Scaling the EC2 Instance Based on Monitored CloudWatch Metrics

Use Auto Scaling to manage the EC2 instances and use EC2 instance and capture the metrics in the CloudWatch.

Description:

Let's take the case of Hotstar — a platform that provides on-demand video streaming services. The more the users join the streaming service platform, the more the resources in terms of servers (EC2 in AWS) Hotstar needs to invest in. This way, the load is distributed across different servers and leads to jitter-free experience for the customers while watching the videos. Another example is Amazon Prime Day, where a bevy of customers access the amazon.com site. Depending on the number of customers logging into the amazon.com site, Amazon would like to add more servers for better customer experience.

Both the above actions lead to increased customer satisfaction, which will eventually boost profits for the companies. This feature of adding and removing servers is called Dynamic Scaling and is a unique feature of the Cloud. Simply put, the users of the Cloud can scale to thousands of servers and scale down when appropriate and pay for what they use. However, that flexibility to add/subtract servers does not come with the on-premises servers, which is why the cost is always fixed. Also, during the slack time, many resources remain under- utilized which is a wastage of CAPEX. One way of adding and deleting the EC2 instances is to do it manually which may lead to extra manual effort, increase in costs, and inaccurate results.

Another approach is to use Auto Scaling to manage the EC2 instances automatically. As Auto Scaling adds more EC2 instances, the software/application installation and configuration can be automated using the AMI (Amazon Machine Images). In the previous use case, we have seen how to capture custom metrics (number of users logged in) in the CloudWatch. Here, we would need the same metric to manage (add/delete) the EC2 instances depending on the number of users logged into the website.

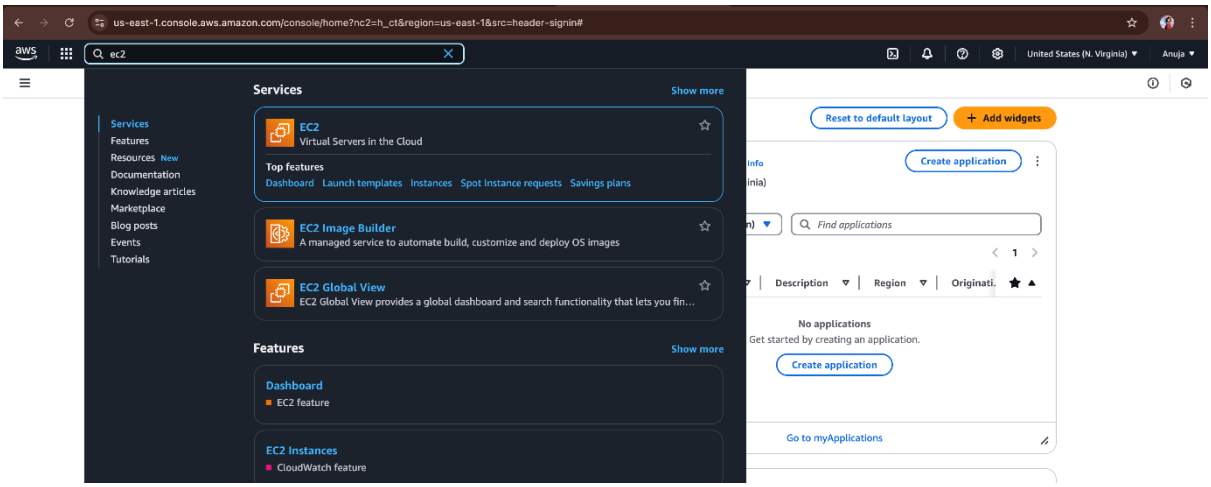
Tools required: AWS Services - CloudWatch, Auto Scaling, EC2

Expected Deliverables:

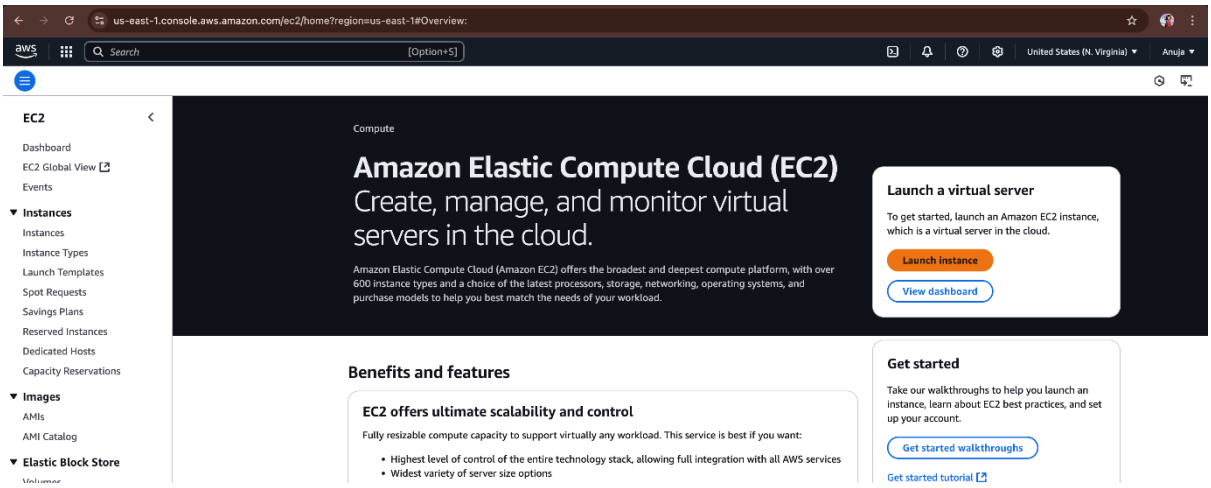
- Use Auto Scaling to manage the EC2 instances
- Use EC2 instance and capture the metrics in the CloudW

Step1: Creating EC2 Instance.

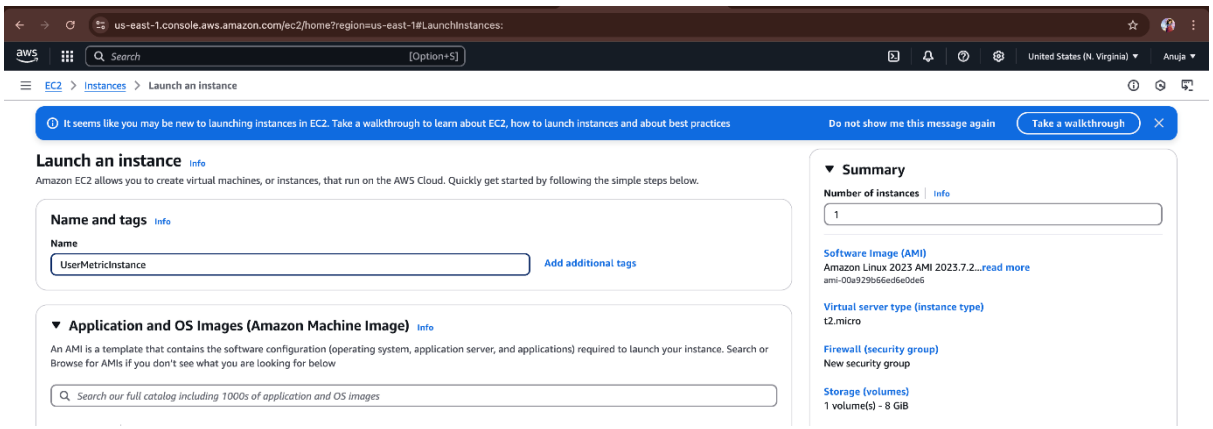
Go to AWS console and search for EC2.



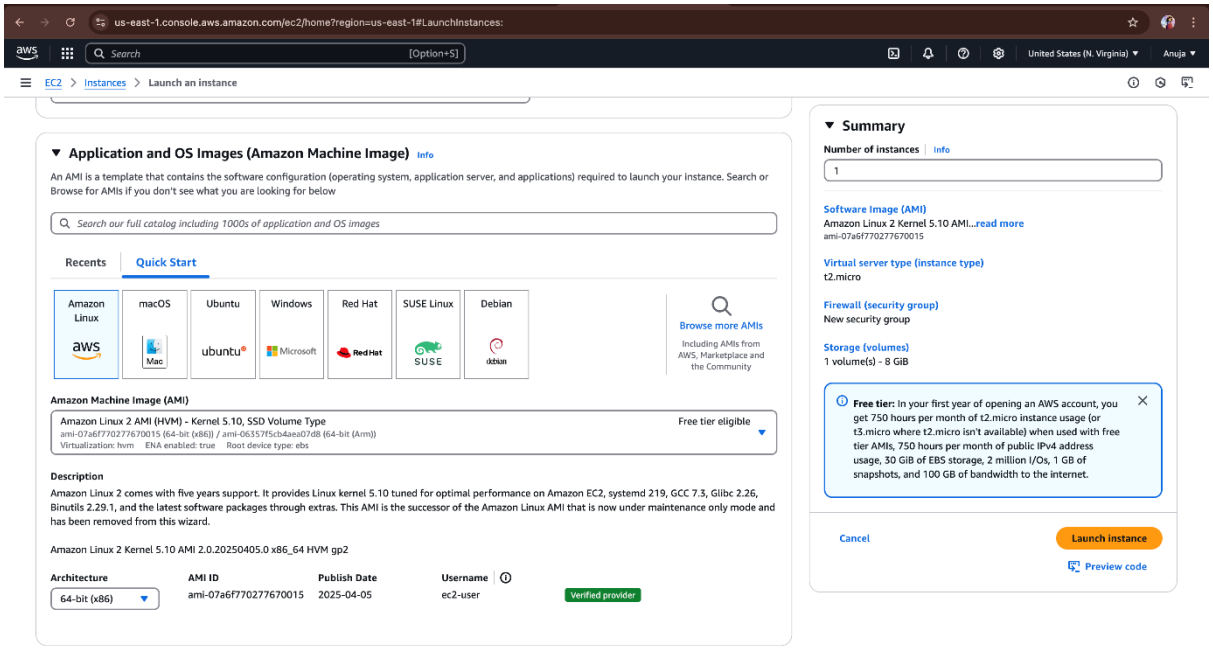
Click on launch instance.



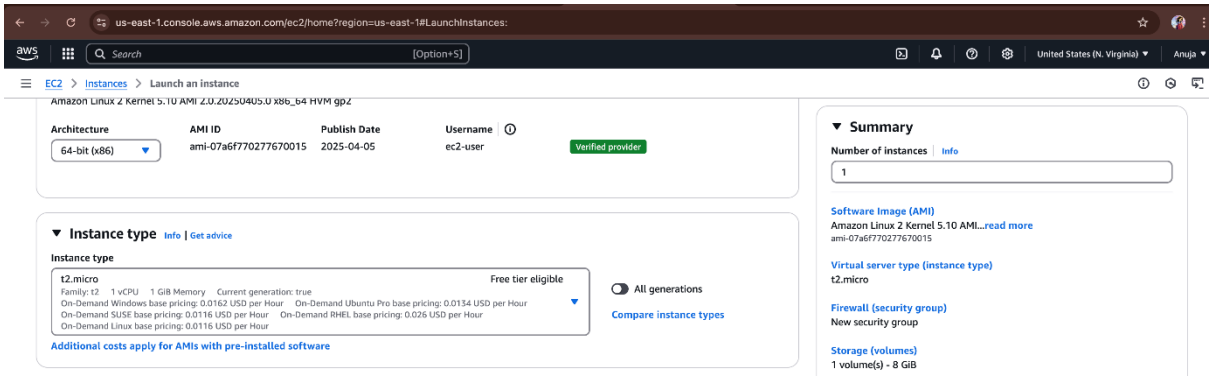
Give a name to your instance as Usermetricsinstance.



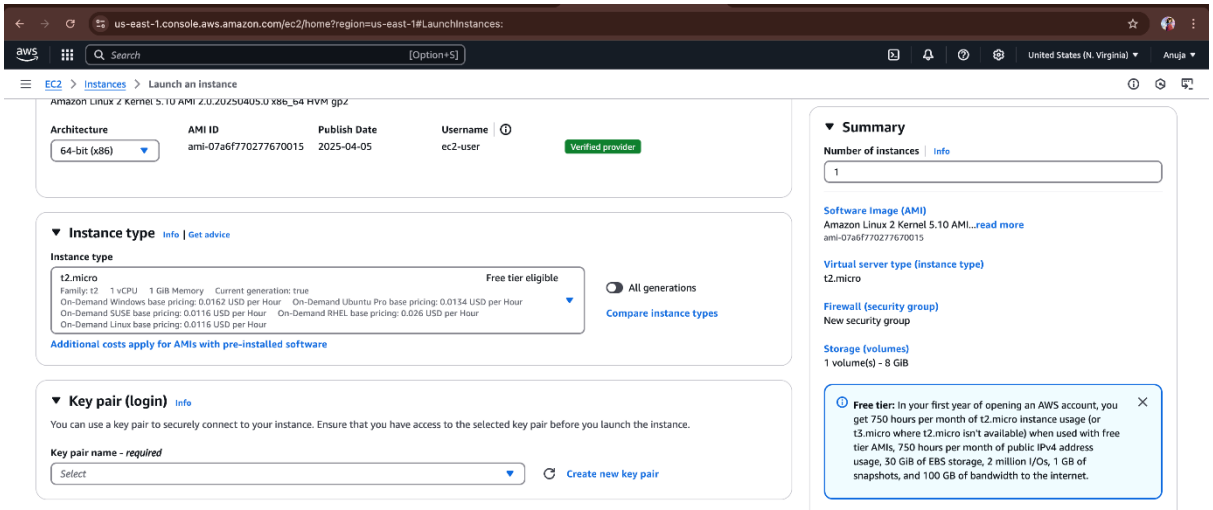
And select Amazon 2 AMI in amazon machine image section.



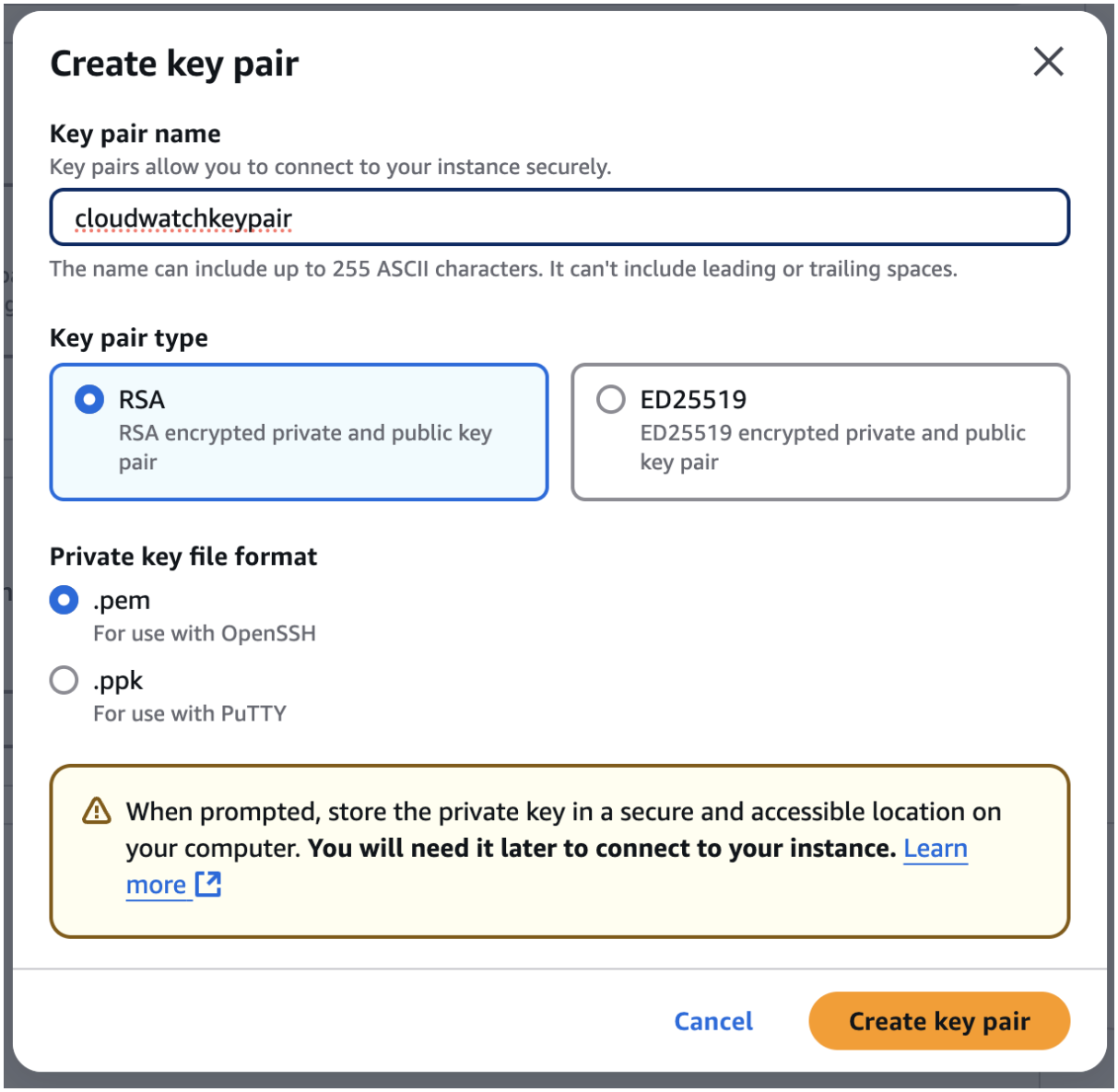
And instance type as t2.micro



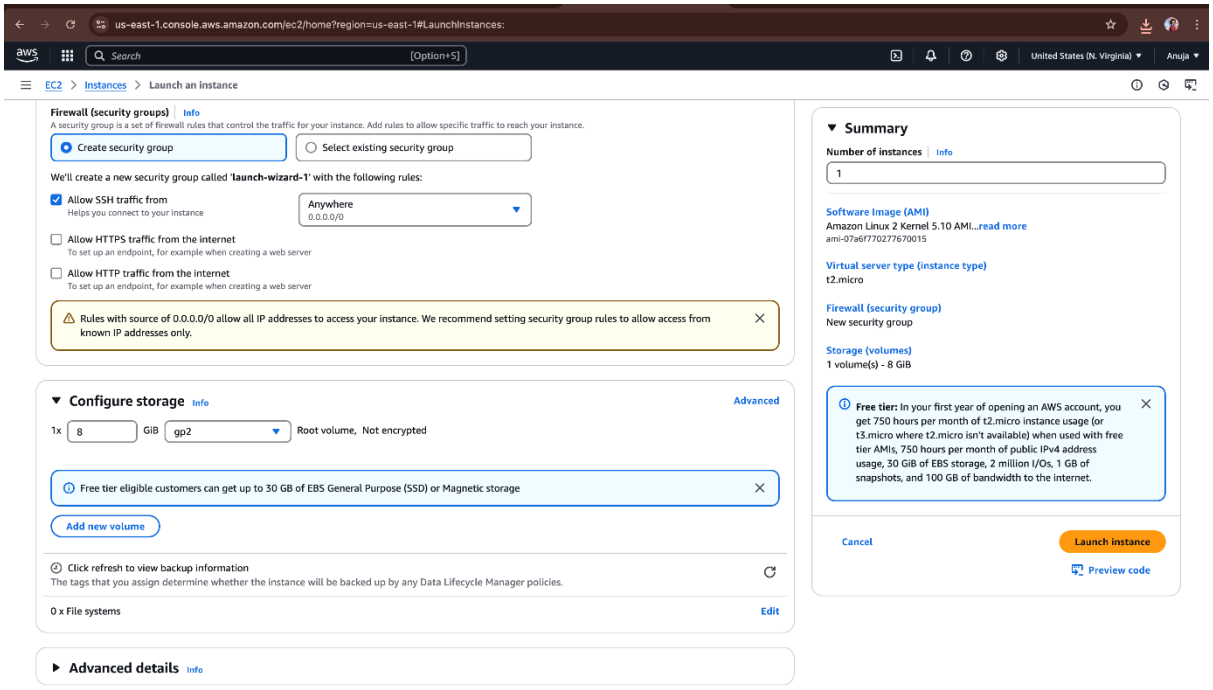
And in keypair section select new keypair.



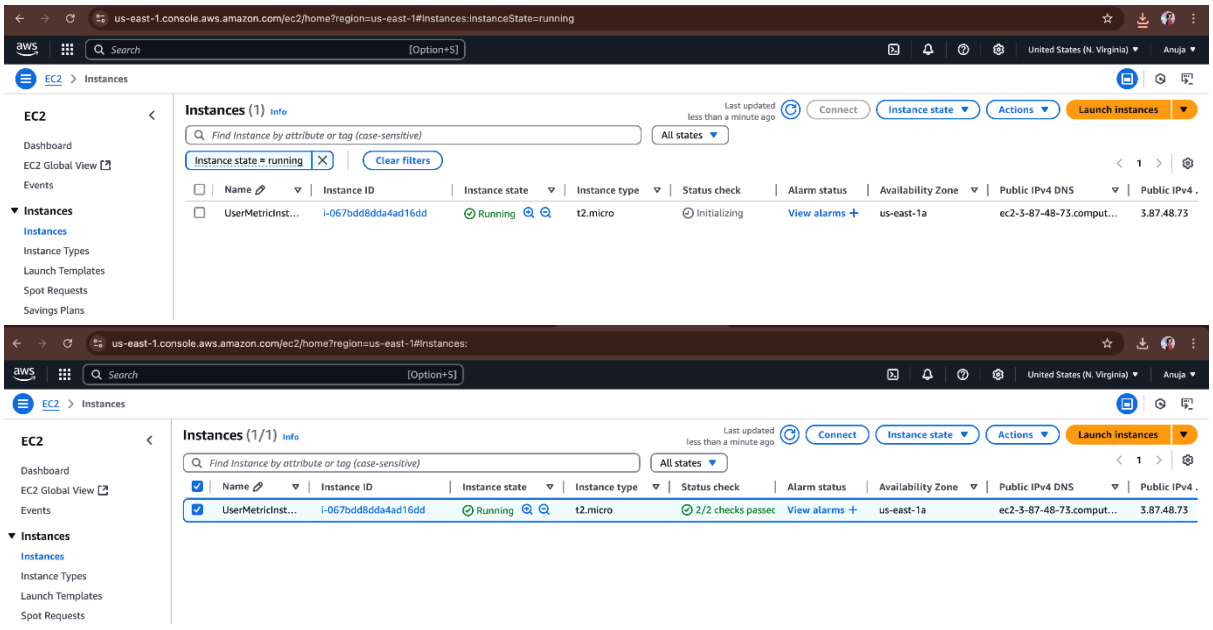
Give the keypair a name and select private key file format as .pem



And make all remaining things as default and launch instance.

