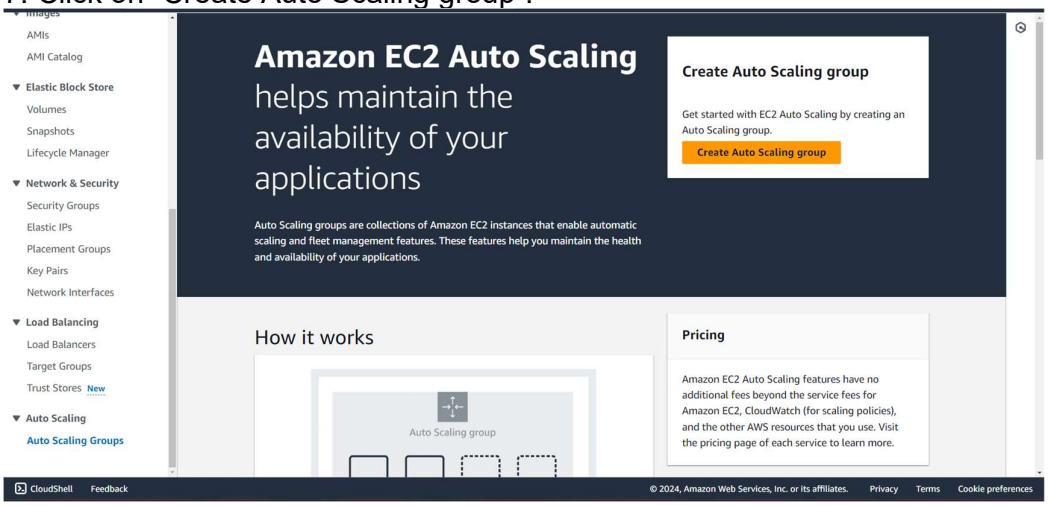
Expand the “Advanced Details” & scroll down to the bottom, in the bash console type the following commands, give the address & repository name from GitHub. Then click on “Create Launch template”



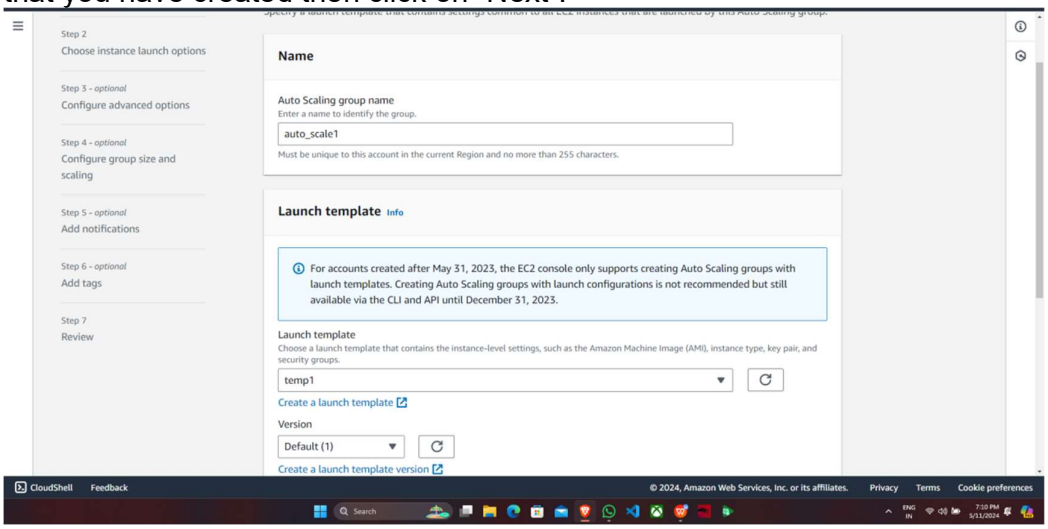
6. Template has successfully created & now click on “Auto Scaling Groups”.