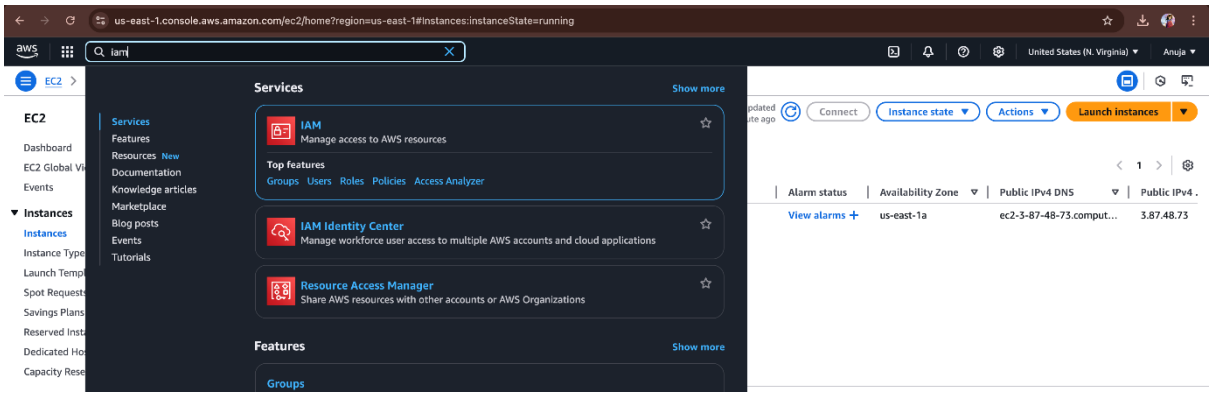


After launching an instance you can see new instance is launching in the instances section.

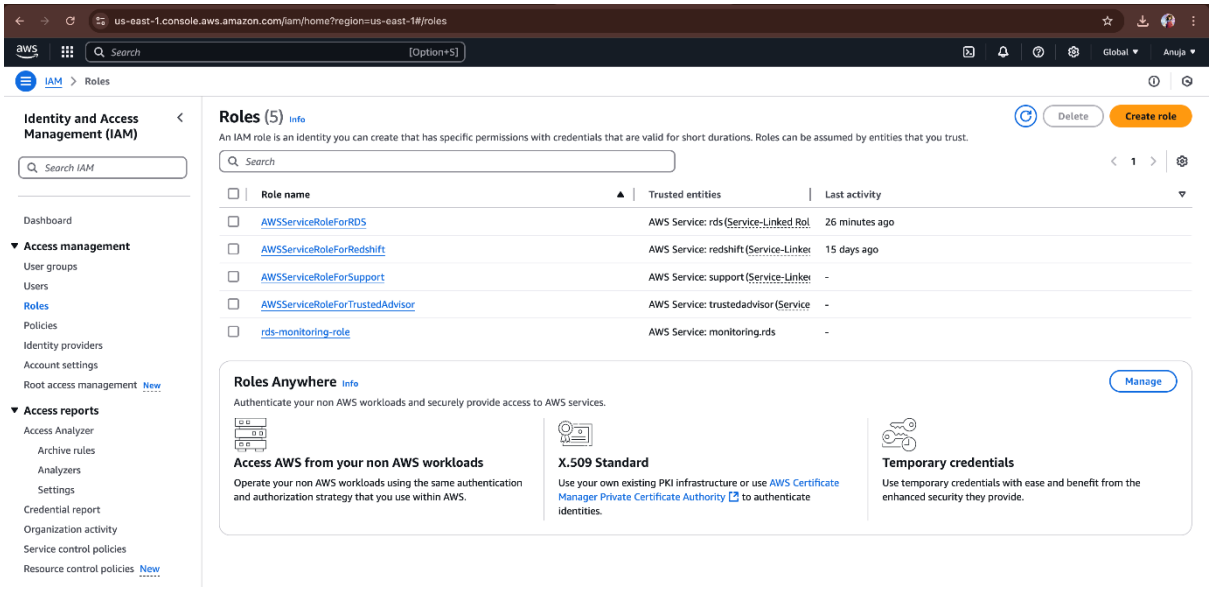


Step2: Creating IAM role and attaching to EC2 Instance.

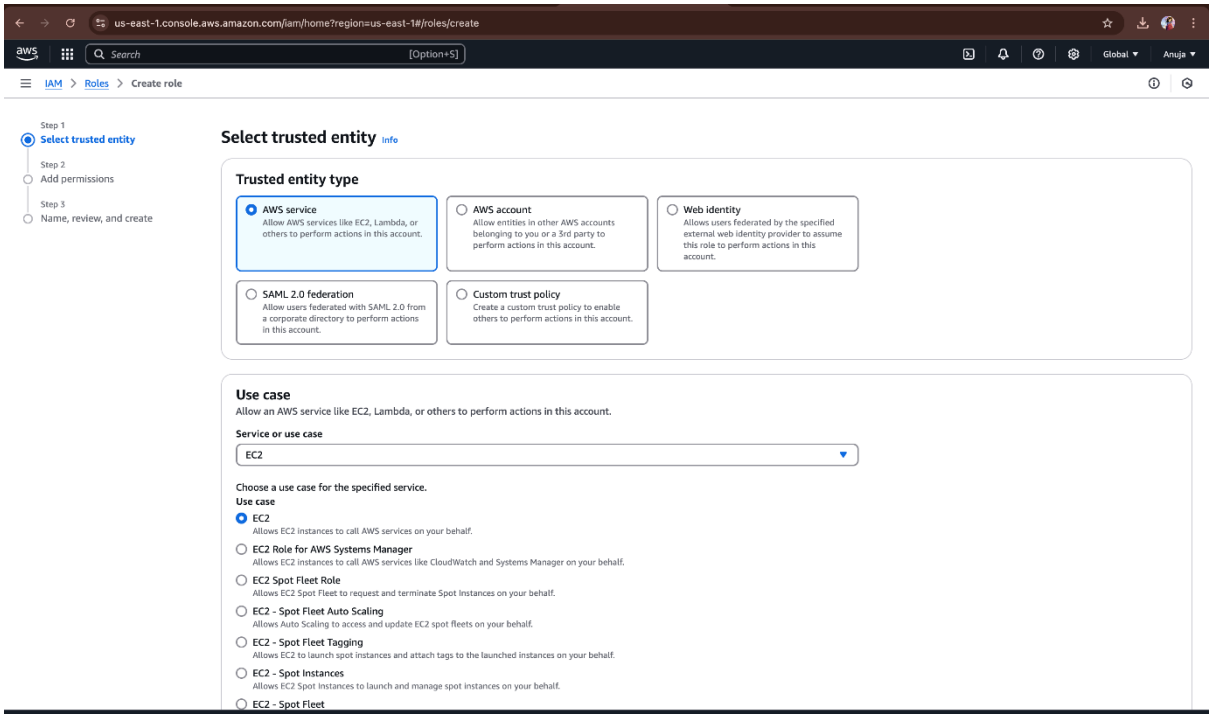
Go to dashboard and click on IAM.



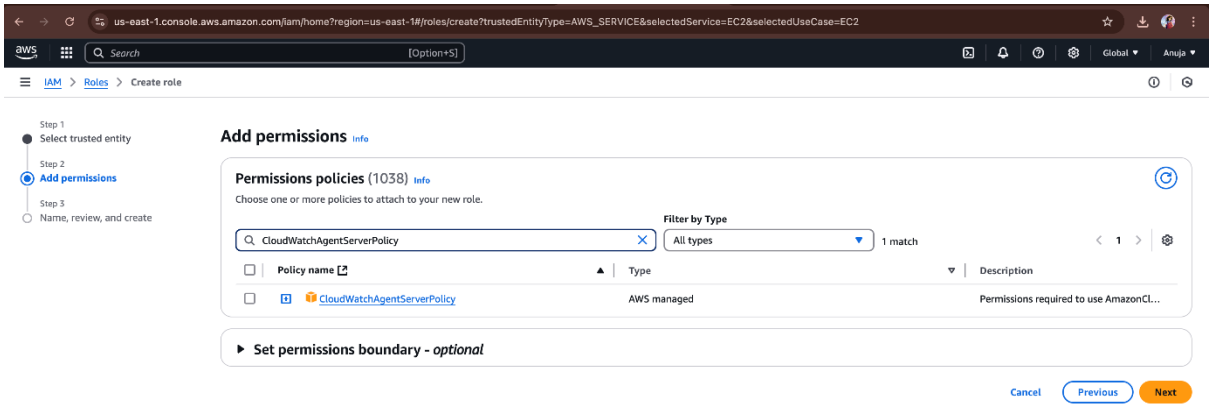
click on create role.

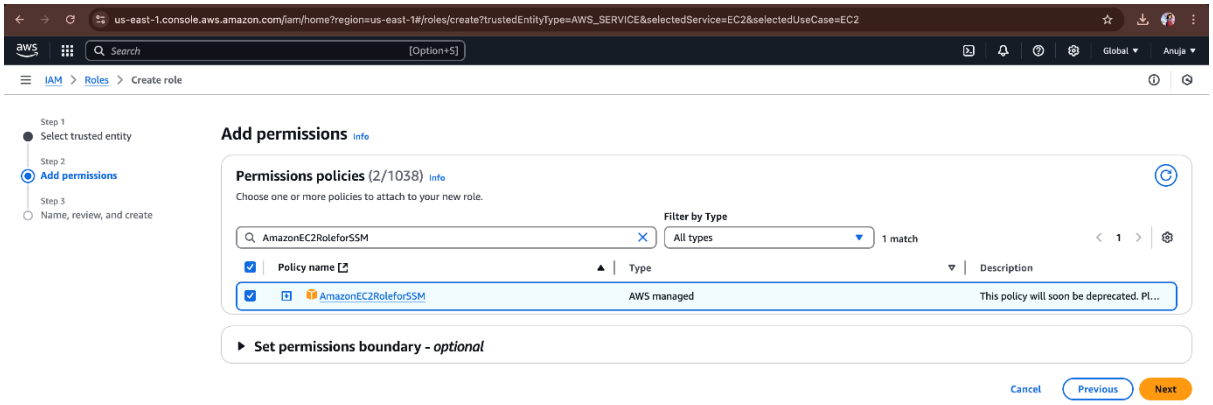


Select trusted entity as AWS service and use case as EC2.

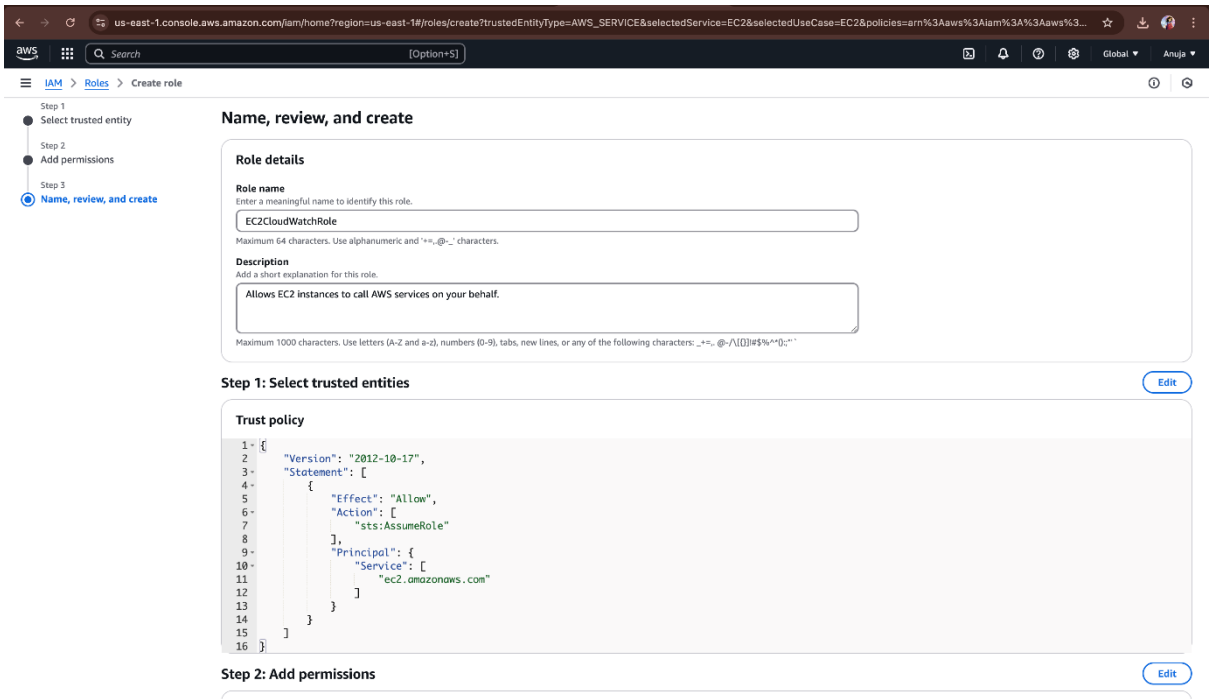


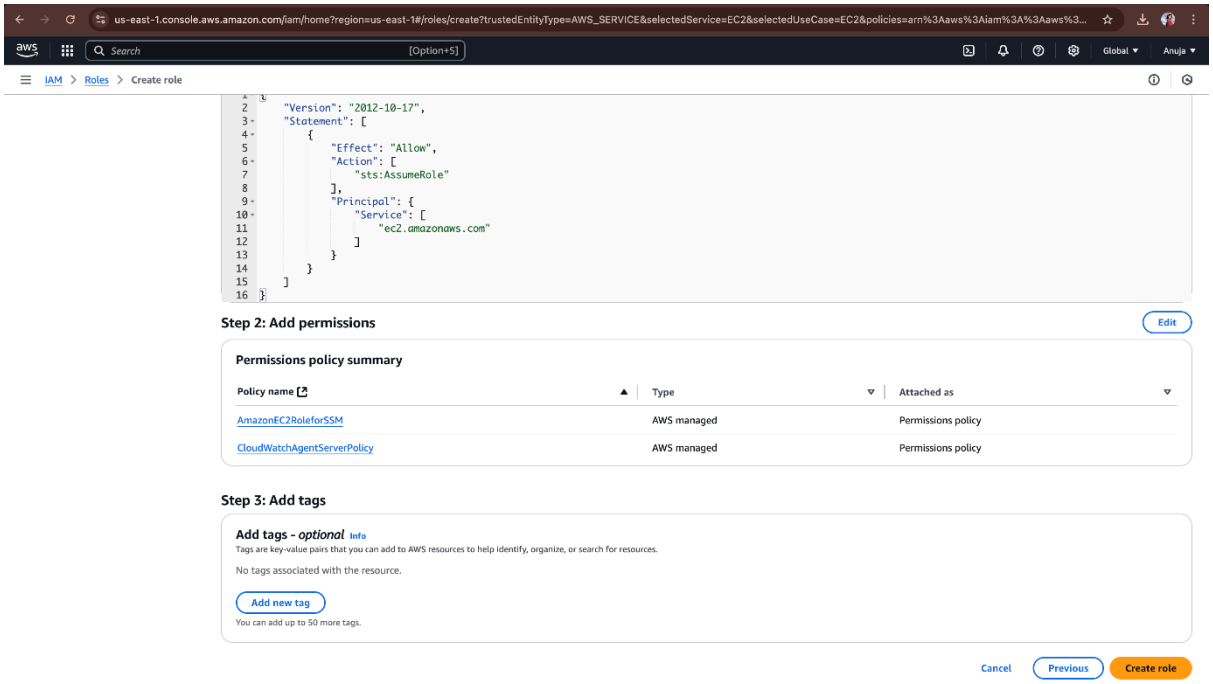
And then select next and click on permission policy in that select this 2 policies.
CloudWatchAgentServerPolicy
AmazonEc2RoleForSSM



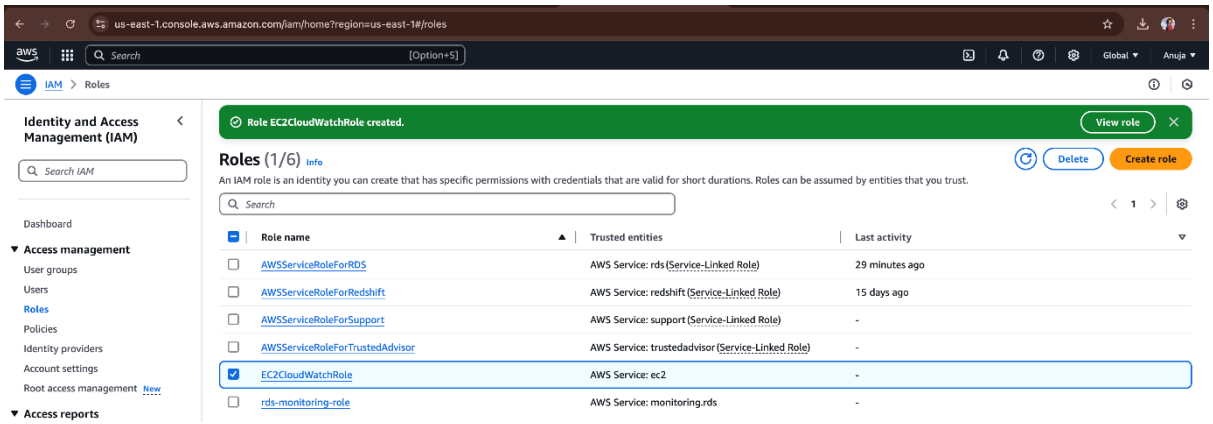


Review all the sections and click on create role.

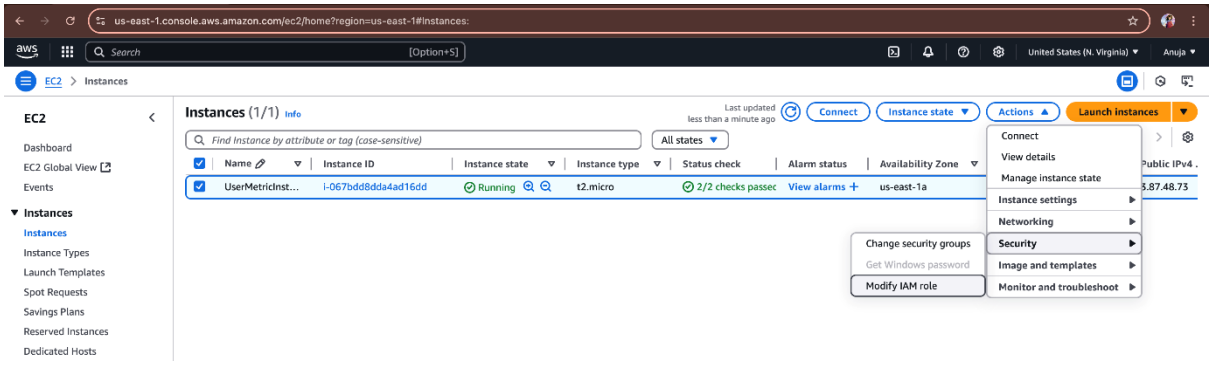




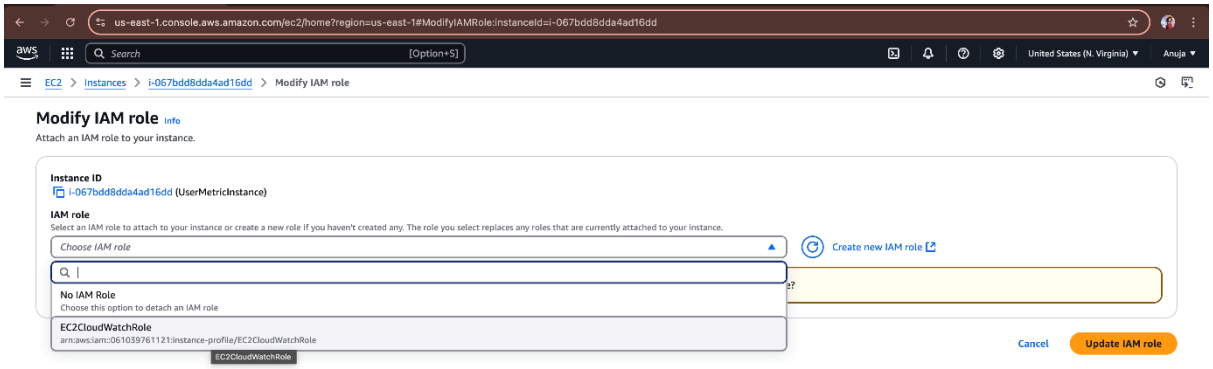
You Can see your role was creating.