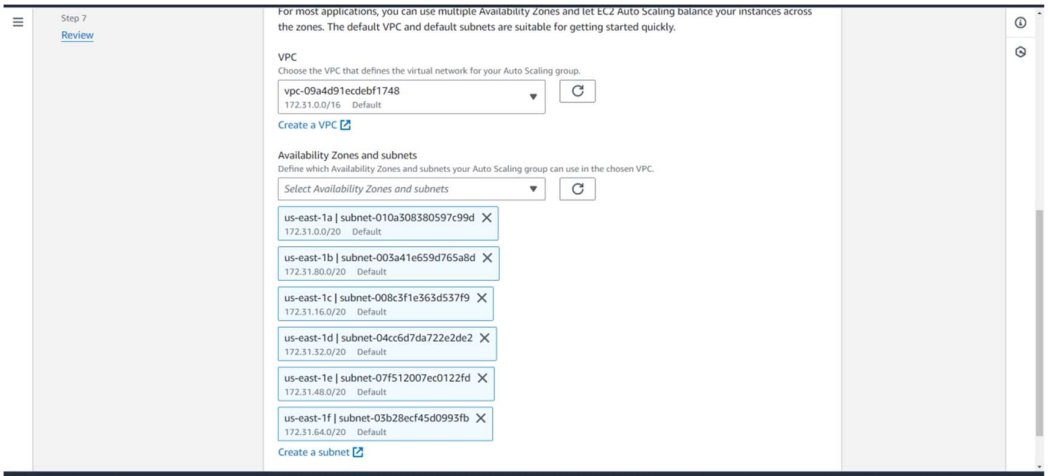
7. Click on “Create Auto Scaling group”.

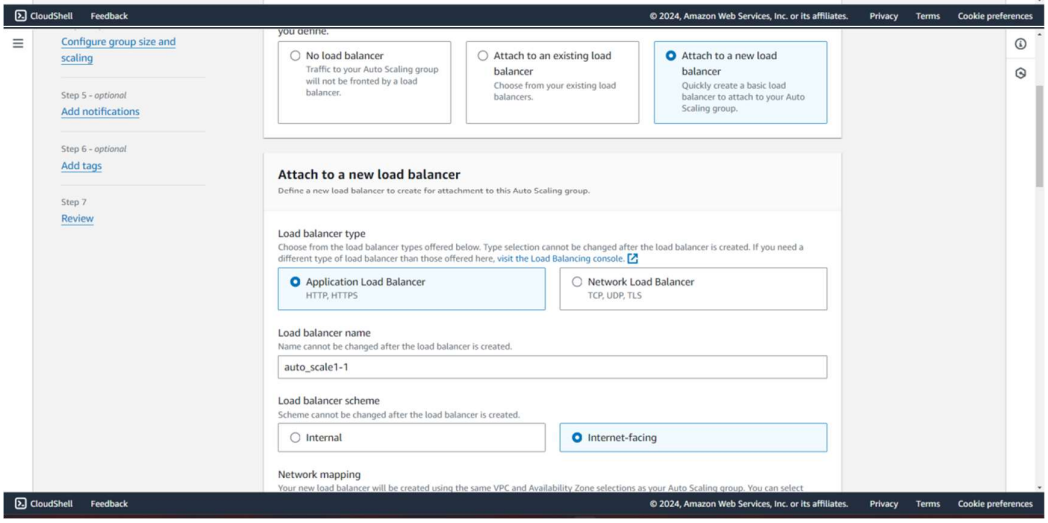


8. Under “Create Auto Scaling group”, give the name & choose the template that you have created then click on “Next”.

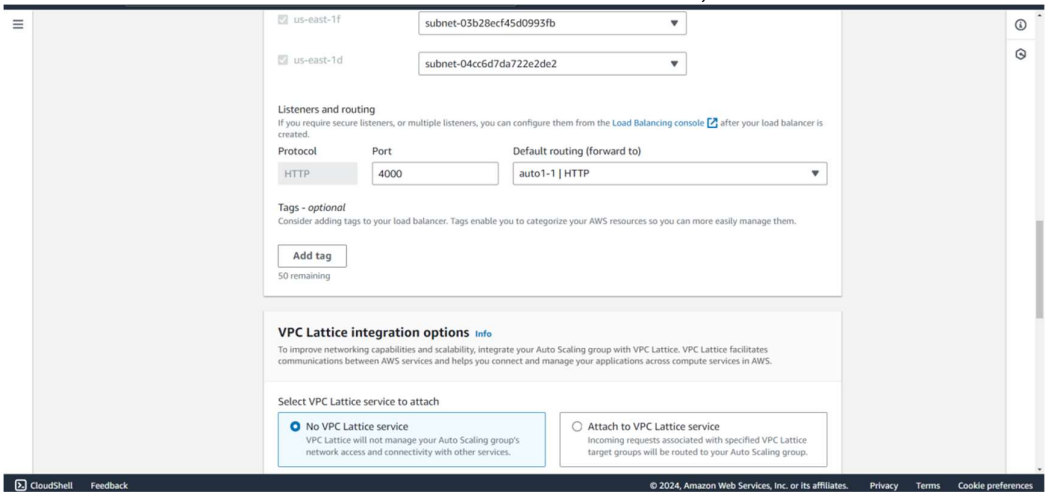


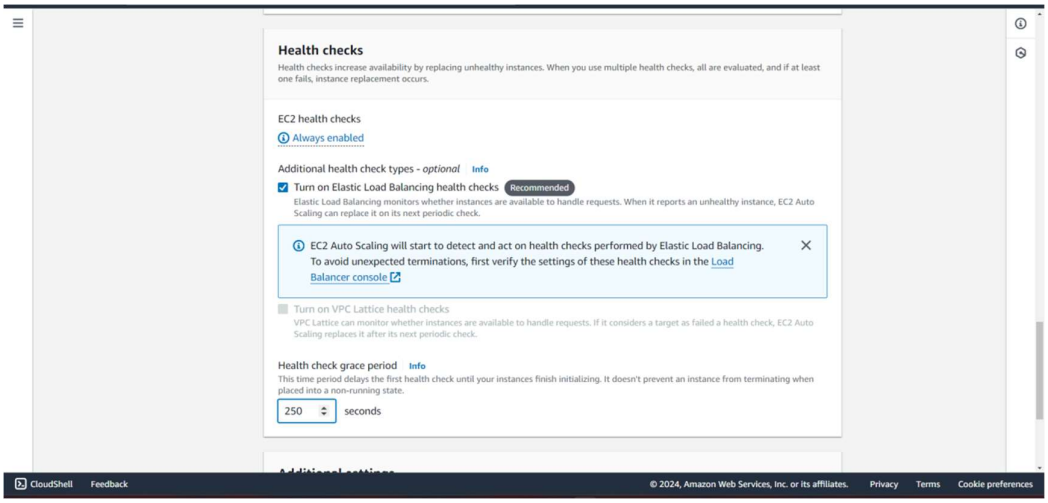
Then select all the “Availability Zones and subnets” & click on “Next”.