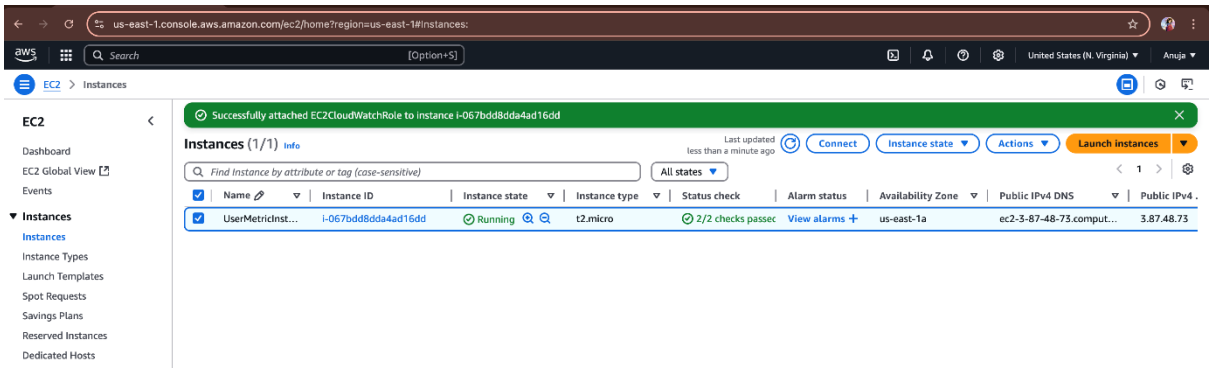
Then after creating the role we need to attach this role to our created EC2 Instance. For that go to EC2 select your instance and actions→ security → Modify IAM role.



Choose your newly created IAM role and attach.

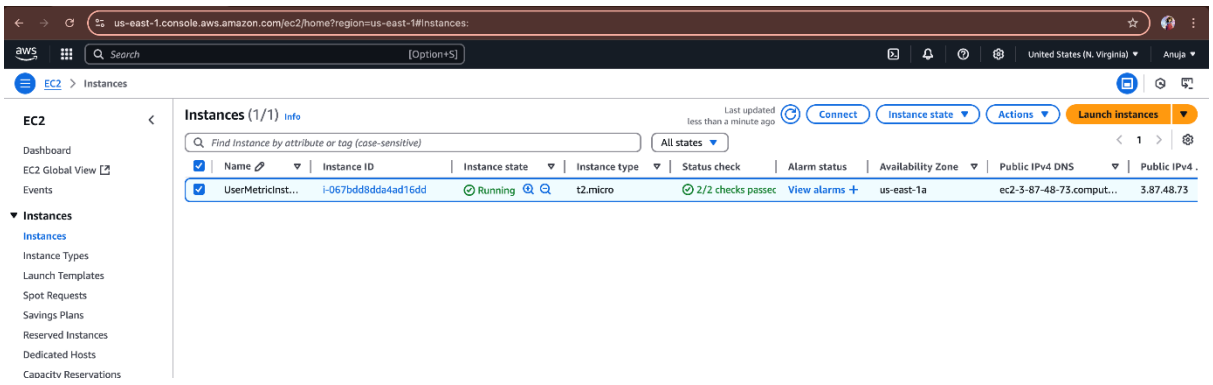


You can see a pop up that your role is successfully attached to your instance.

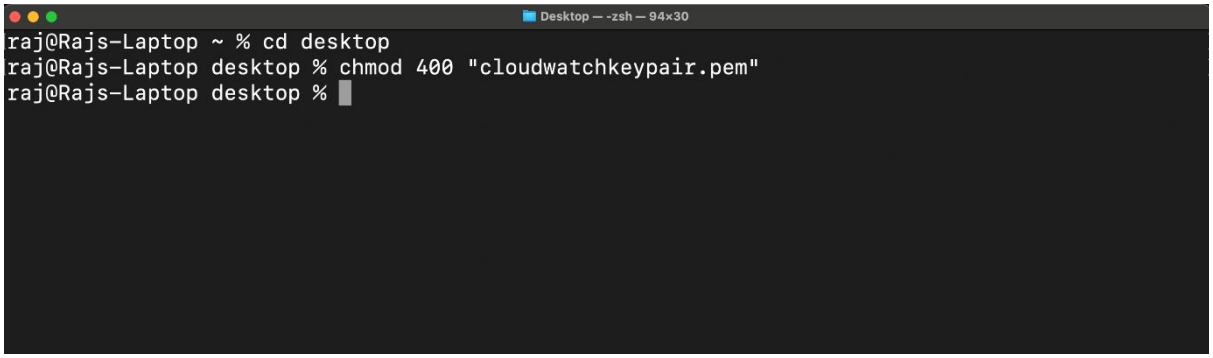


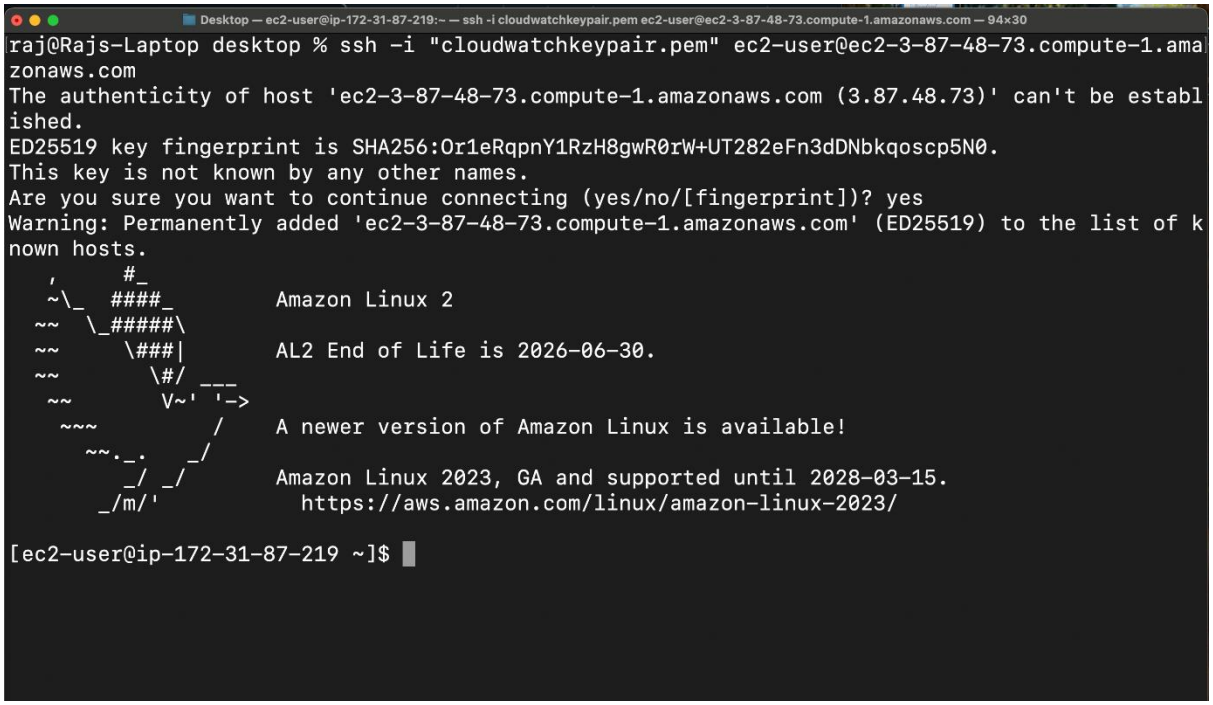
Step3: Pushing custom metrics to cloud watch

Go to Ec2 and connect to ssh.

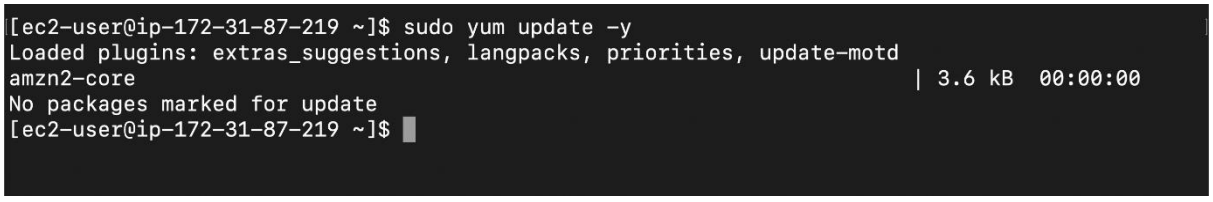


Connect to your EC2 instance using terminal.

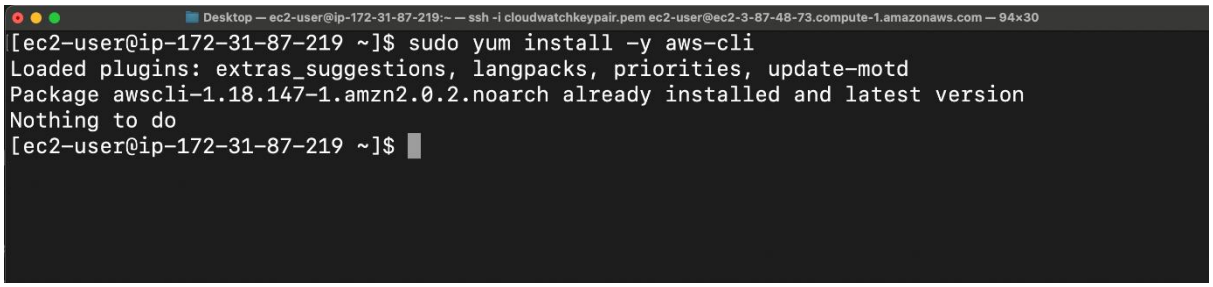




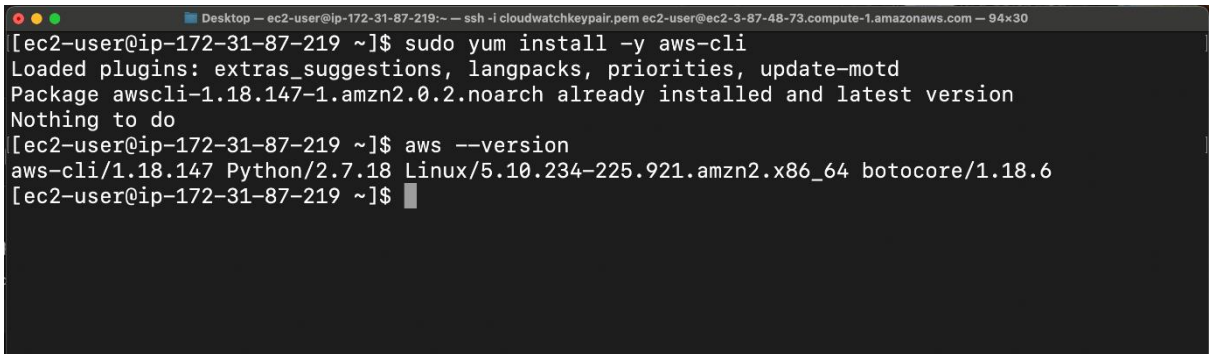
Then install all the updates using the command yum update.



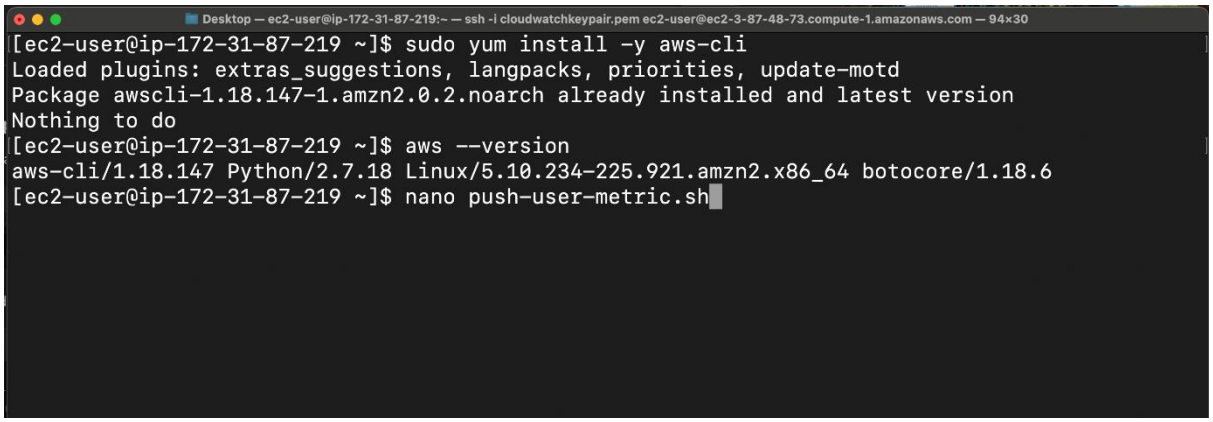
Then now need to install AWS-Cli.



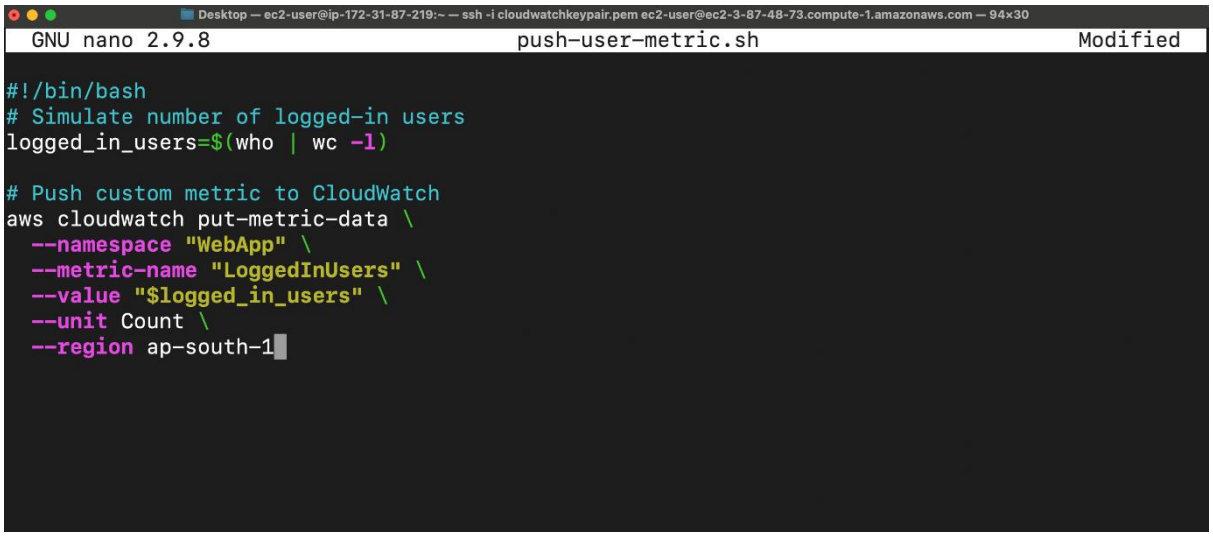
Check AWS version.



Now open the file using Nano editor to push the metrices.



In that file insert this code.



After saving that file you can see the content of the file was updated.

```
Desktop — ec2-user@ip-172-31-87-219:~ — ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com — 94×30
[ec2-user@ip-172-31-87-219 ~]$ cat push-user-metric.sh
#!/bin/bash
# Simulate number of logged-in users
logged_in_users=$(who | wc -l)

# Push custom metric to CloudWatch in us-east-1 region
aws cloudwatch put-metric-data \
  --namespace "WebApp" \
  --metric-name "LoggedInUsers" \
  --value "$logged_in_users" \
  --unit Count \
  --region us-east-1
[ec2-user@ip-172-31-87-219 ~]$
```

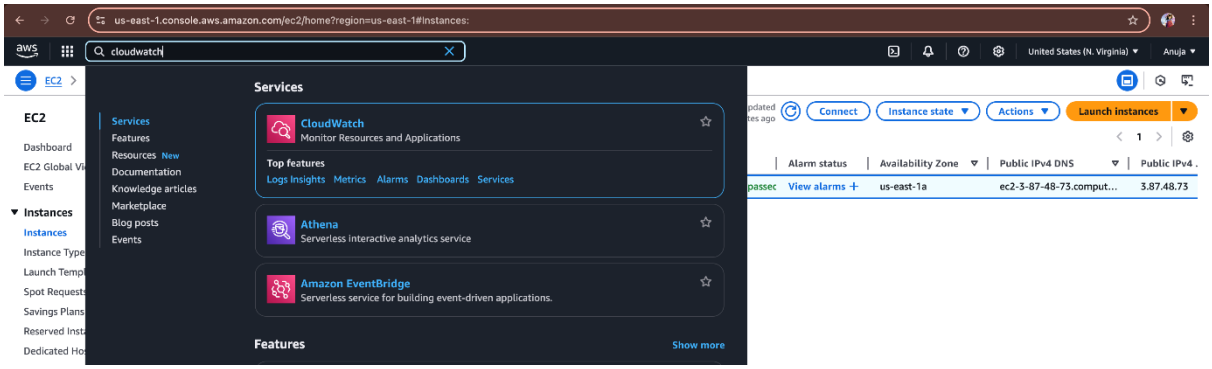
Now the permission of the new created file using the command shown.

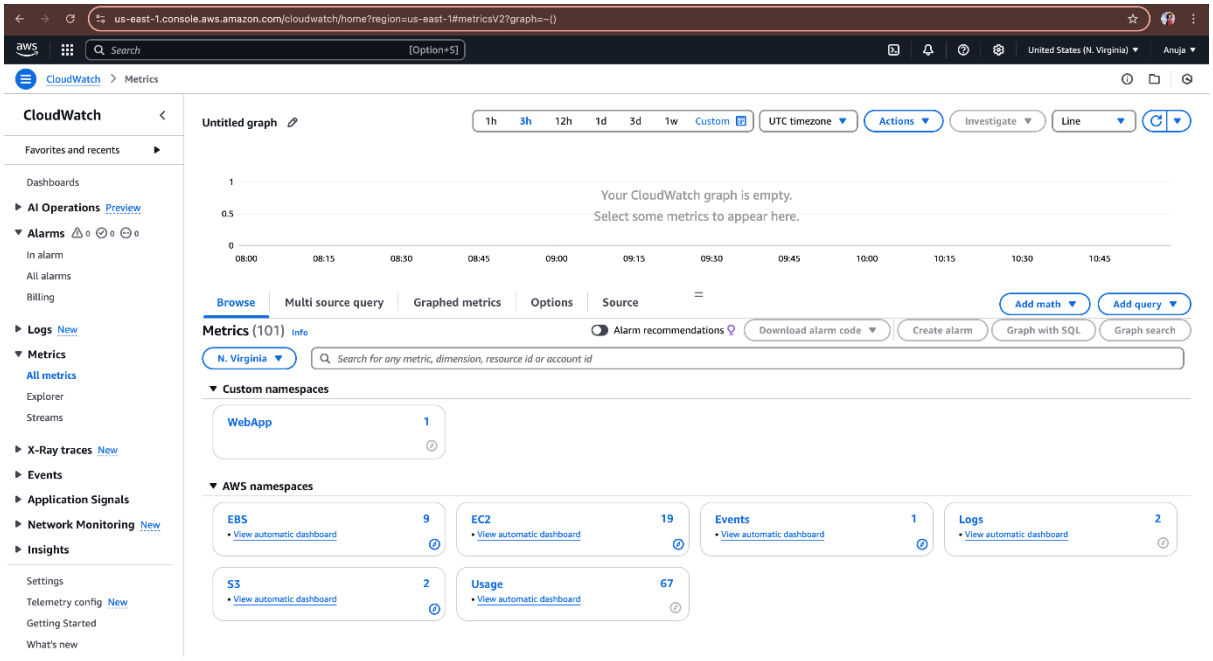
```
[ec2-user@ip-172-31-87-219 ~]$ chmod +x push-user-metric.sh
[ec2-user@ip-172-31-87-219 ~]$
[ec2-user@ip-172-31-87-219 ~]$
```

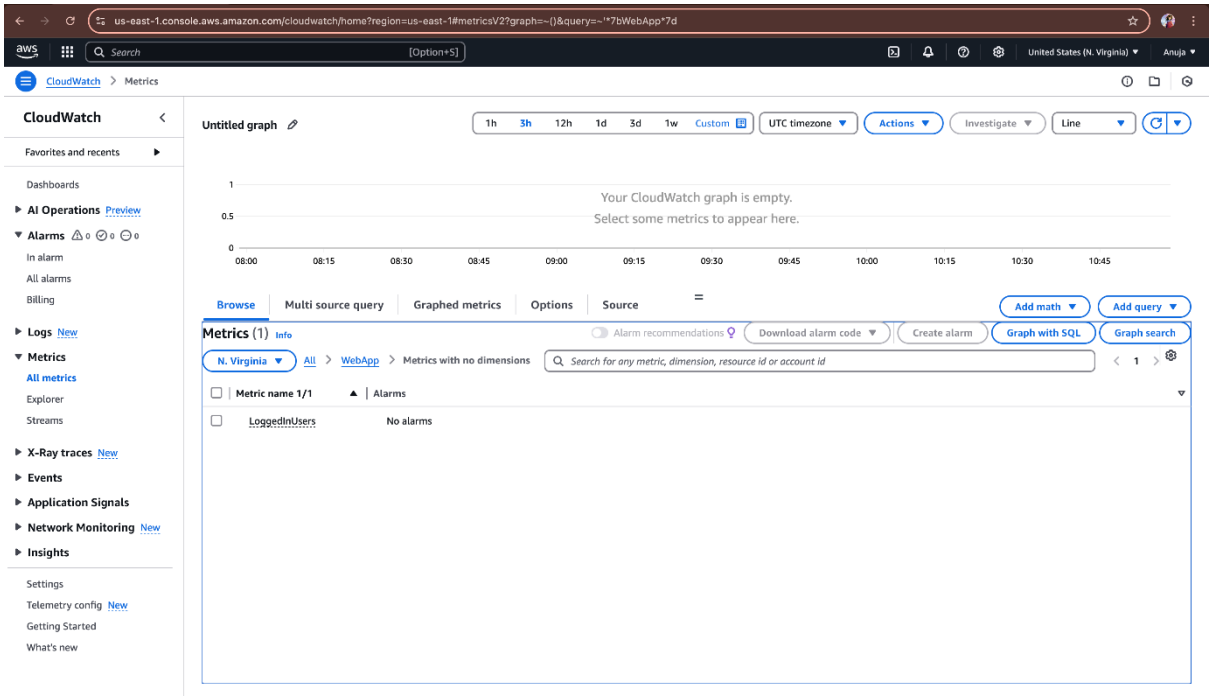
Now final step need to push metrics.

```
Desktop — ec2-user@ip-172-31-87-219:~ — ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com — 94×30
[ec2-user@ip-172-31-87-219 ~]$ ./push-user-metric.sh
[ec2-user@ip-172-31-87-219 ~]$
```

Now go to cloud watch and check for newly created metrices in cloudwatch metrices.