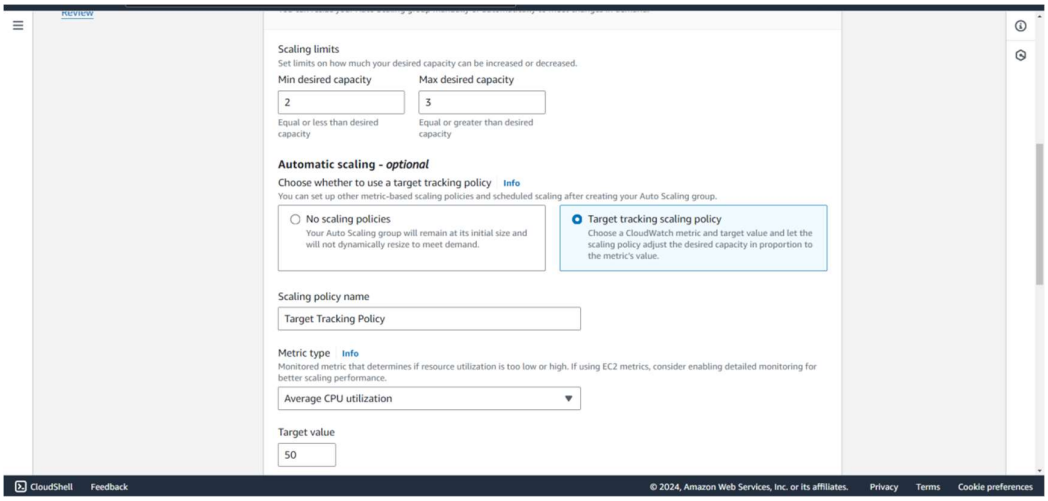


Click on “No VPC Lattice service”.In “Health checks”, click on the checkbox

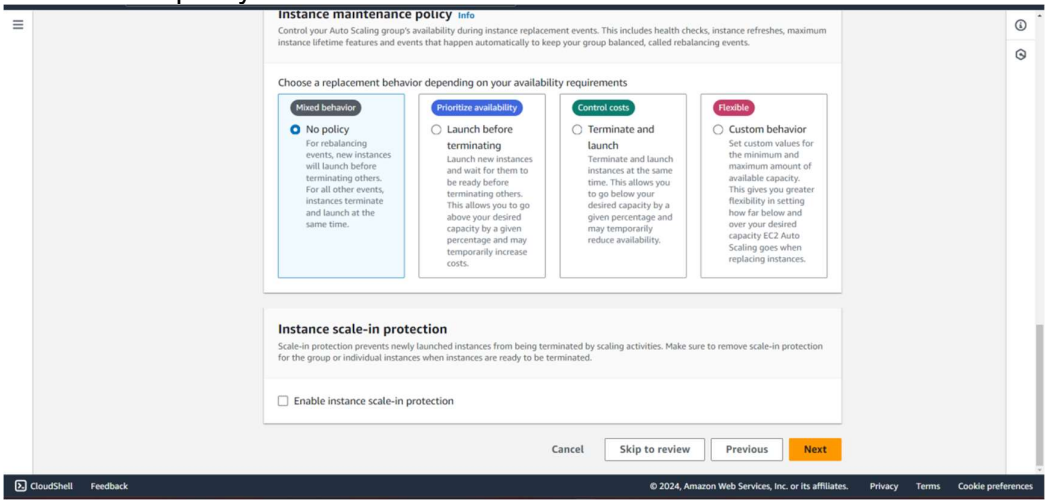


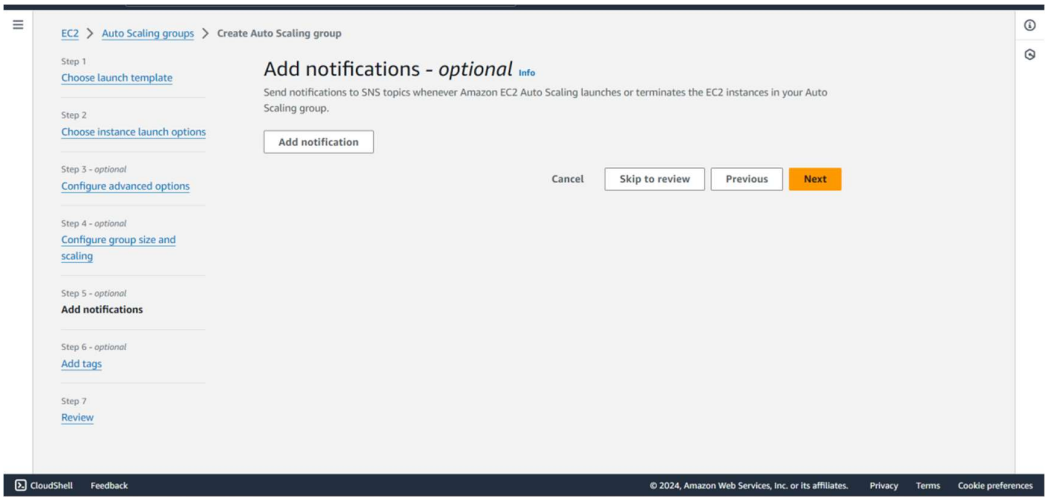


“Turn on Elastic load …” Then click on “Next”.

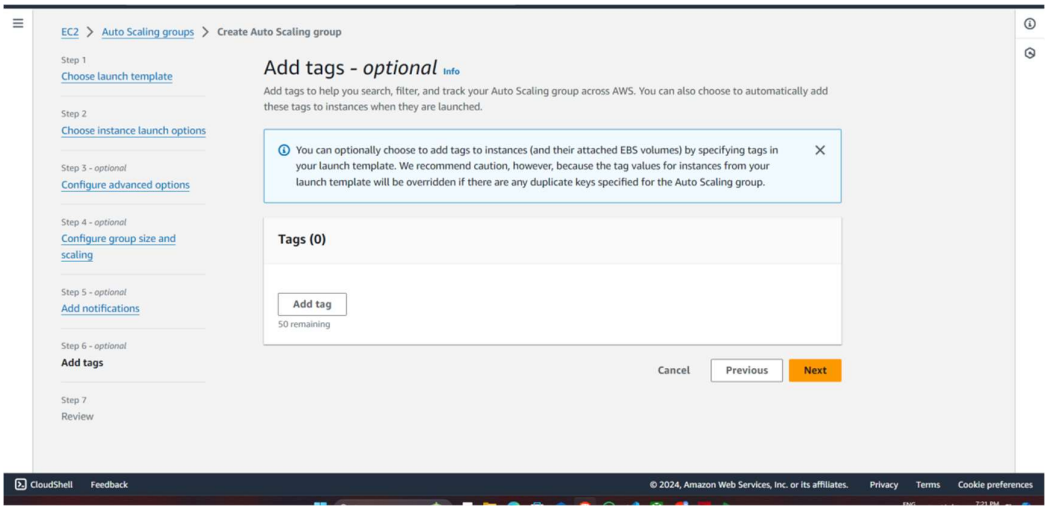


Click on “No policy” then on “Next”.Click on “Next”.