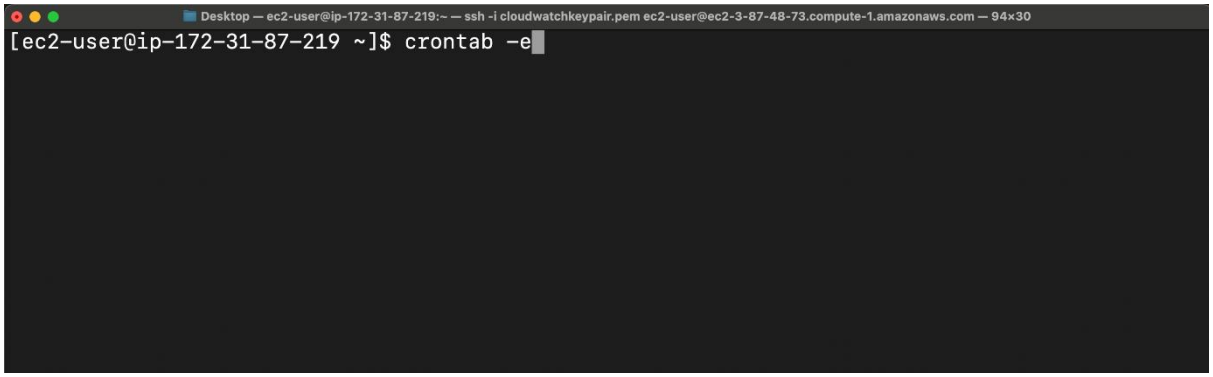




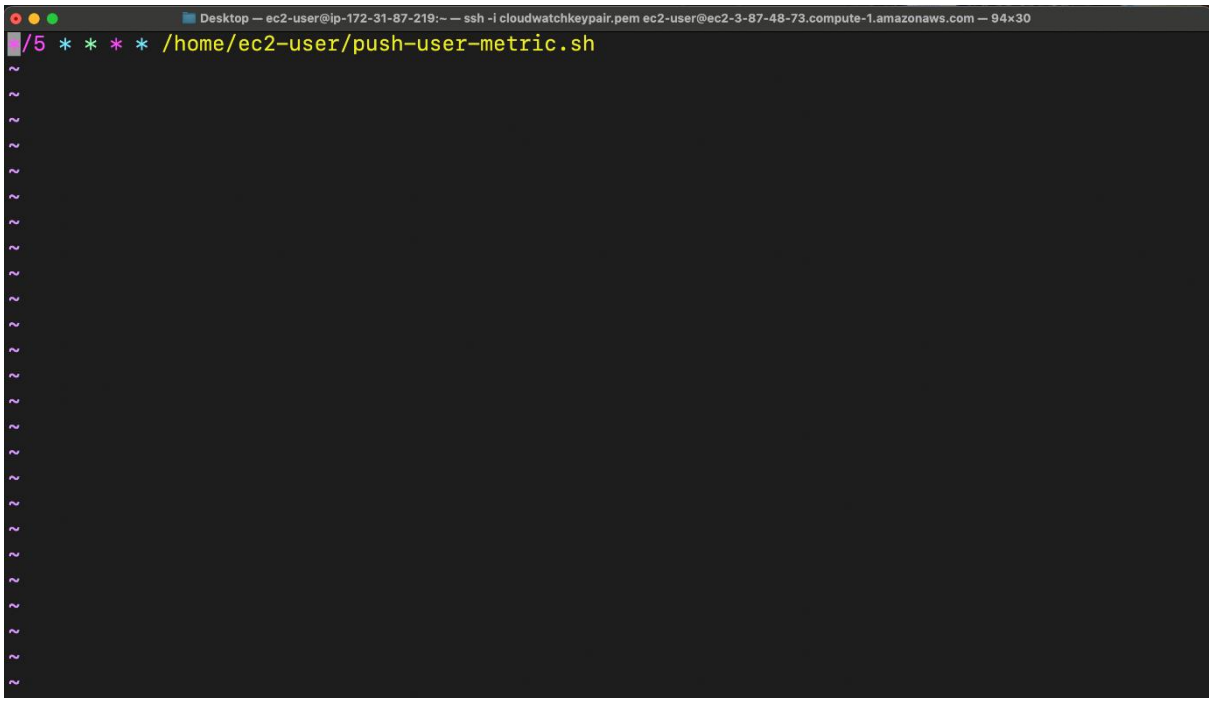


Step 4: Creating a Image for our instance.

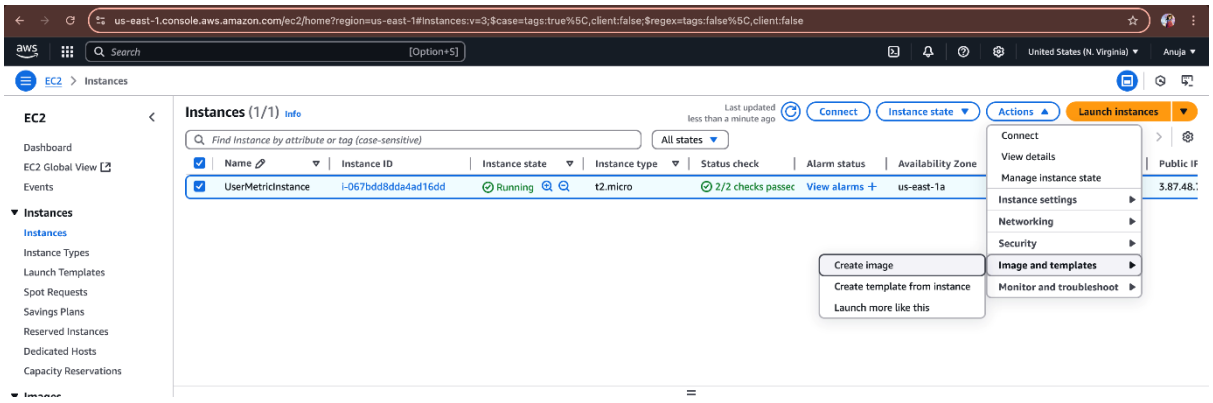
First you need to open your terminal and open crontab -e



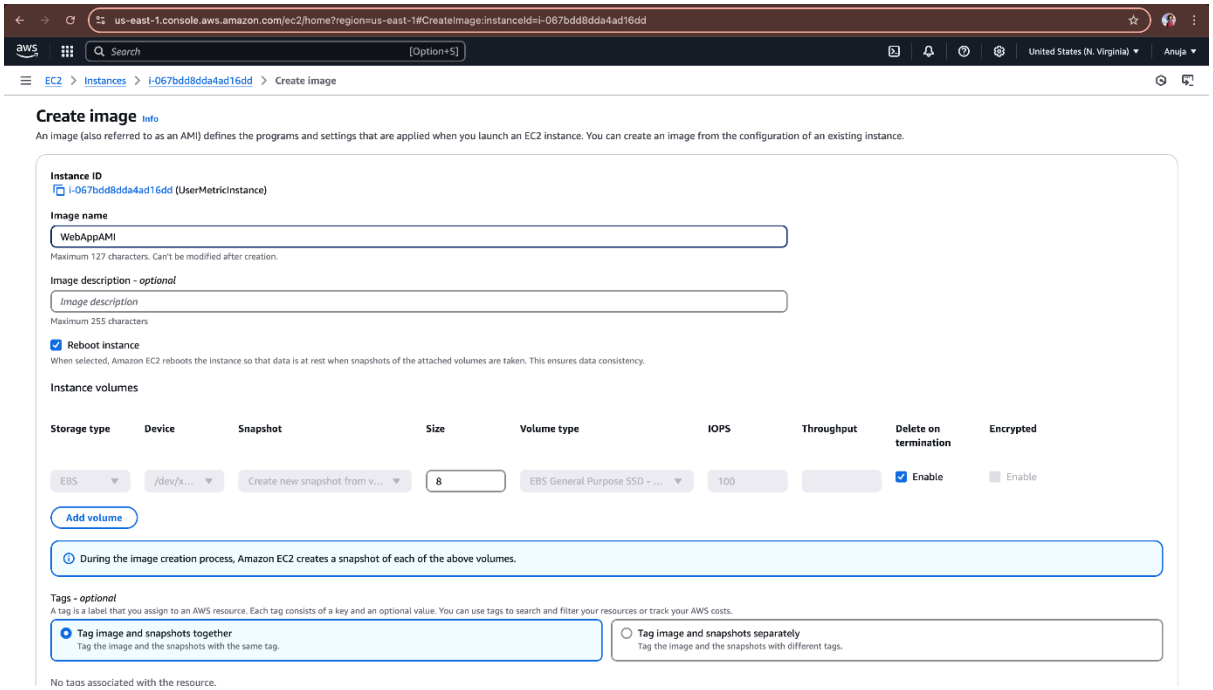
Again push your metrices.



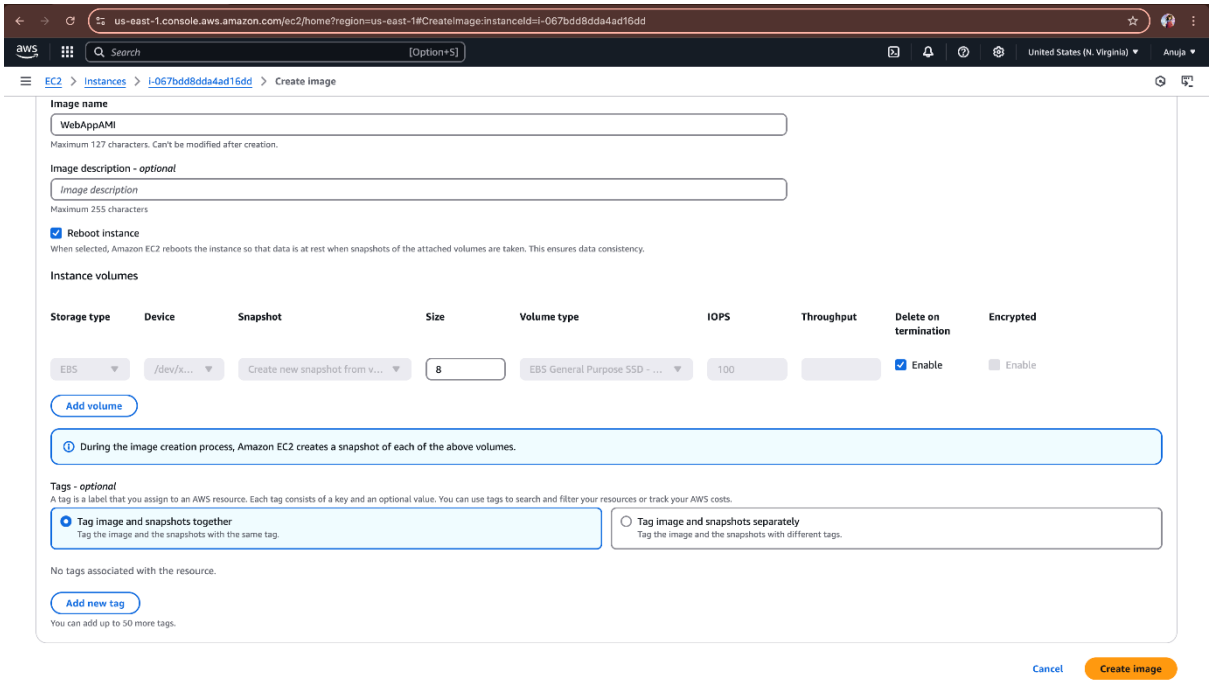
Then go to your instance select your instance and click on Actions → Image and templates → create image.



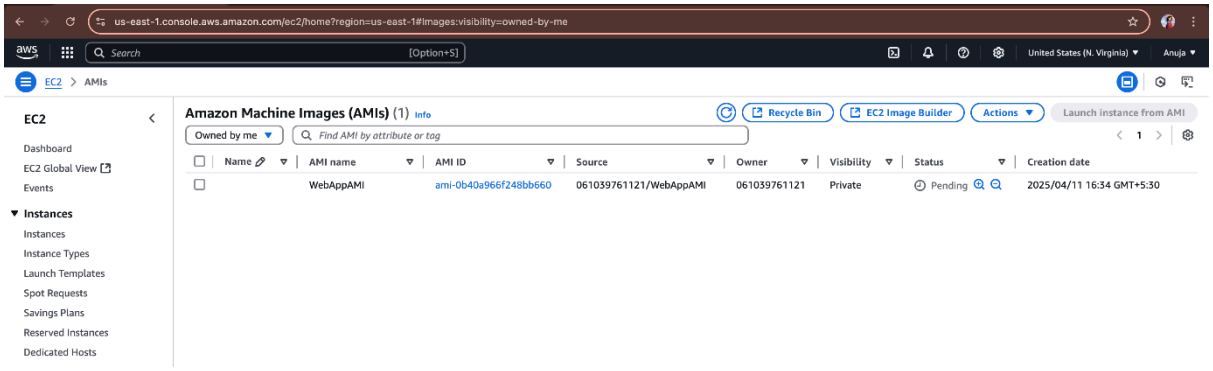
Give your image name and click on next.



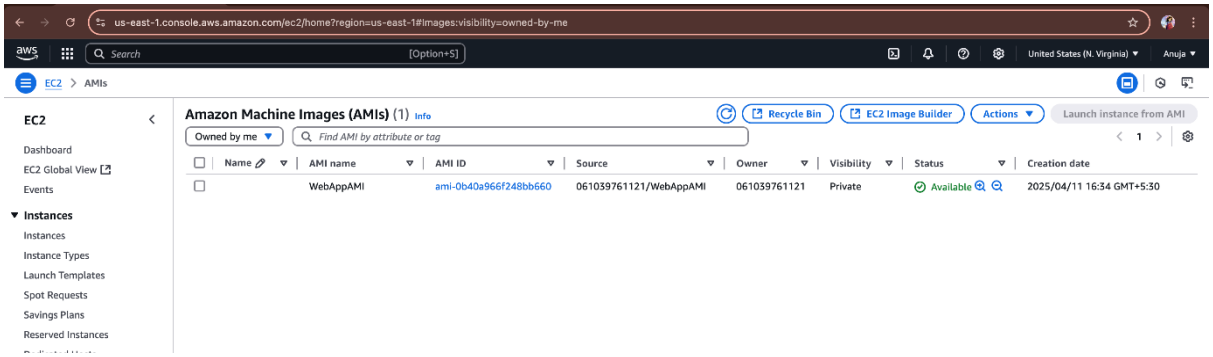
Then click on create image.



In AMI’S section you can see your image is creating.

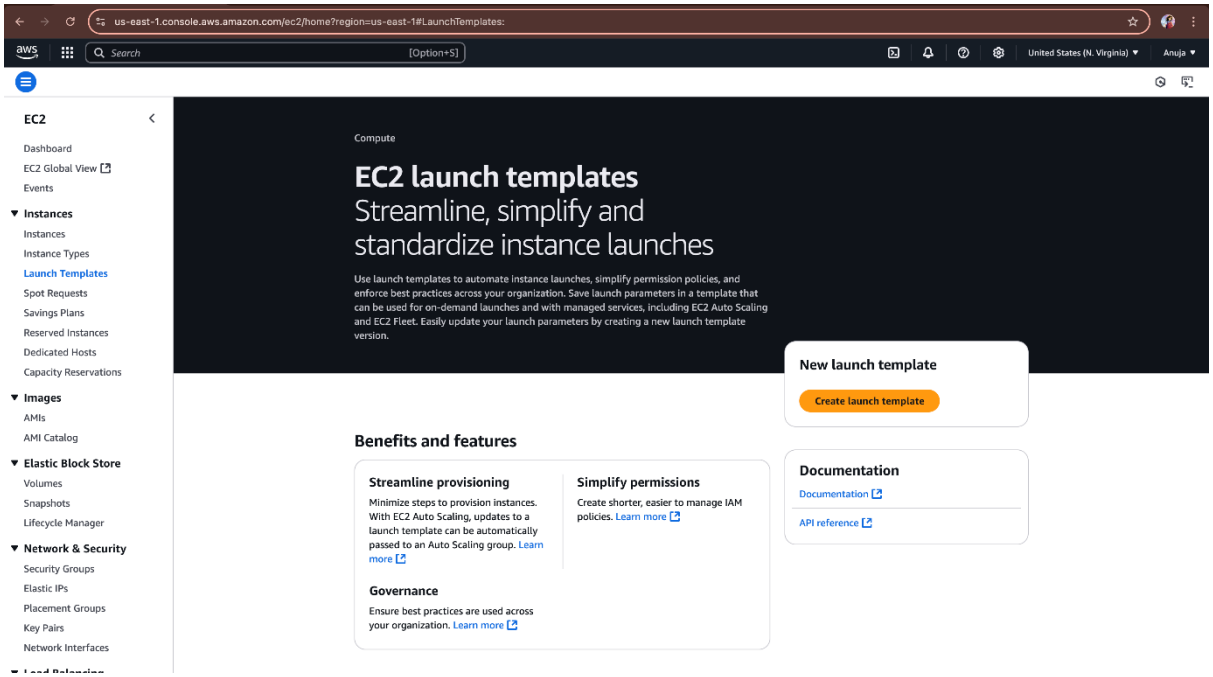


Once your image is available you can ready to use it.

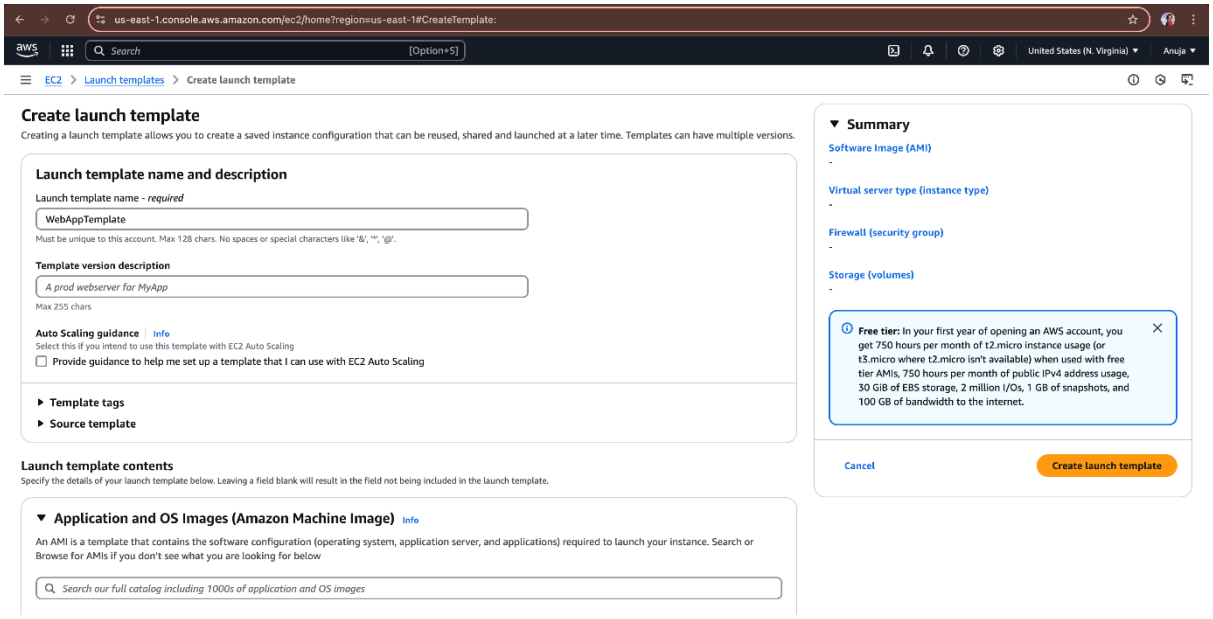


Step 5: Creating a launch template.

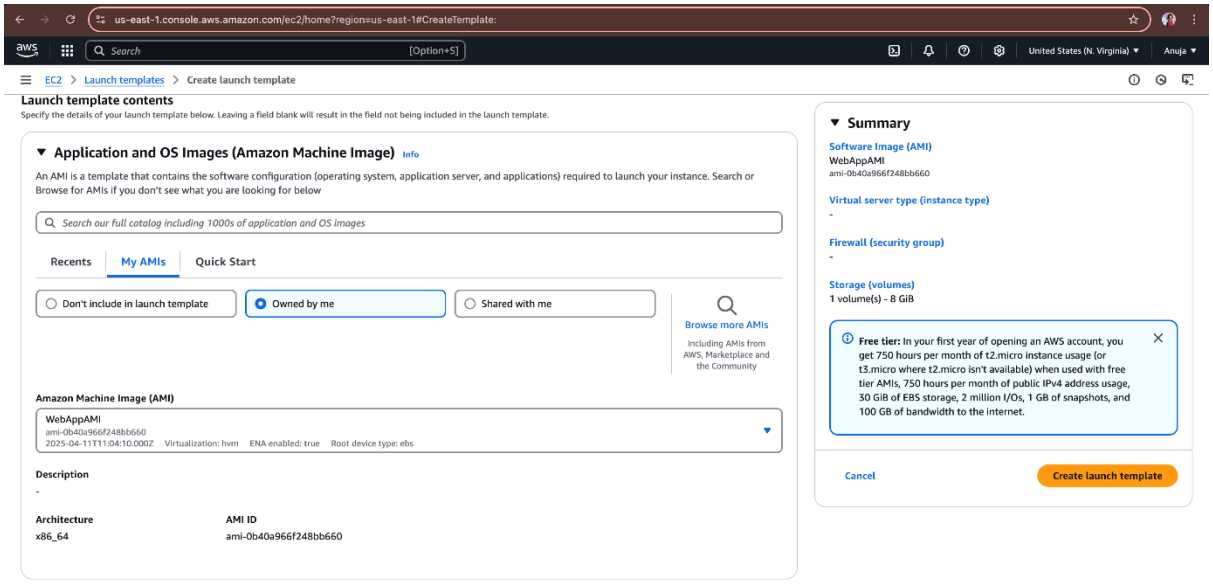
Go to EC2 in instance section you can find launch template. Then press on create launch template



Give your launch template a name.

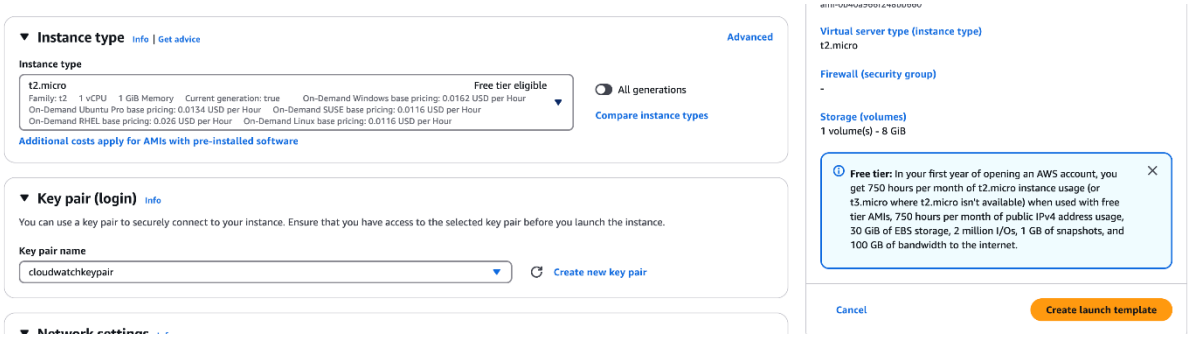


In applications and OS image section click on My Ami's press on owned by me then select the image you created earlier.

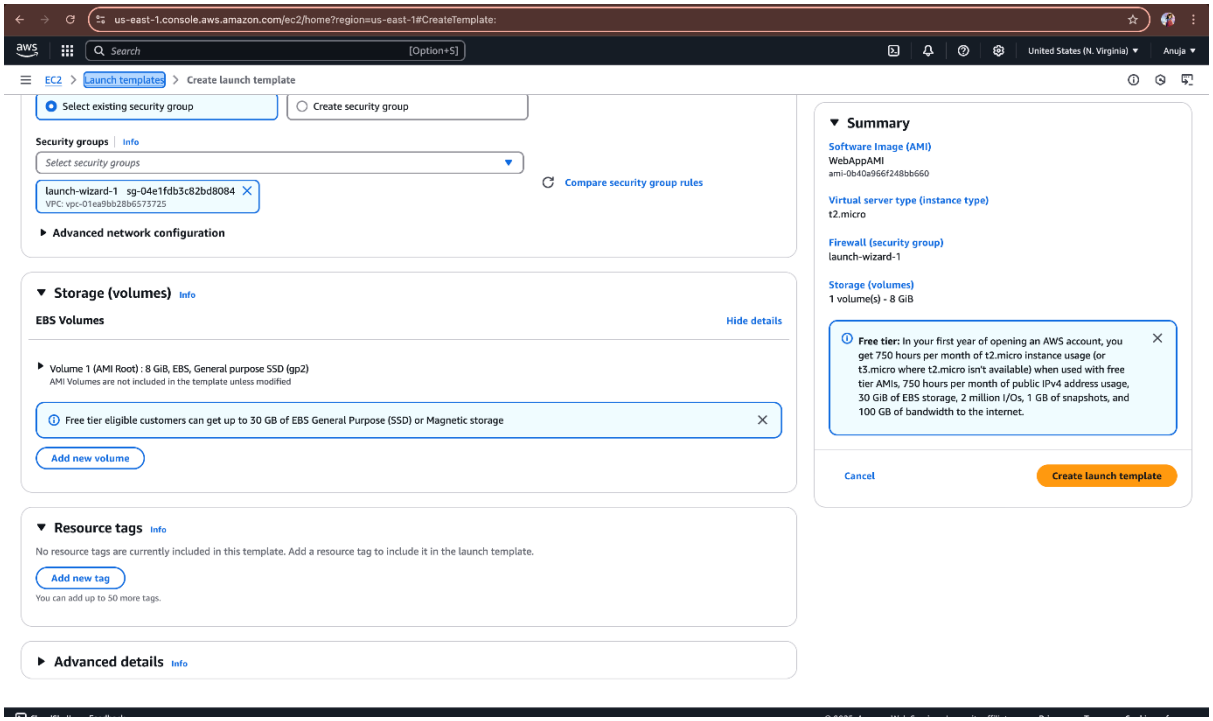


.

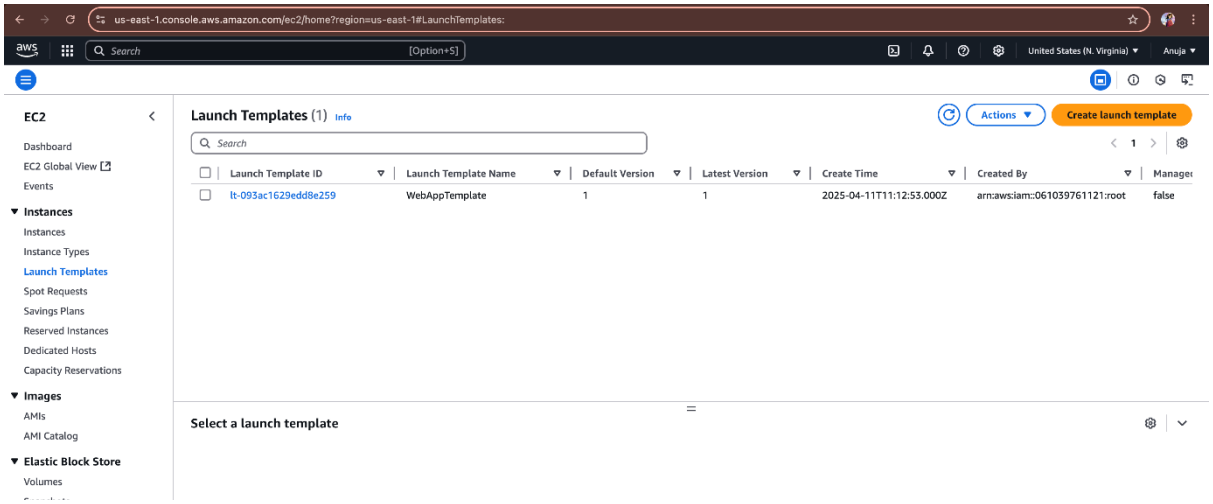
Select instance type as t2.micro and keypair same we have created earlier.



Keep rest all setting as default. And press on create launch template.

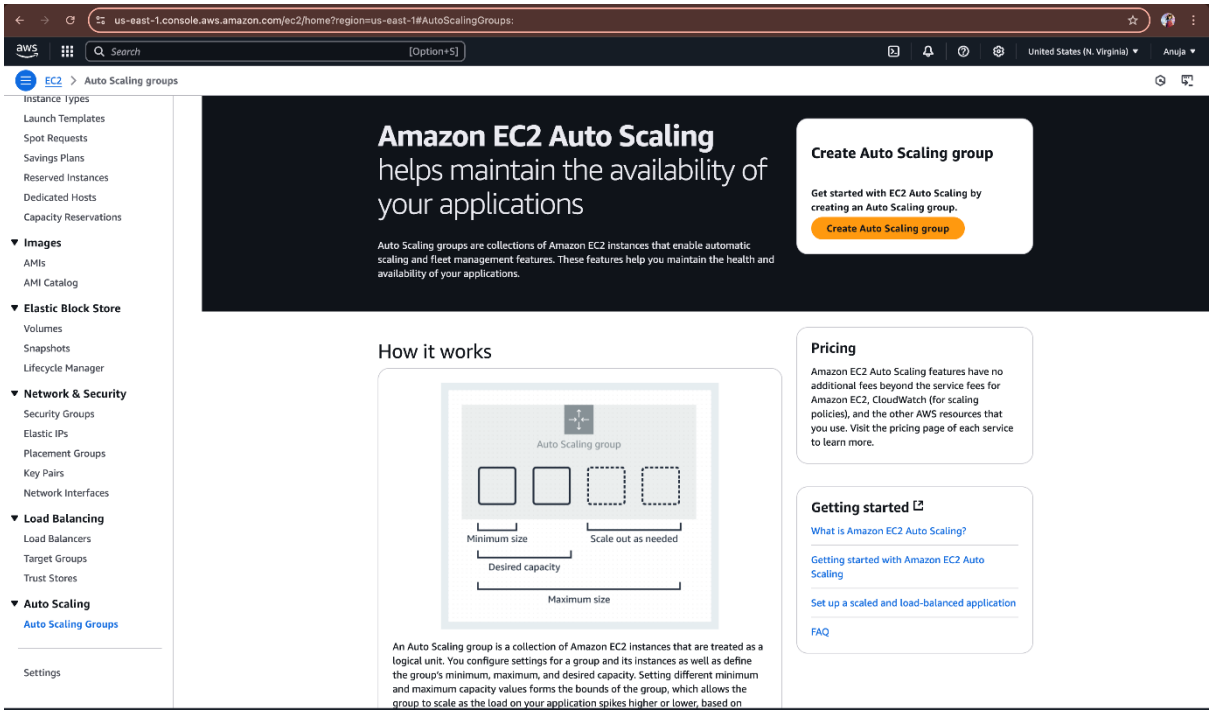


You can see you have launched your template.

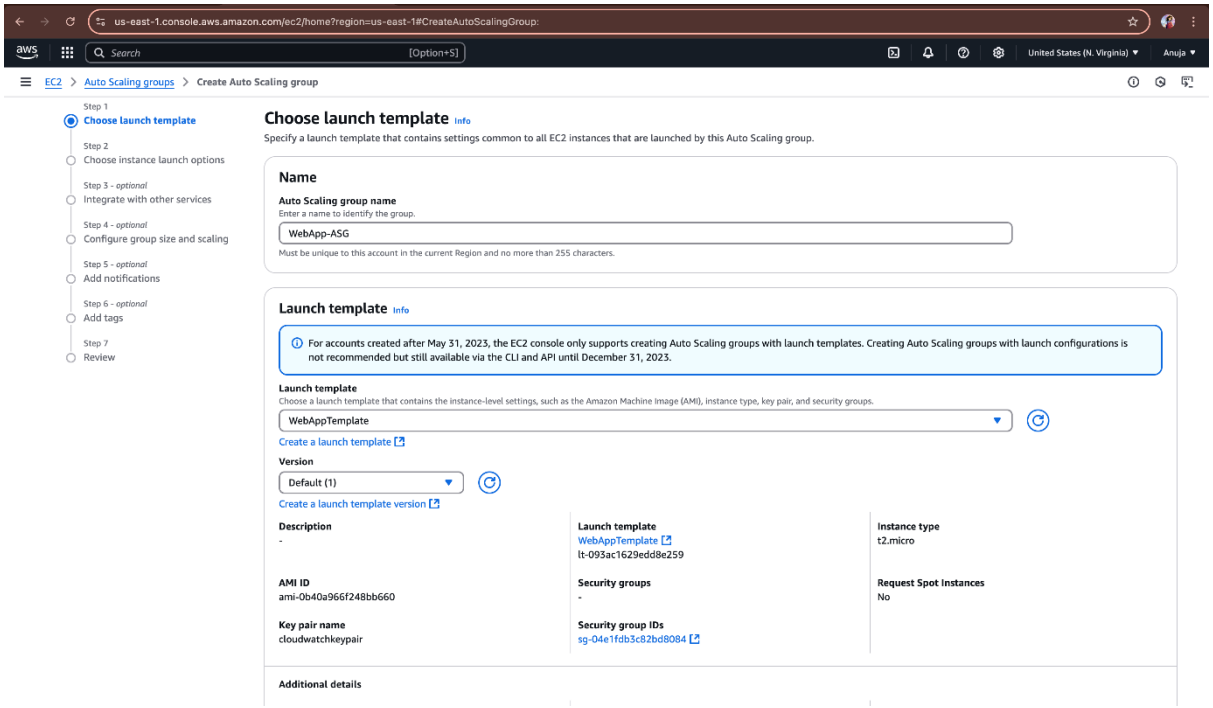


Step 6: Now its time to create Autoscaling group.

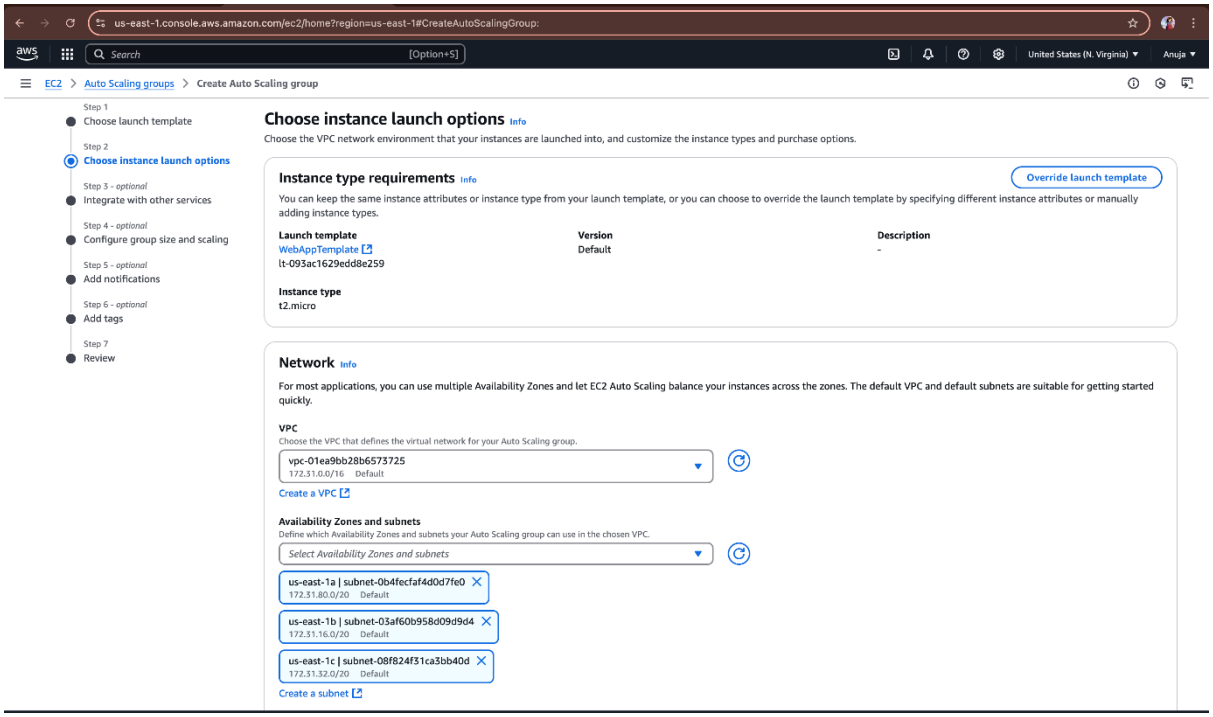
Go to EC2 and at down you can find the option as Autoscaling Groups. Go and press on create autoscaling group.



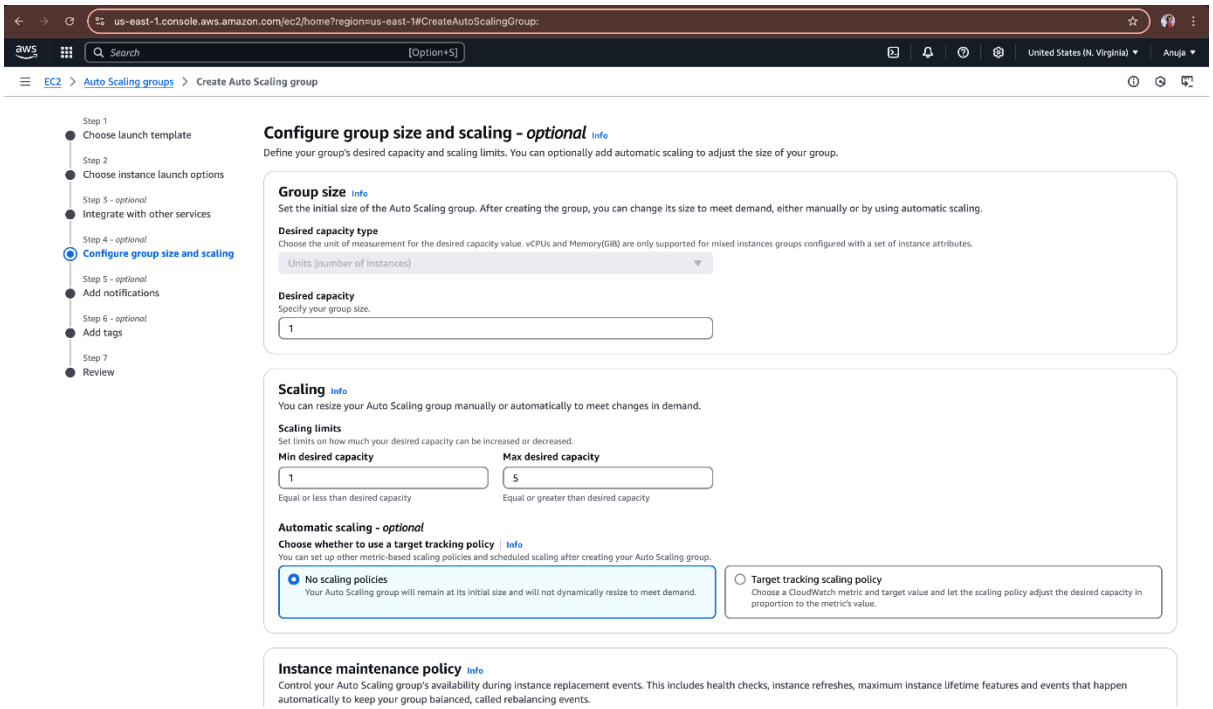
In choose launch template section give a name for your auto scaling group and select the launch template which you have created.



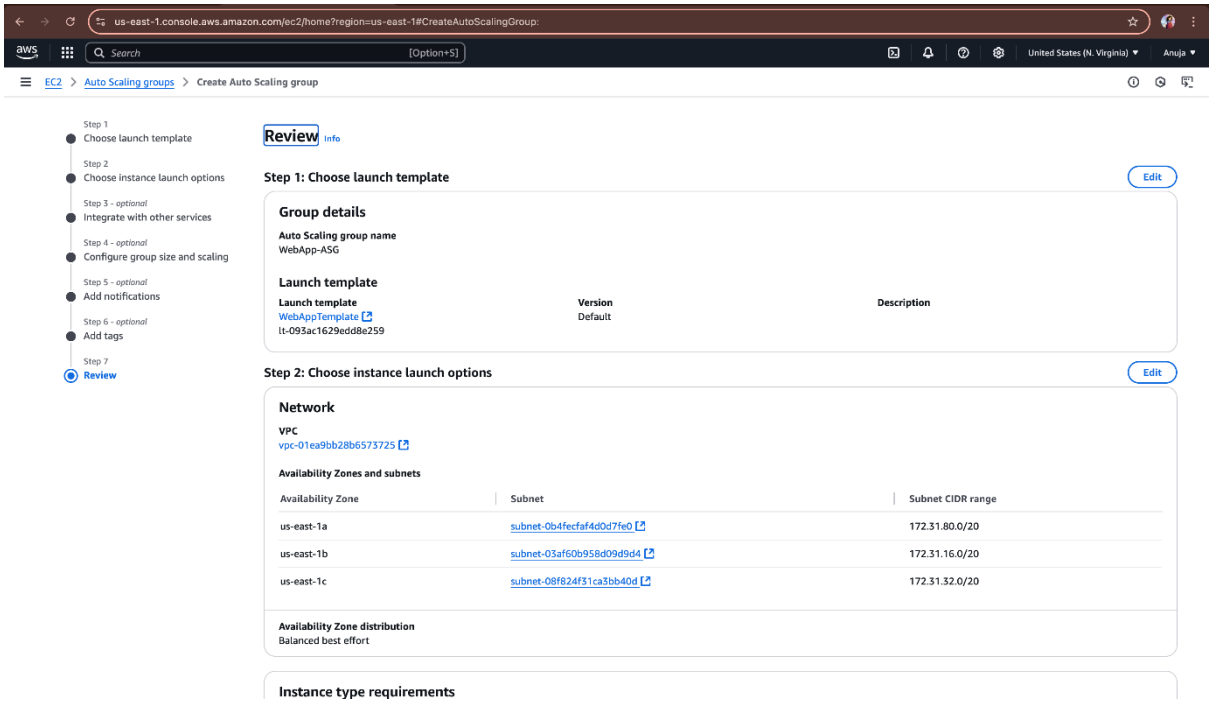
In Choose instance launch options select your VPC and select your subnets.