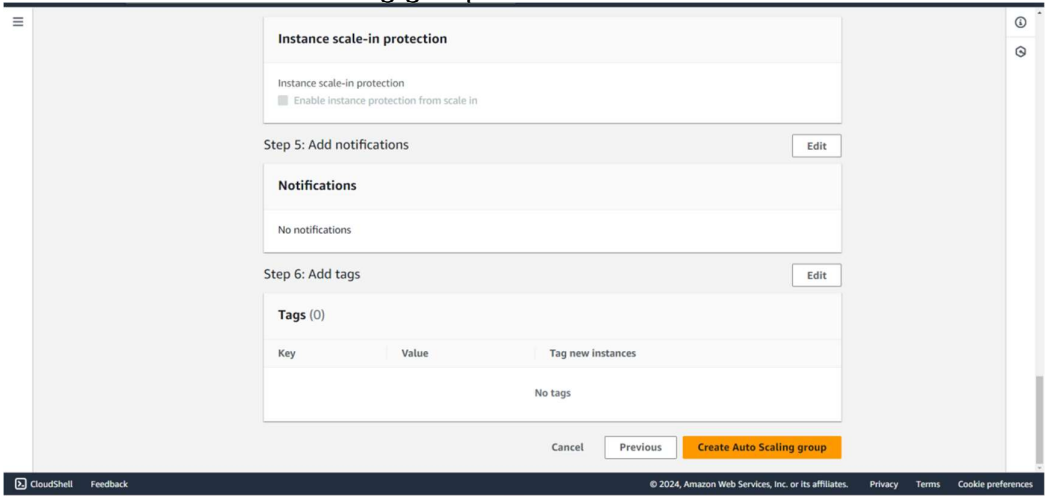




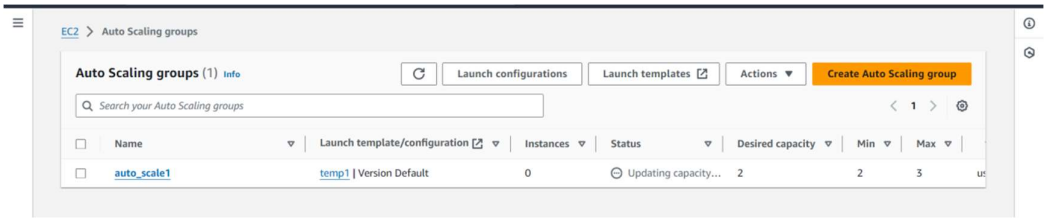
Click on “Next”.



Click on “Create Auto Scaling group”.



9. Auto scaling group is successfully created.

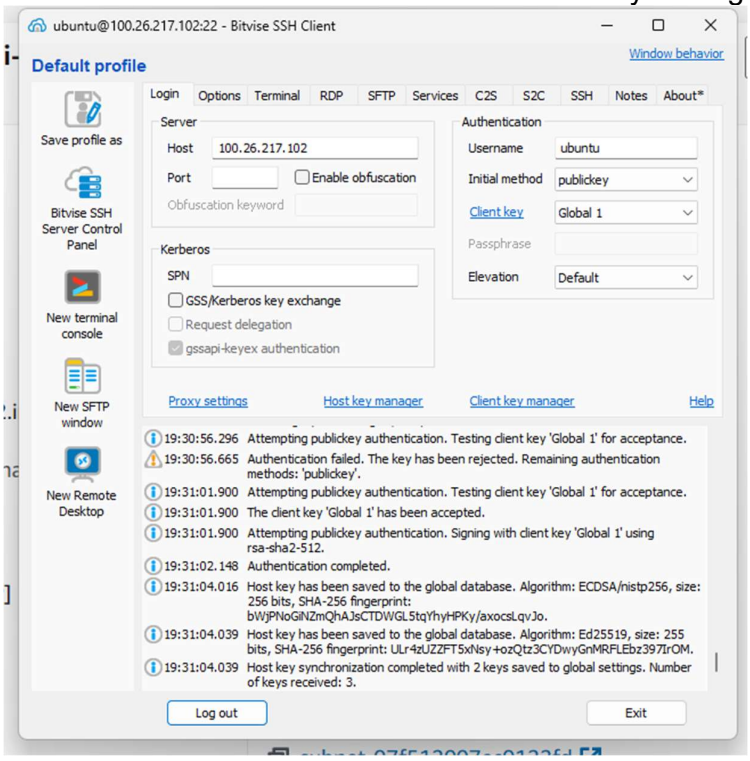


10. Now go to “Instance” and check for running instances with no name but then click on
any one of the instance ID & copy the “Public IPv4 address”.

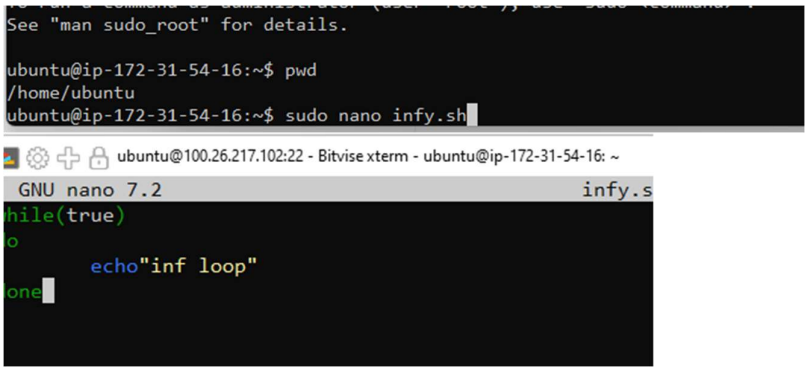
11. Paste the address in a new Window.

12. Now add “:4000” at the end of the IPv4 address and press enter.

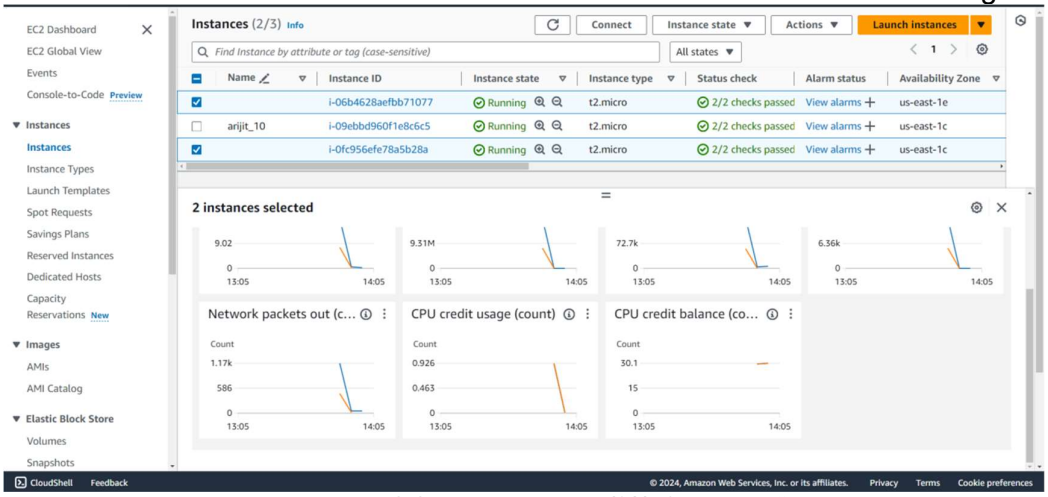
13. Then copy the address of any one of the instance & paste in the host of the “Bitvise SSH Client” then click on “Client key Manager”.