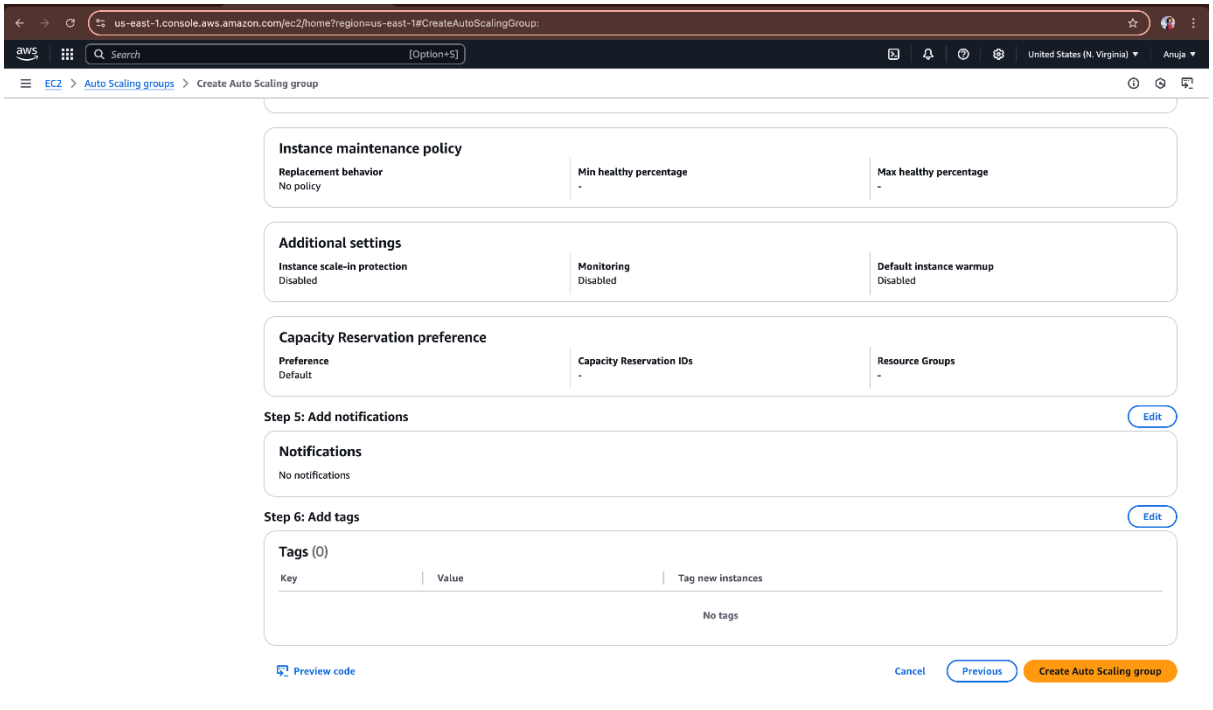


In configure group size and scaling Choose group size desire capacity as 1 and in scaling minimum desire capacity as 1 and maximum desire capacity as 5. And choose no scaling option we will do this later.

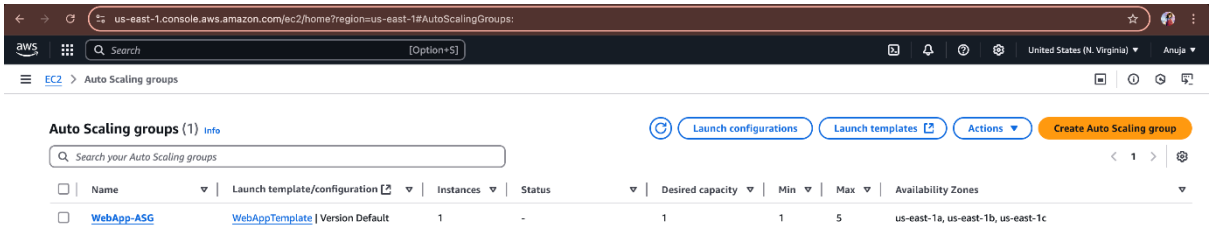


Review all your details and press create auto scaling group.





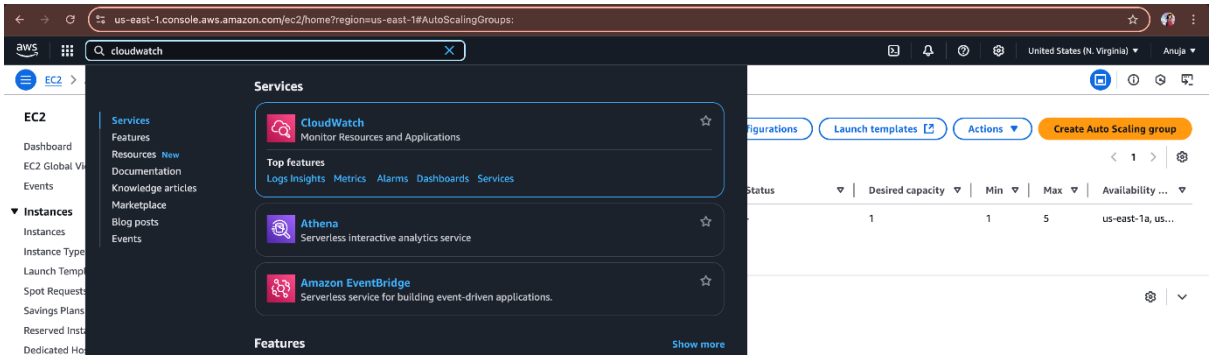
You can see your auto scaling group was created.

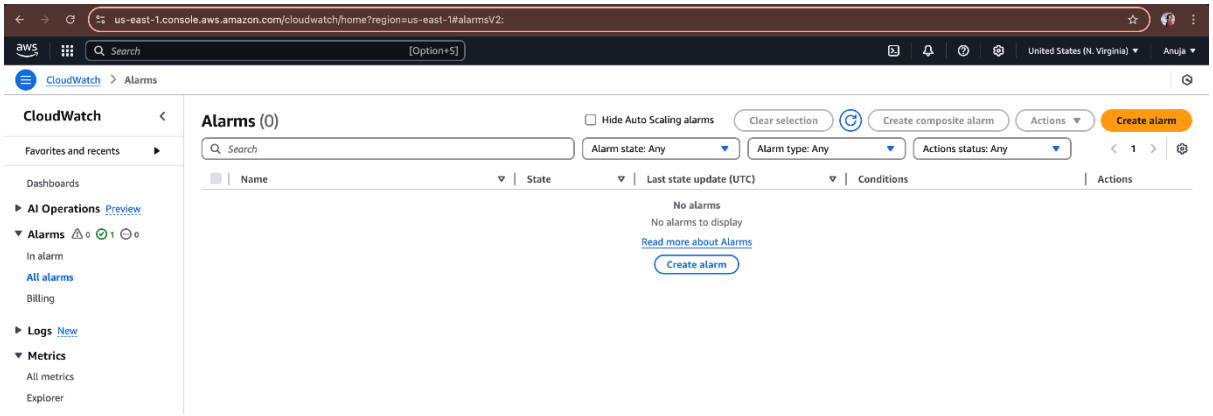


Step 7: Creating 2 alarms for scaling in and scaling out of instance in cloud watch.

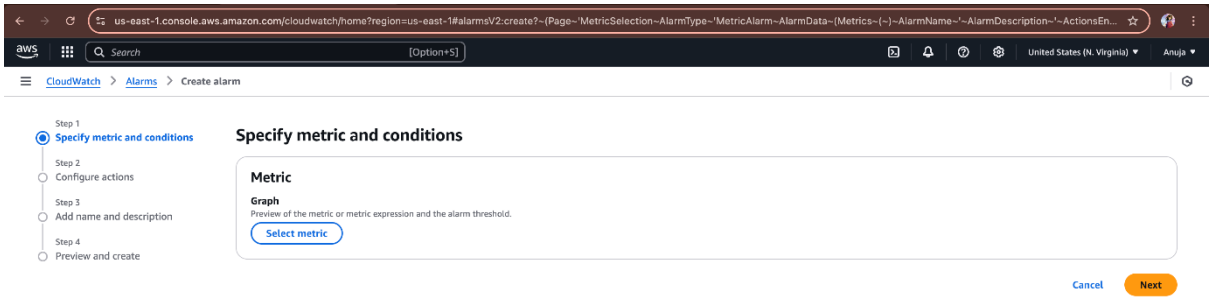
Alarm 1:

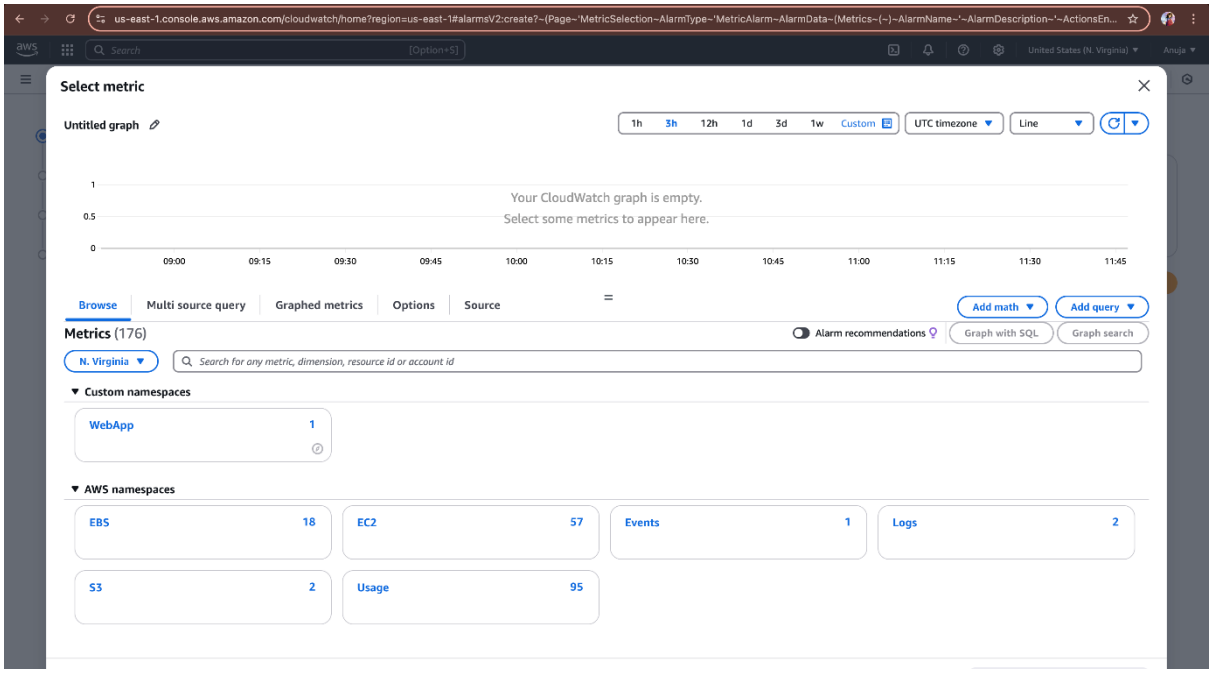
Go to cloud watch and click on create alarms.

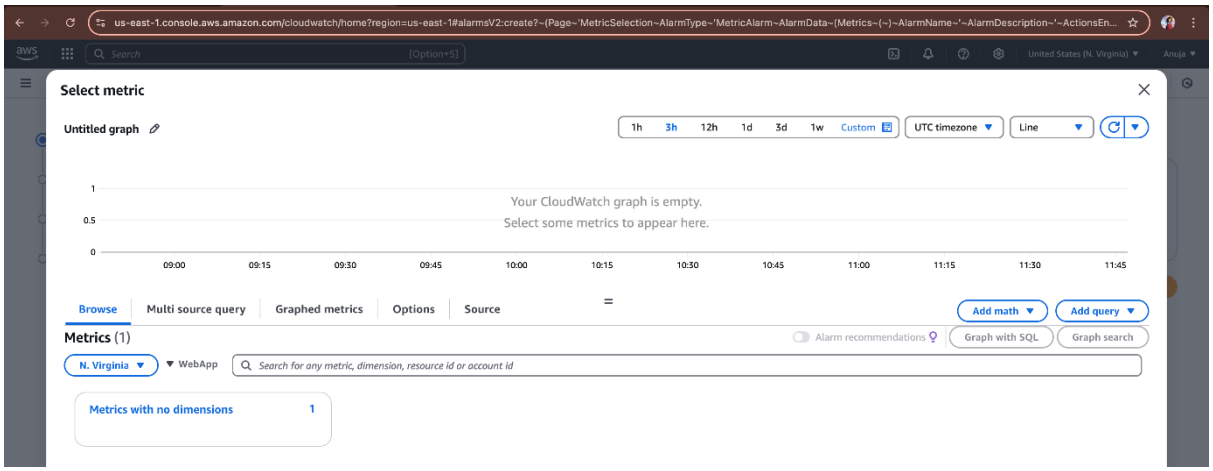


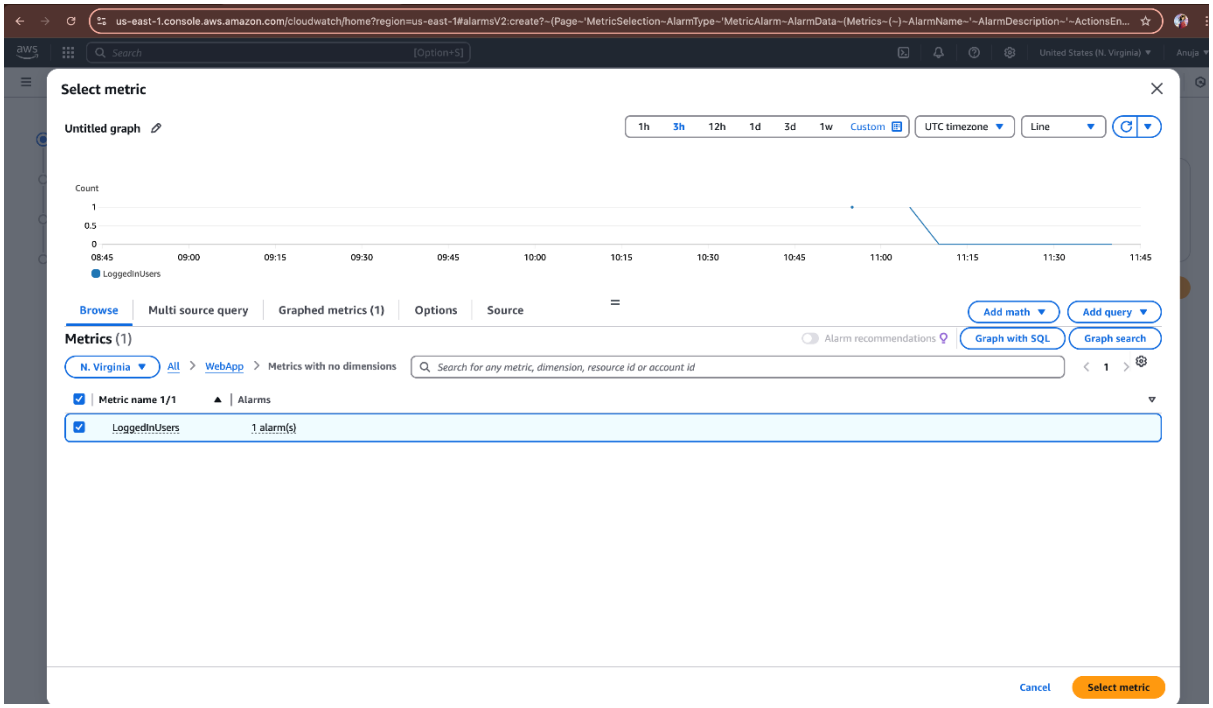


Now select your metrics which you created eailer.

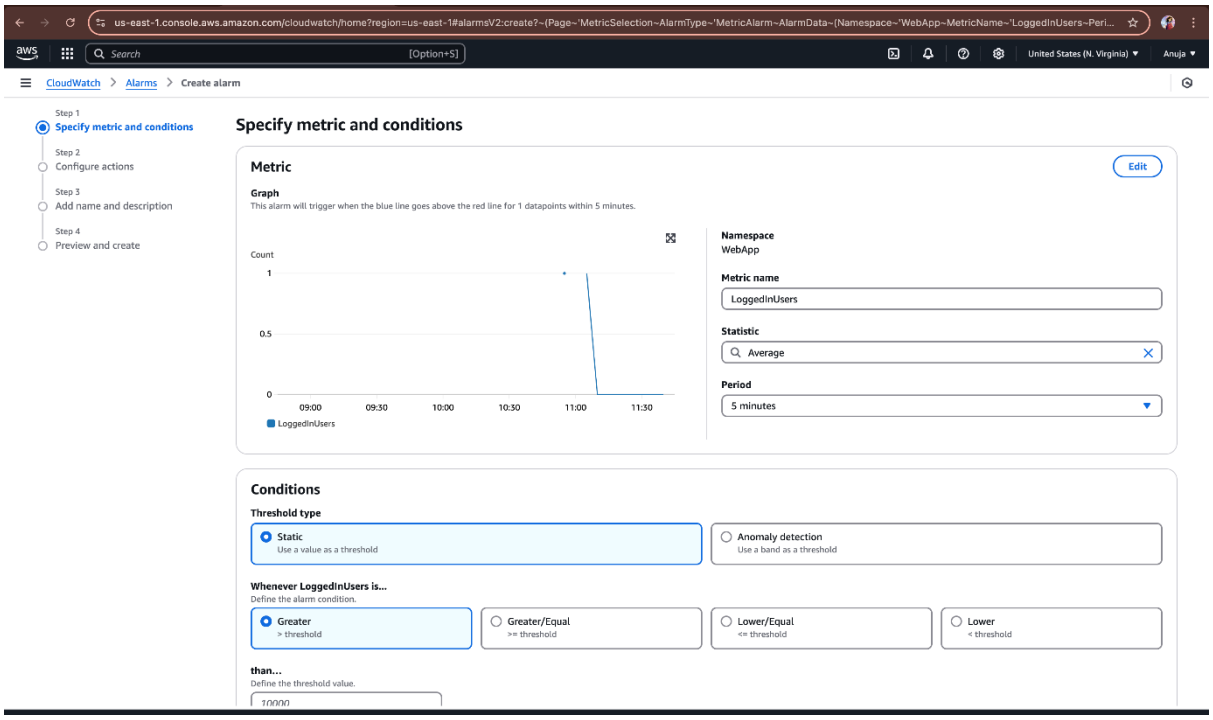

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~'MetricSelection~AlarmType~'MetricAlarm~AlarmData~(Metrics~(~)~AlarmName~'~AlarmDescription~'~ActionsEn...
Search
[Option+S]
United States (N. Virginia)
Anuja
CloudWatch
Alarms
Create alarm
Step 1
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create
Specify metric and conditions
Metric
Graph
Preview of the metric or metric expression and the alarm threshold.
Select metric
Cancel
Next


us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~'MetricSelection~AlarmType~'MetricAlarm~AlarmData~(Metrics~(~)~AlarmName~'~AlarmDescription~'~ActionsEn...
Select metric
Untitled graph
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Line
Your CloudWatch graph is empty.
Select some metrics to appear here.
09:00
09:15
09:30
09:45
10:00
10:15
10:30
10:45
11:00
11:15
11:30
11:45
Browse
Multi source query
Graphed metrics
Options
Source
Add math
Add query
Metrics (176)
Alarm recommendations
Graph with SQL
Graph search
N. Virginia
Search for any metric, dimension, resource id or account id
Custom namespaces
WebApp 1
AWS namespaces
EBS 18
EC2 57
Events 1
Logs 2
S3 2
Usage 95

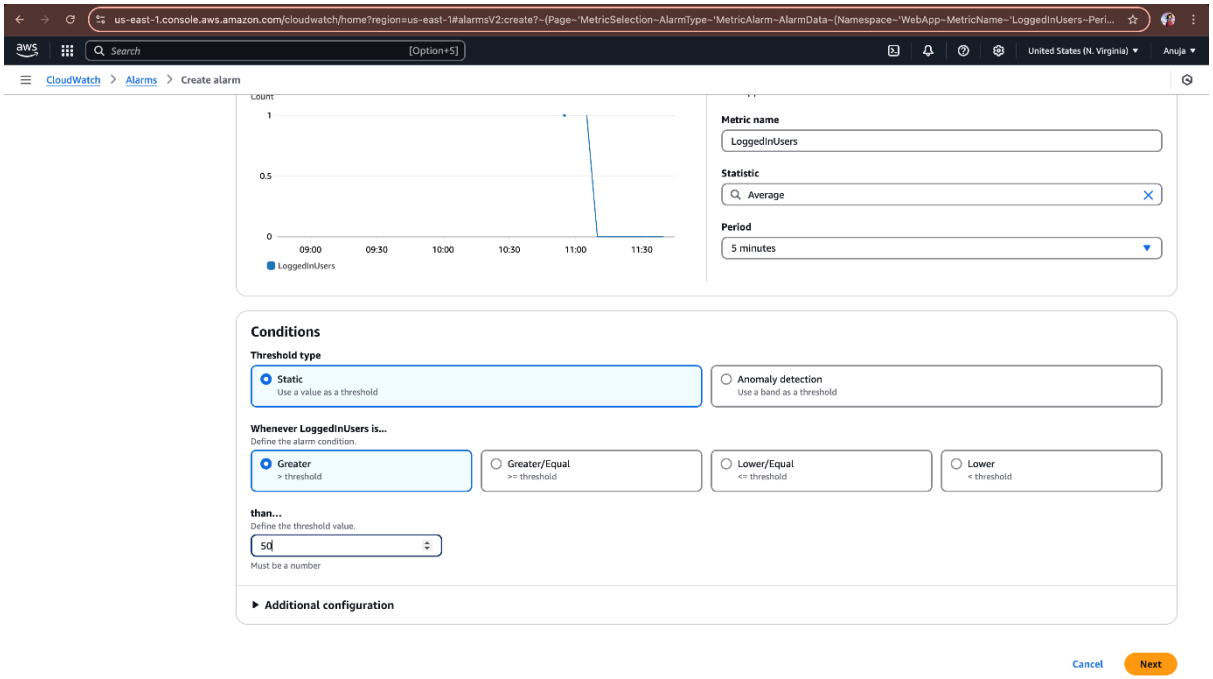

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~'MetricSelection~AlarmType~'MetricAlarm~AlarmData~(Metrics~(~)~AlarmName~'~AlarmDescription~'~ActionsEn...
Select metric
Untitled graph
1h
3h
12h
1d
3d
1w
Custom
UTC timezone
Line
Your CloudWatch graph is empty.
Select some metrics to appear here.
09:00
09:15
09:30
09:45
10:00
10:15
10:30
10:45
11:00
11:15
11:30
11:45
Browse
Multi source query
Graphed metrics
Options
Source
Add math
Add query
Metrics (1)
Alarm recommendations
Graph with SQL
Graph search
N. Virginia
WebApp
Search for any metric, dimension, resource id or account id
Metrics with no dimensions 1



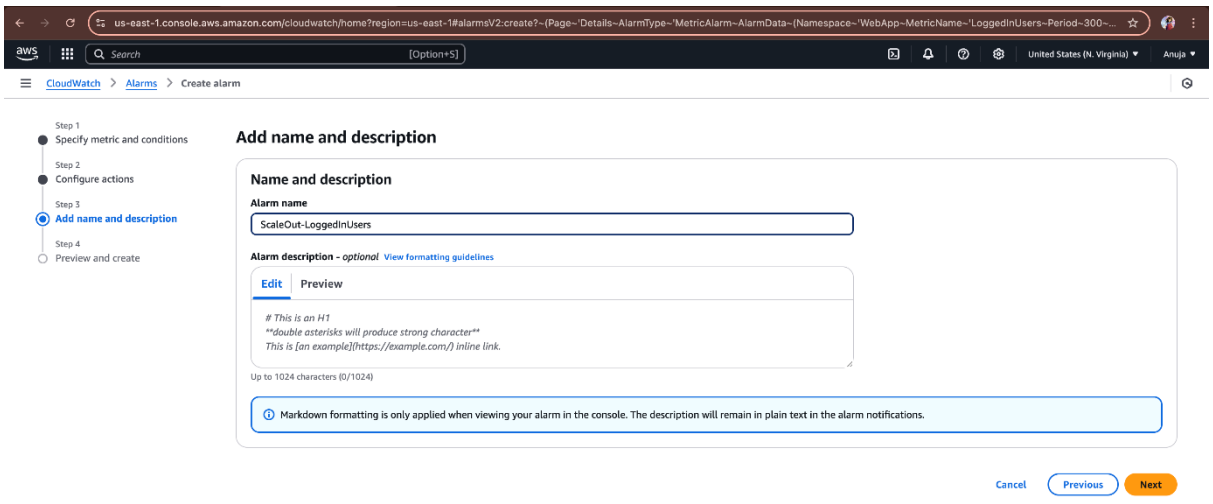
Then you can see the you have selected your desire metrices path.



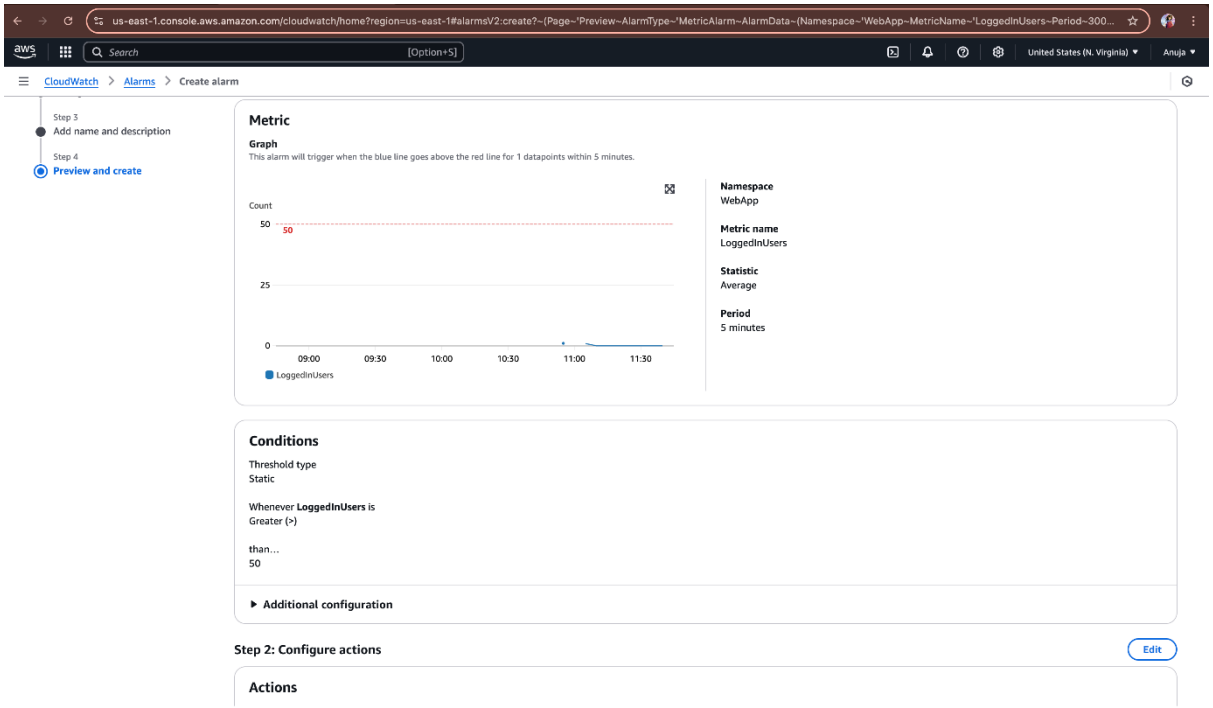
Then you need to give conditions thershold type as static and whenever logged in users is greater than 50.



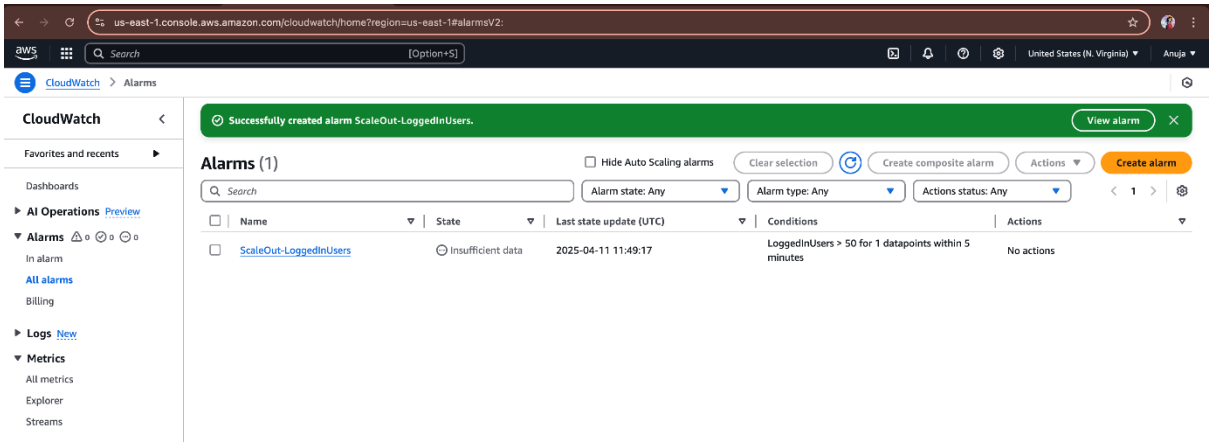
Add name as scaled out logged in users.



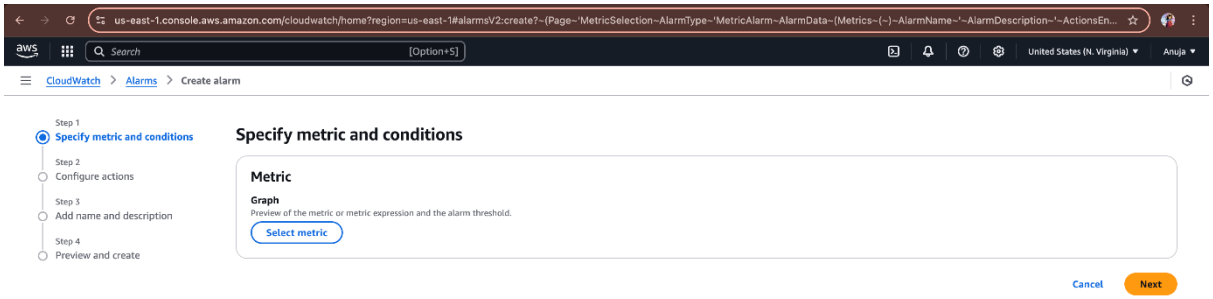
Then perview all your data and create the alaram.



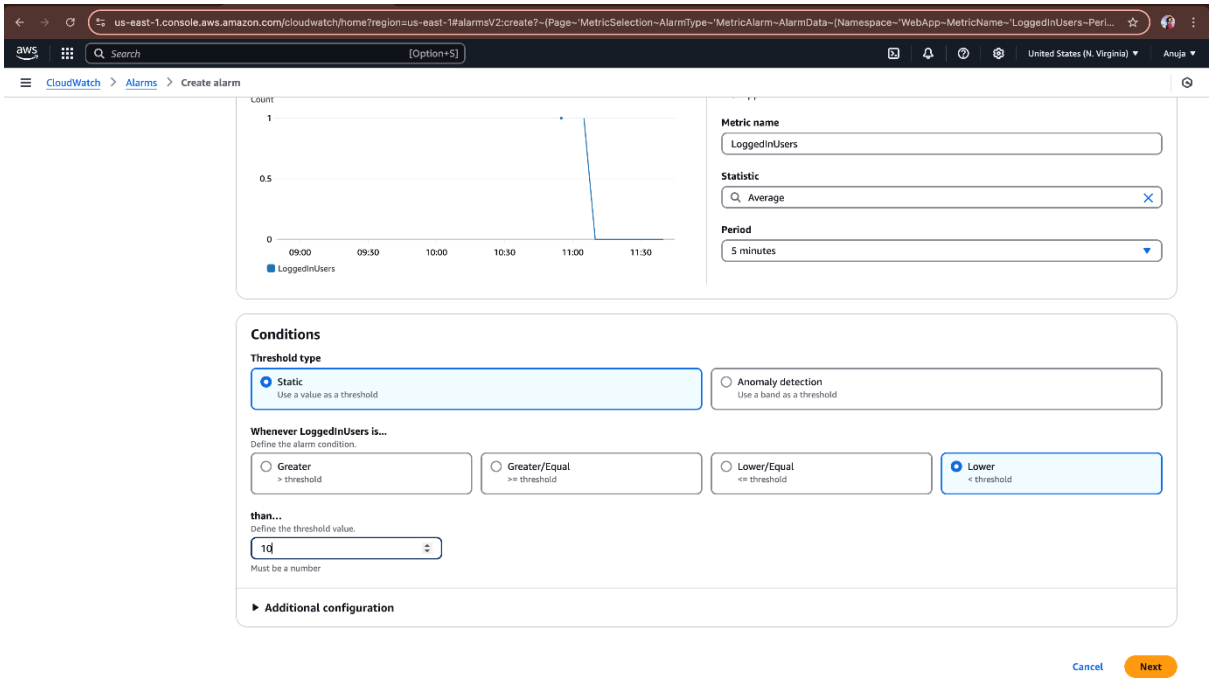
You can see youre alaram was created.



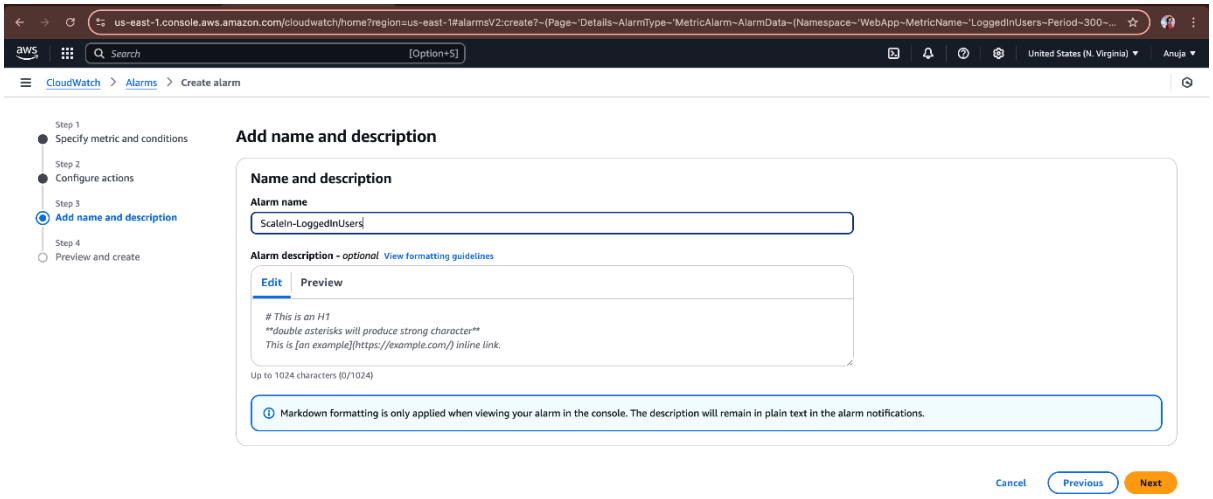
Same way go and create one more alaram.



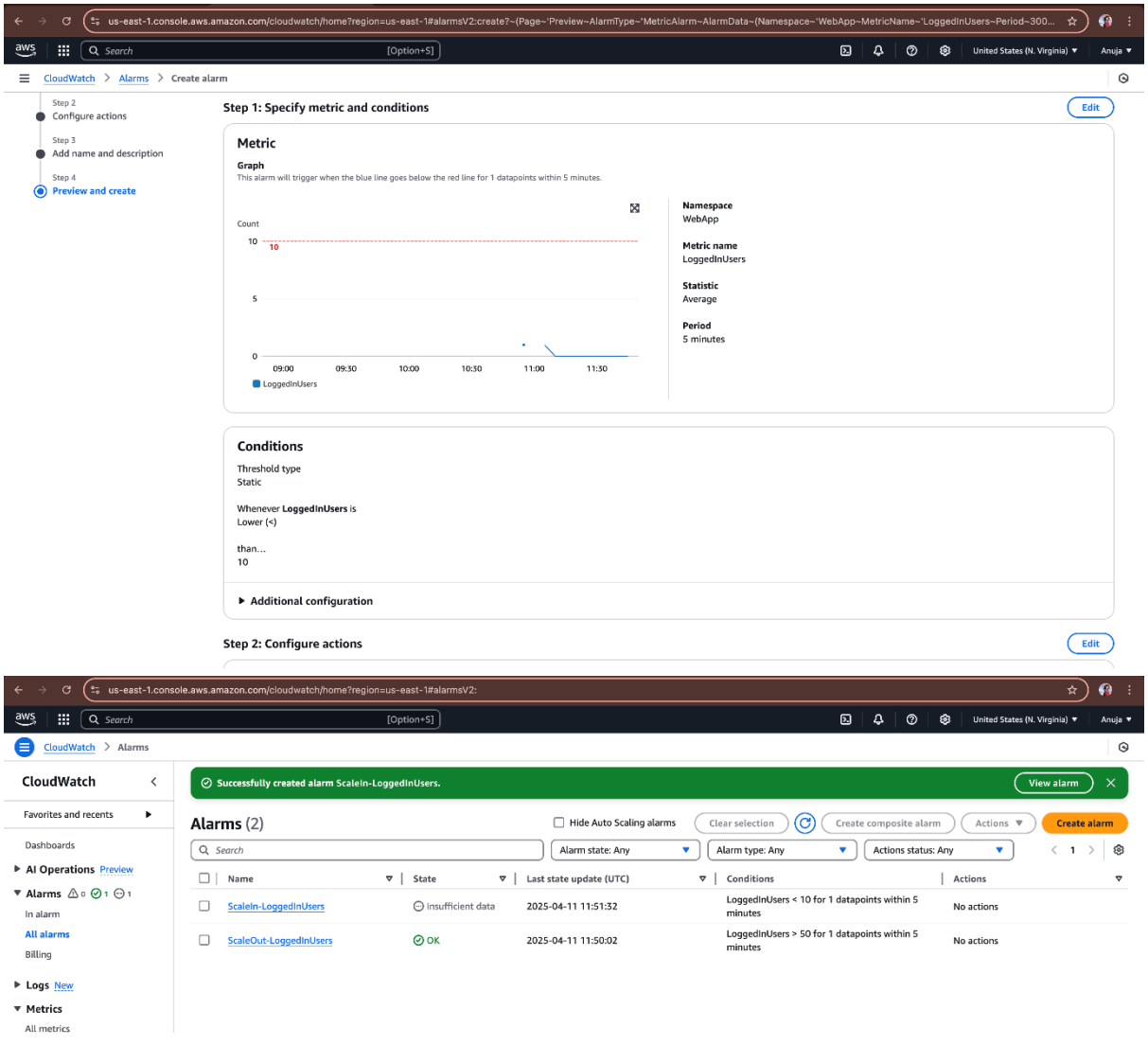
Then you need to give conditions thershold type as static and whenever logged in users is lower than 10.



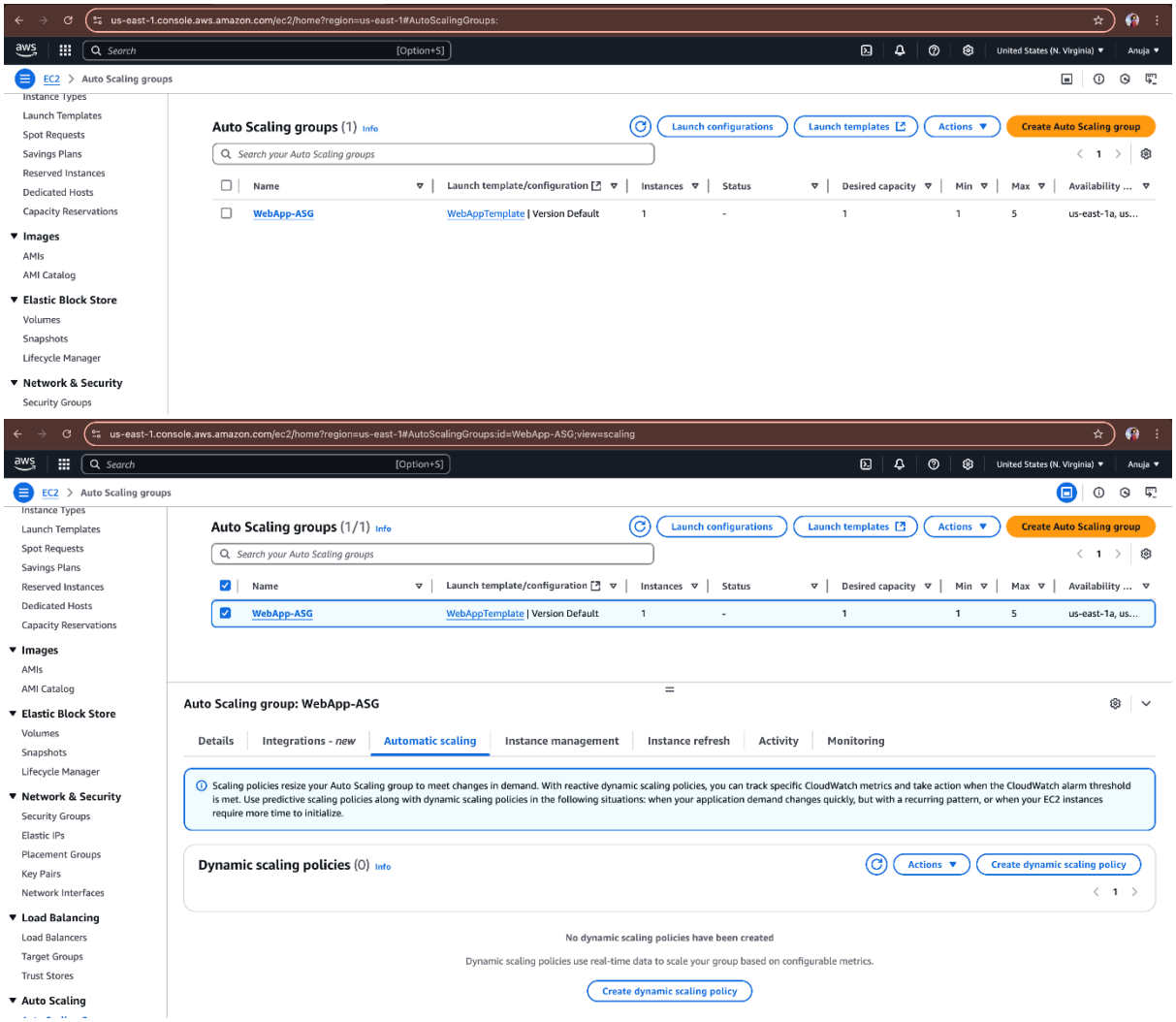
Give alaram name as scale in logged in users.



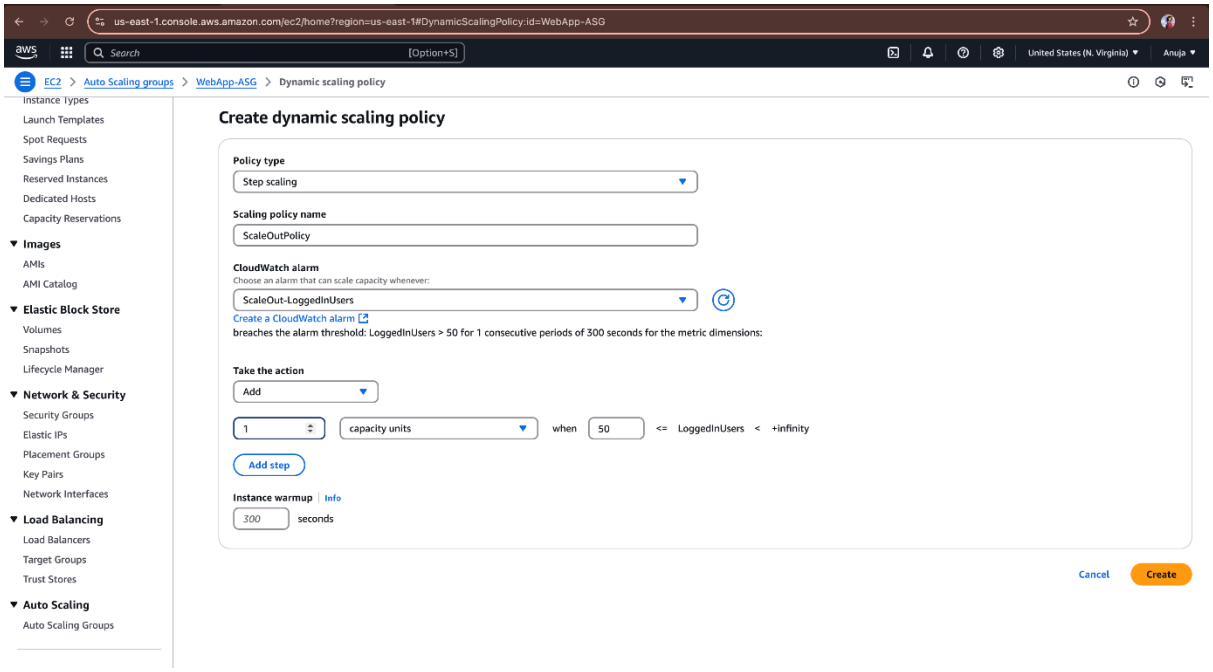
Preview all the details and create the second alaram.



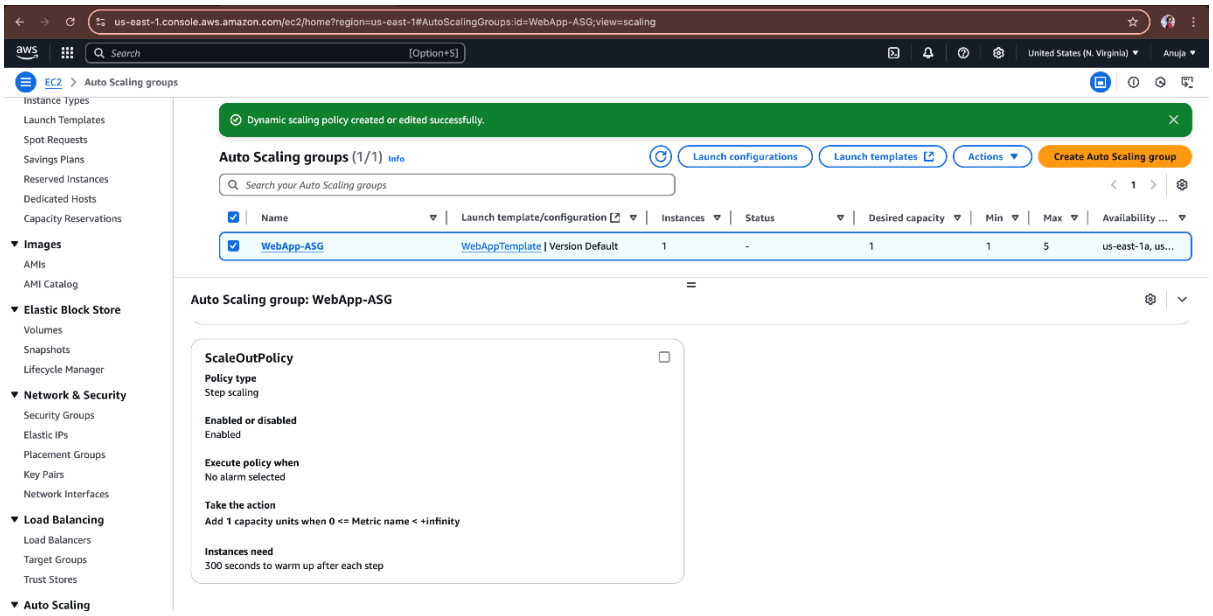
Step 8: Now both the alarams were created now we need to attach these alarams to Auto scaling group.
Go to auto scaling group then click on auto scaling.



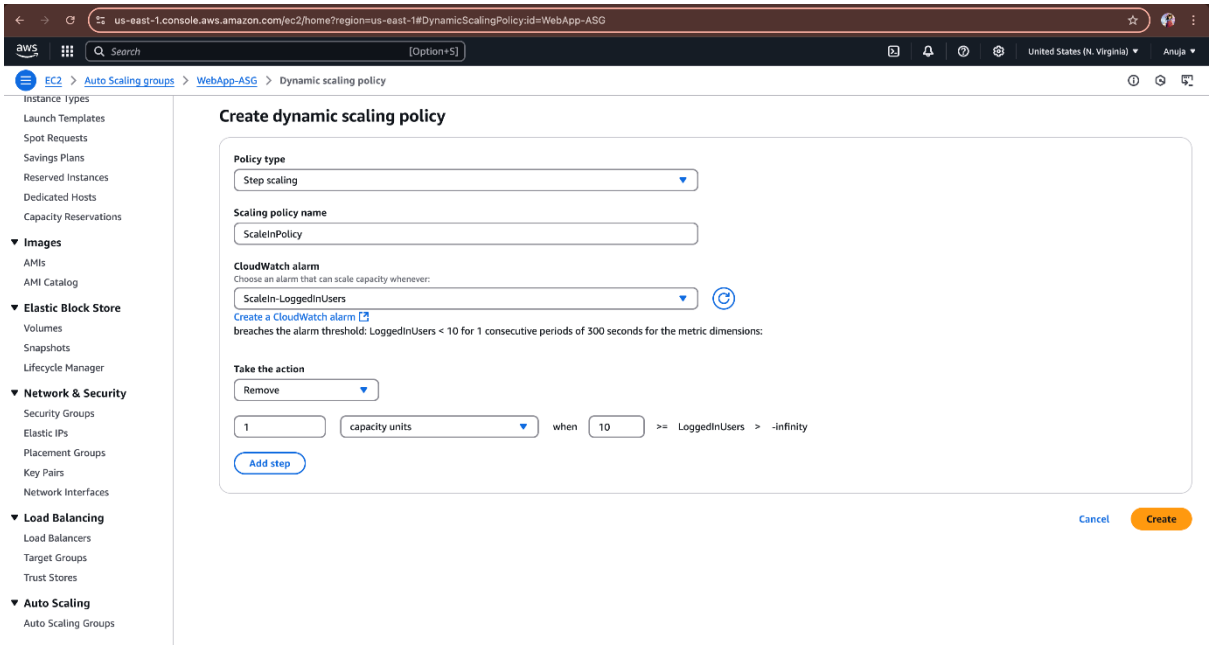
Create a dynamic scaling policy as shown In the below picutre with name scale out policy.



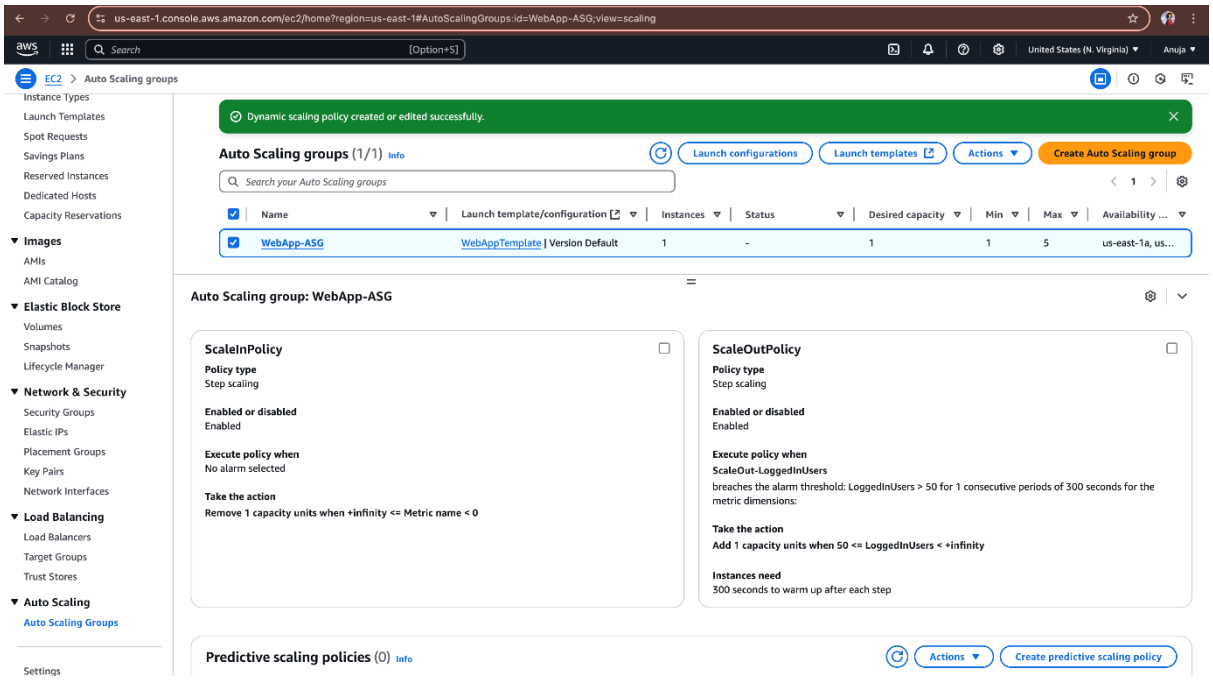
Click on create.



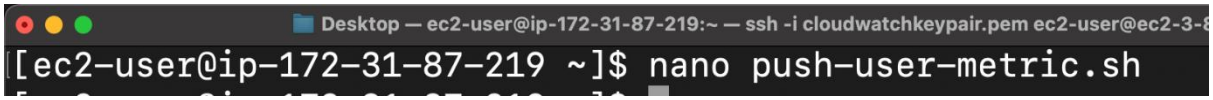
Create one more scaling policy named scale in policy as shown in below image.

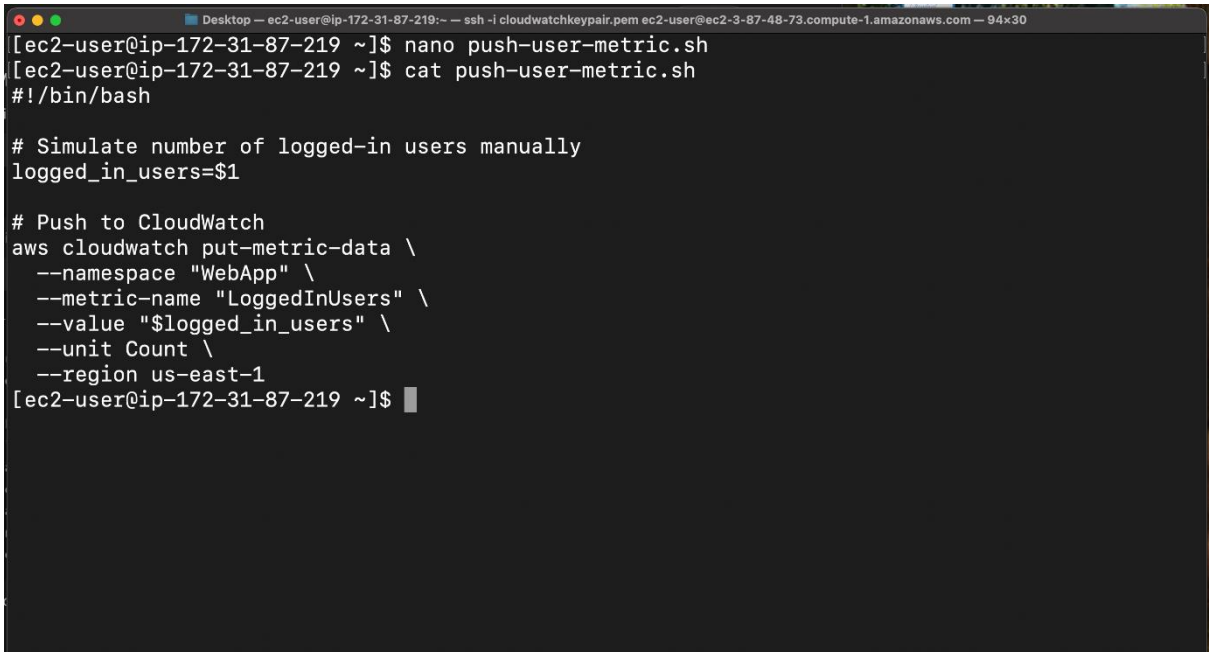


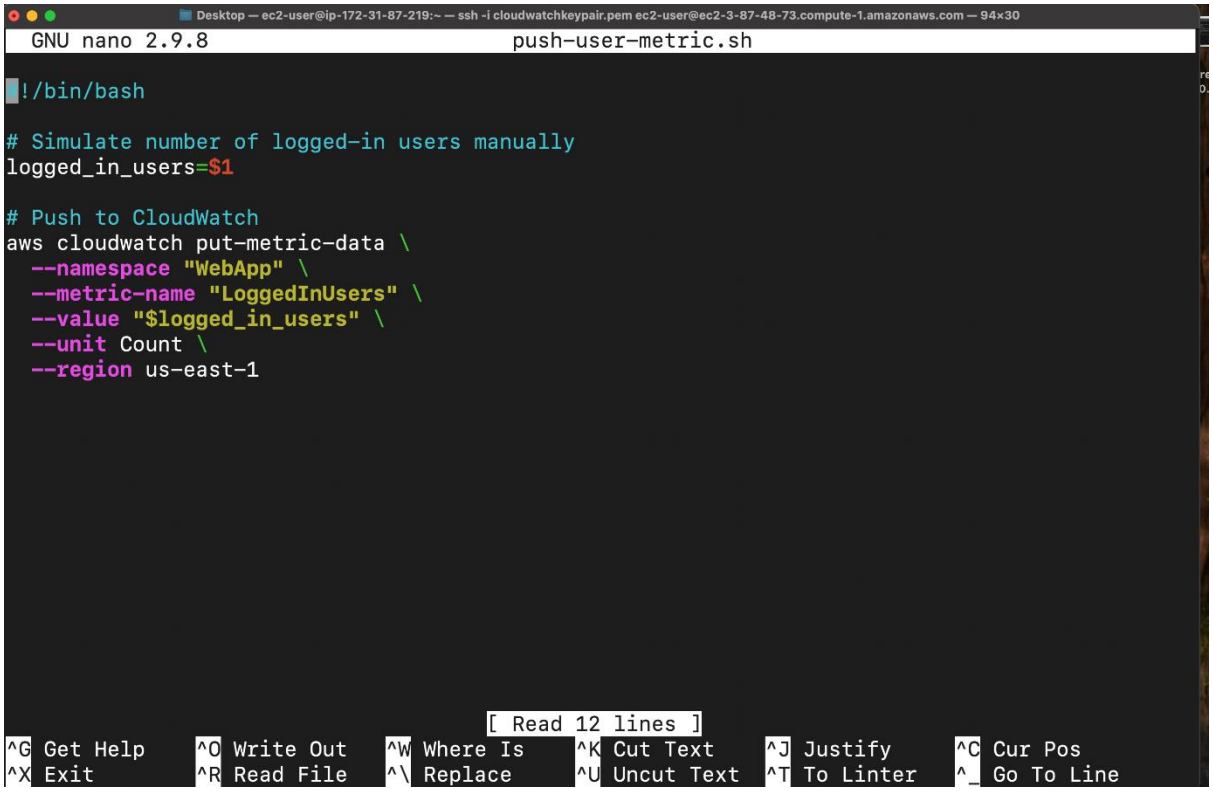
Now you can see you have attached 2 your 2 scaling polices.



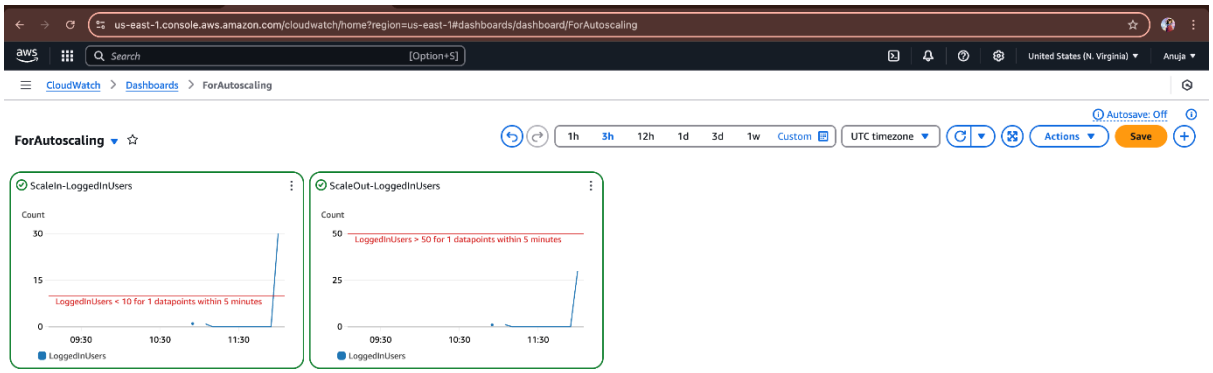
Final step: Now you can test your Auto scaling group with pushing the user metrices data and you can see status of your alarams in cloud watch dashboards.



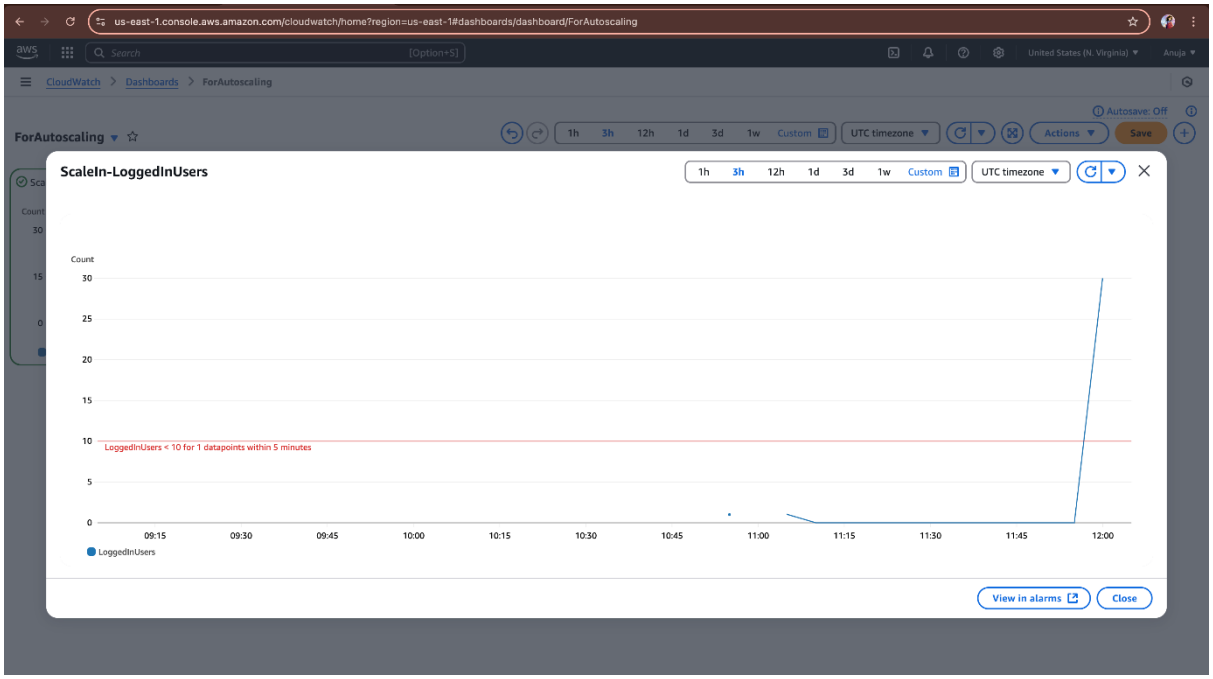


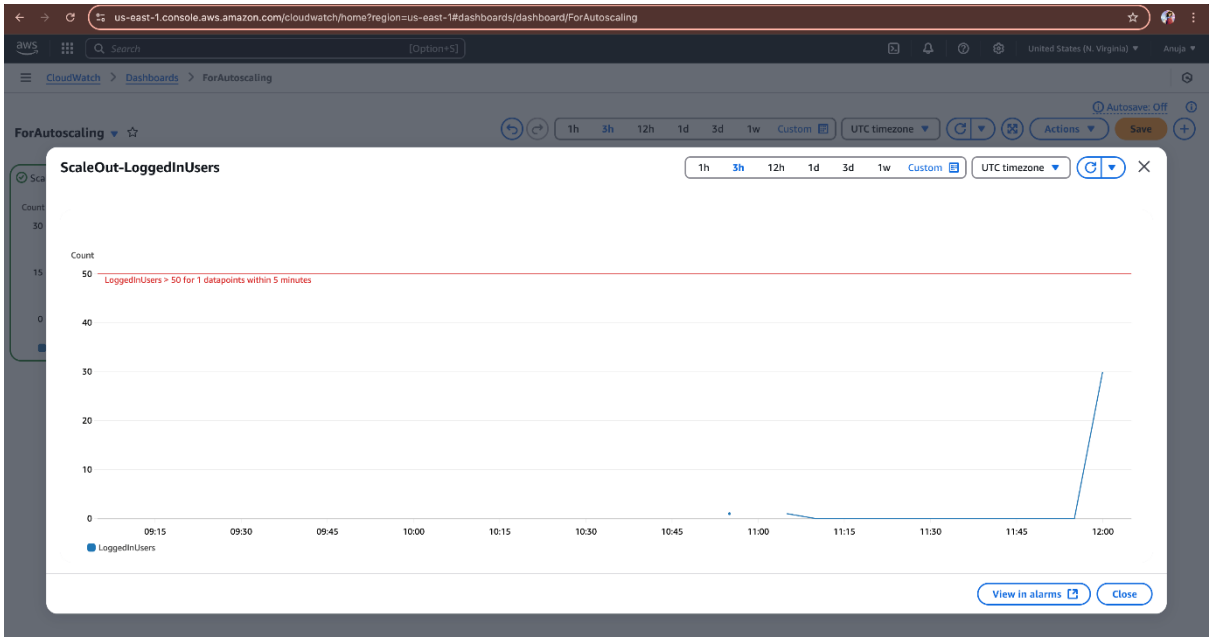


We have created dashboards for autoscaling group.



You can see your dashboards.





You can see one EC2 instance is automatically launched.