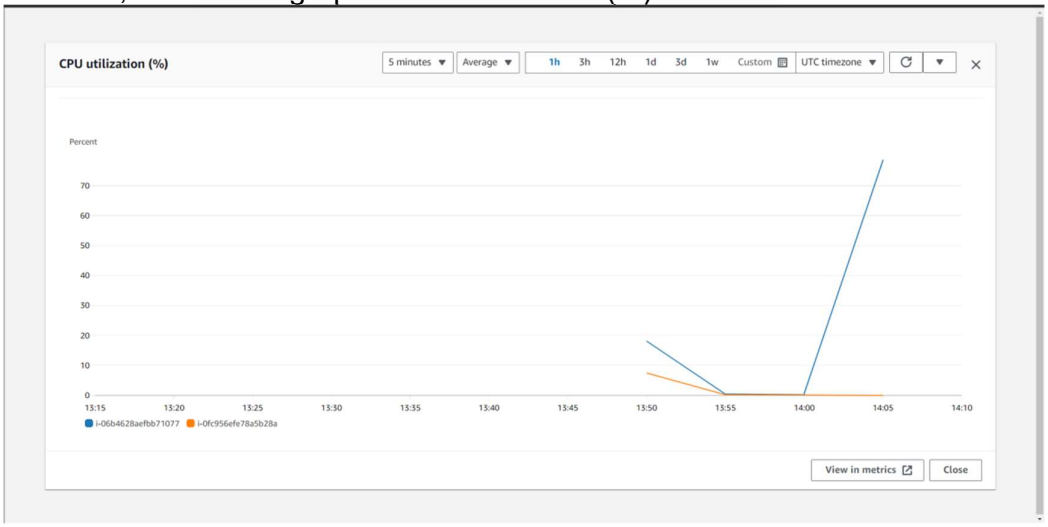
14. Then import the key & click on “Log in”.
15. The “Log out” came means that is is successfully logged in.
16.Now in “New terminal console and type the following commands.



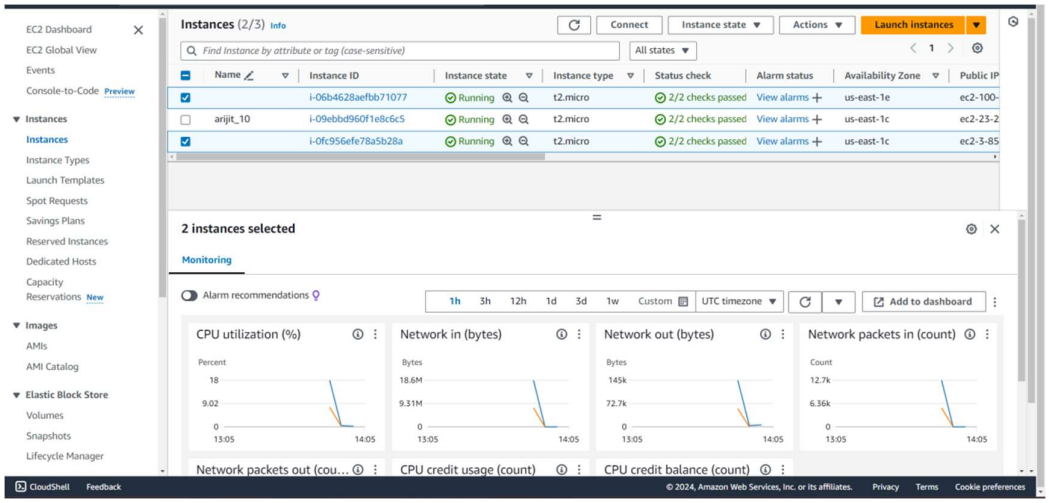
17. Now In Instances select both the unnamed instances & click on ‘Enlarge’.



18. Here, we see the graph of CPU utilization(%) & select “Local timezone”.



19. Here, we see that two instances are running along with the initialization of the third instance.


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
Lifecycle Manager
Instances (2/3) Info
Connect
Instance state
Actions
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states
Name
Instance ID
Instance state
Instance type
Status check
Alarm status
Availability Zone
Public IP
i-06b4628aefbb71077 Running t2.micro 2/2 checks passed View alarms us-east-1e ec2-100-
arijit_10 i-09ebbd960f1e8c6c5 Running t2.micro 2/2 checks passed View alarms us-east-1c ec2-23-2
i-0fc956efe78a5b28a Running t2.micro 2/2 checks passed View alarms us-east-1c ec2-3-85
2 instances selected
Monitoring
Alarm recommendations
1h 3h 12h 1d 3d 1w Custom UTC timezone
Add to dashboard
CPU utilization (%)
Percent
18
9.02
0
13:05
14:05
Network in (bytes)
Bytes
18.6M
9.31M
0
13:05
14:05
Network out (bytes)
Bytes
145k
72.7k
0
13:05
14:05
Network packets in (count)
Count
12.7k
6.36k
0
13:05
14:05
Network packets out (cou...
CPU credit usage (count)
CPU credit balance (count)
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences