

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.

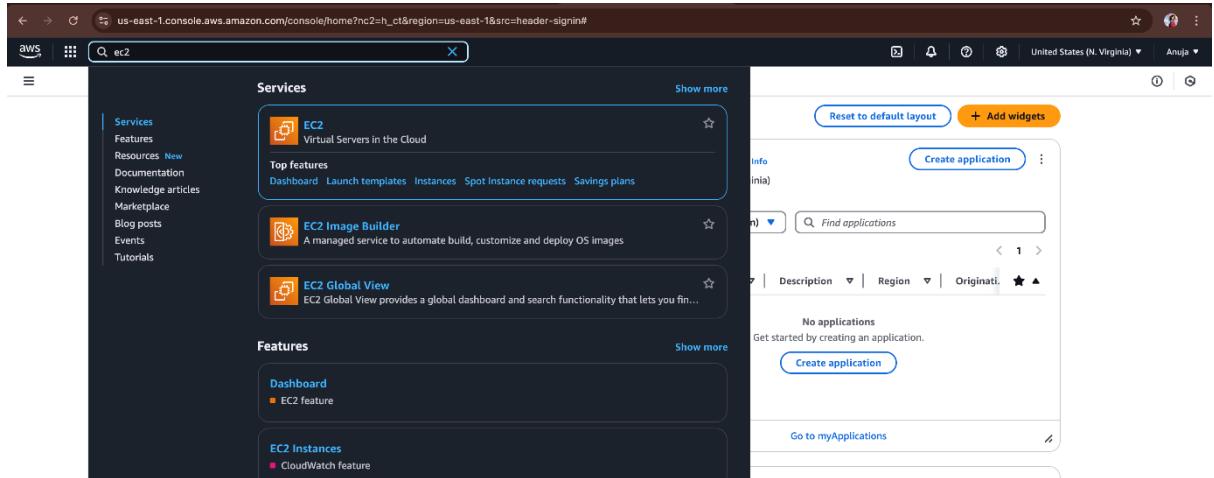
Tools required: AWS Services - CloudWatch, Auto Scaling, EC2

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.



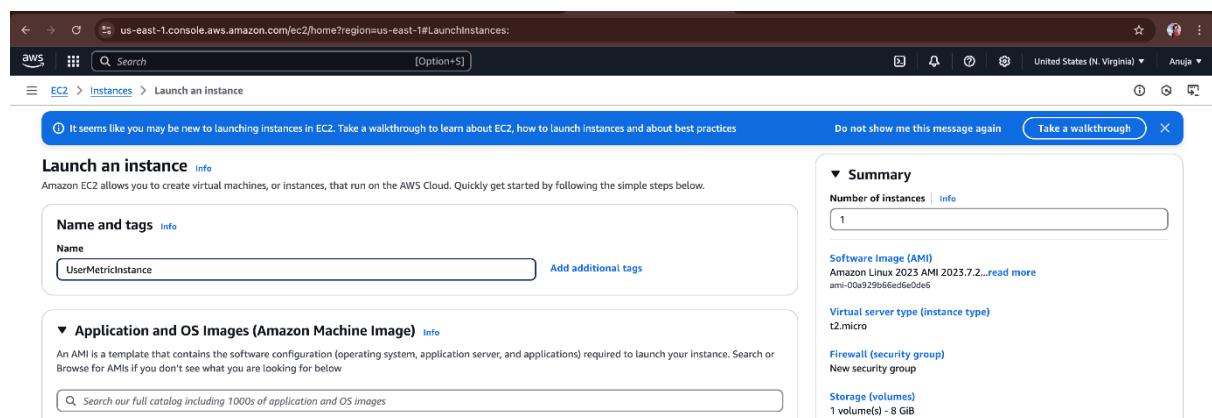
The screenshot shows the AWS console with the search bar set to 'ec2'. The main content area is titled 'Services' and lists the EC2 service. The EC2 card is highlighted, showing its description: 'Virtual Servers in the Cloud' and 'Top features: Dashboard, Launch templates, Instances, Spot Instance requests, Savings plans'. Below the EC2 card, there are cards for 'EC2 Image Builder' and 'EC2 Global View'. To the right, there is a 'Create application' section with a 'Find applications' search bar and a 'Create application' button. The top navigation bar shows the region as 'United States (N. Virginia)' and the user as 'Anuja'.

Click on launch instance.



The screenshot shows the EC2 home page. The main title is 'Amazon Elastic Compute Cloud (EC2)' with the subtext 'Create, manage, and monitor virtual servers in the cloud.' Below this, there is a section titled 'Benefits and features' with a sub-section 'EC2 offers ultimate scalability and control'. To the right, there is a 'Launch a virtual server' box with a 'Launch instance' button and a 'View dashboard' button. The left sidebar shows navigation links for EC2, Dashboard, EC2 Global View, Events, Instances, Images, and Elastic Block Store.

Give a name to your instance as **Usermetricsinstance**.



The screenshot shows the 'Launch an instance' wizard. The current step is 'Name and tags'. The 'Name' field is filled with 'UserMetricInstance'. The right side of the screen shows a 'Summary' section with 1 instance, using the 'Amazon Linux 2023 AMI 2023.7.2...' image, t2.micro instance type, and New security group. The storage is 1 volume(s) - 8 GiB. A message at the top left says 'It seems like you may be new to launching instances in EC2. Take a walkthrough to learn about EC2, how to launch instances and about best practices'.

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

Create key pair



Key pair name

Key pairs allow you to connect to your instance securely.

cloudwatchkeypair

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type

RSA

RSA encrypted private and public key pair

ED25519

ED25519 encrypted private and public key pair

Private key file format

.pem

For use with OpenSSH

.ppk

For use with PuTTY



When prompted, store the private key in a secure and accessible location on your computer. **You will need it later to connect to your instance.** [Learn more](#)

Cancel

Create key pair

And make all remaining things as default and launch instance.

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

We'll create a new security group called 'launch-wizard-1' with the following rules:

- Allow SSH traffic from Anywhere 0.0.0.0/0
- Allow HTTPS traffic from the internet To set up an endpoint, for example when creating a web server
- Allow HTTP traffic from the internet To set up an endpoint, for example when creating a web server

⚠ Rules of source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Configure storage Info
Advanced

1x 8 GiB gp2 Root volume, Not encrypted

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Add new volume

Click refresh to view backup information The tags that you assign determine whether the instance will be backed up by any Data Lifecycle Manager policies.

0 x File systems

Advanced details Info

Summary
Number of instances Info
1

Software Image (AMI)
Amazon Linux 2 Kernel 5.10 AMI... read more

Virtual server type (Instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

ⓘ Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Cancel Launch instance Preview code

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

Instances (1) Info
Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
<input type="checkbox"/>	UserMetricInst...	i-067bdd8dd4ad16dd	Running	t2.micro	Initializing	View alarms +	us-east-1a	ec2-3-87-48-73.comput...	3.87.48.73

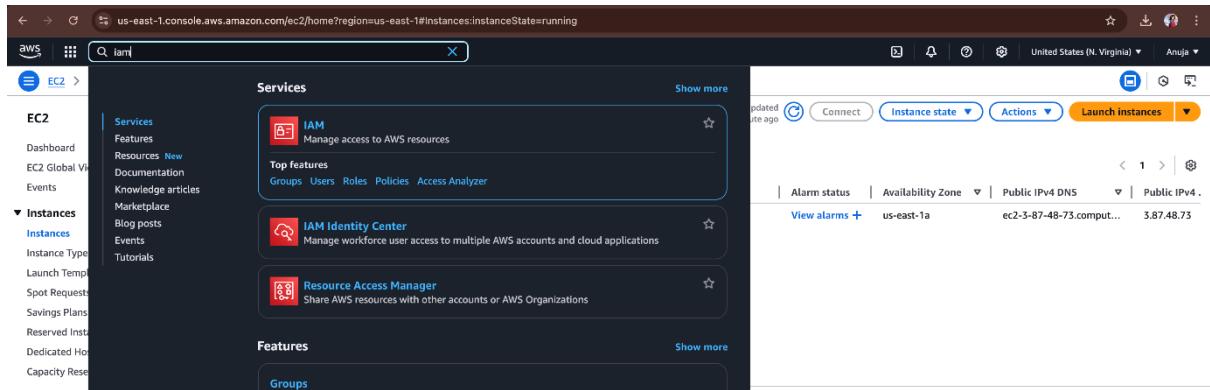
Instances (1/1) Info
Find Instance by attribute or tag (case-sensitive) All states

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4 DNS Public IPv4

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
<input checked="" type="checkbox"/>	UserMetricInst...	i-067bdd8dd4ad16dd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1a	ec2-3-87-48-73.comput...	3.87.48.73

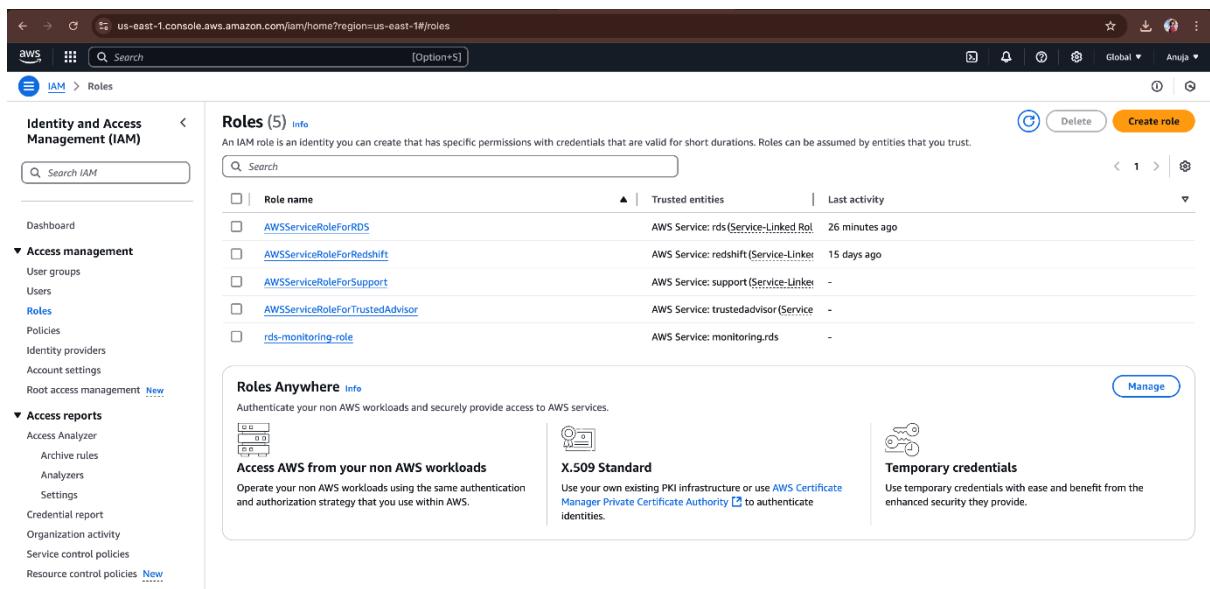
Step2: Creating IAM role and attaching to EC2 Instance.

Go to dashboard and click on IAM.



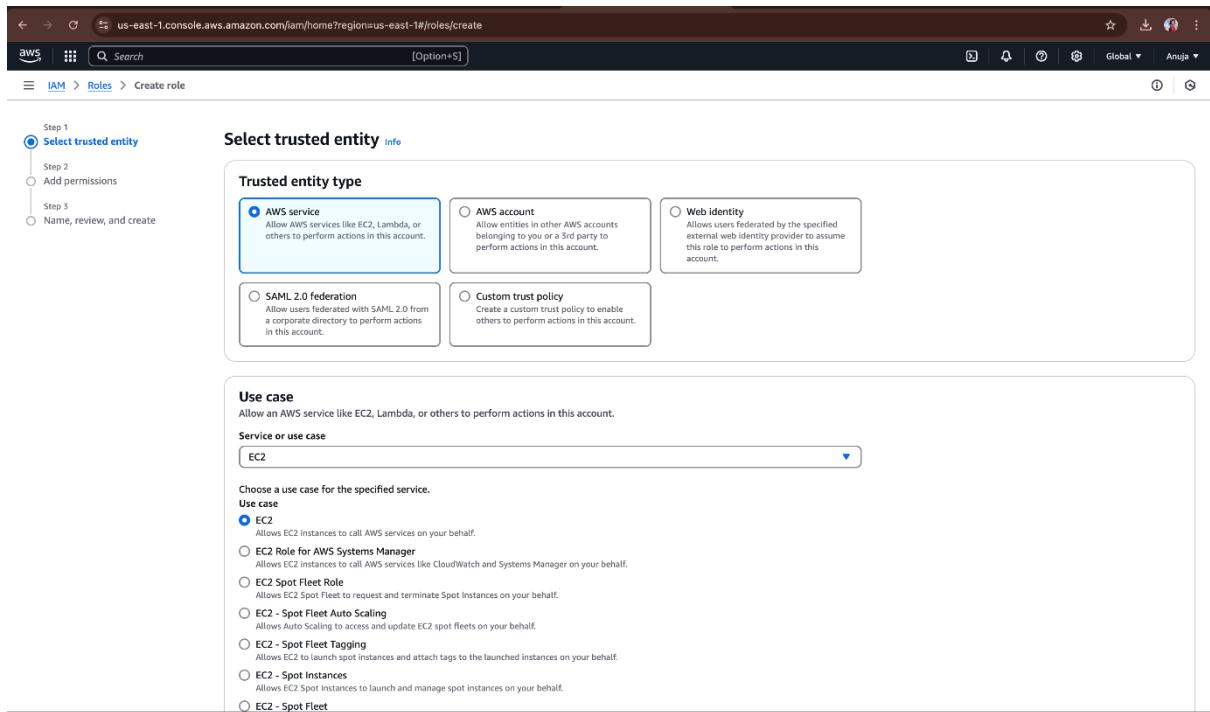
The screenshot shows the AWS EC2 Instances page. A search bar at the top has 'iam' typed into it. On the left, a sidebar for 'EC2' shows 'Instances' selected, with sub-options like 'Launch Temp' and 'Reserved Inst'. On the right, an instance named 'ec2-3-87-48-73.compute...' is listed with a Public IPv4 of '3.87.48.73'.

click on create role.



The screenshot shows the AWS IAM Roles page. A search bar at the top has '(Option+S)' typed into it. The left sidebar shows 'Access management' selected, with 'Roles' highlighted. The main area displays a table of roles, including 'AWSRoleForRDS', 'AWSRoleForRedshift', 'AWSRoleForSupport', 'AWSRoleForTrustedAdvisor', and 'rds-monitoring-role'. A 'Create role' button is located at the top right of the role list.

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



Step 1
 Select trusted entity
 Step 2
 Step 3
 Name, review, and create

Select trusted entity info

Trusted entity type

AWS service
Allow AWS services like EC2, Lambda, or others to perform actions in this account.

AWS account
Allow entities in other AWS accounts belonging to you or a 3rd party to perform actions in this account.

Web identity
Allow users federated by the specified external web identity provider to assume this role to perform actions in this account.

SAML 2.0 federation
Allow users federated with SAML 2.0 from a corporate directory to perform actions in this account.

Custom trust policy
Create a custom trust policy to enable others to perform actions in this account.

Use case
Allow an AWS service like EC2, Lambda, or others to perform actions in this account.

Service or use case
EC2

Choose a use case for the specified service.

Use case

EC2
Allows EC2 instances to call AWS services on your behalf.

EC2 Role for AWS Systems Manager
Allows EC2 instances to call AWS services like CloudWatch and Systems Manager on your behalf.

EC2 Spot Fleet Role
Allows EC2 Spot Fleet to request and terminate Spot Instances on your behalf.

EC2 - Spot Fleet Auto Scaling
Allows Auto Scaling to access and update EC2 spot fleets on your behalf.

EC2 - Spot Fleet Tagging
Allows EC2 to launch spot instances and attach tags to the launched instances on your behalf.

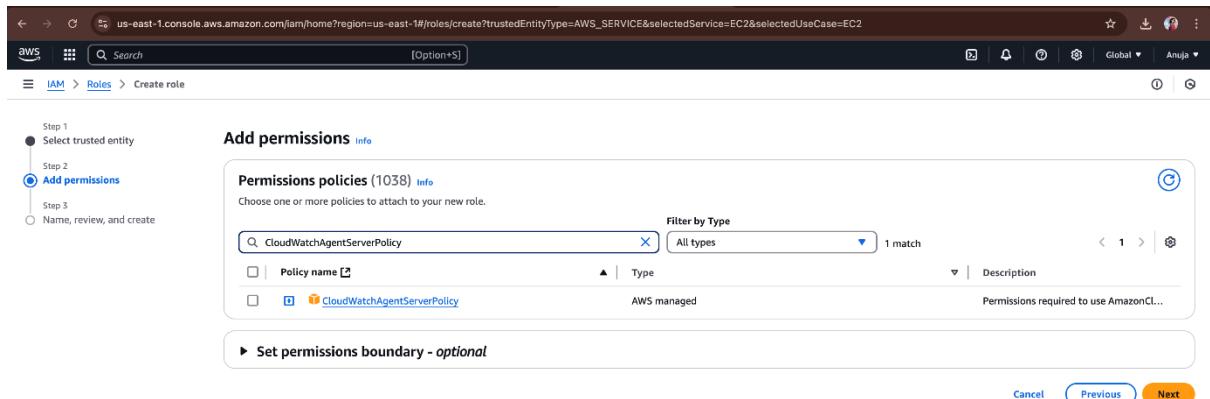
EC2 - Spot Instances
Allows EC2 Spot Instances to launch and manage spot instances on your behalf.

EC2 - Spot Fleet

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

CloudWatchAgentServerPolicy

AmazonEc2RoleForSSM



Step 1
 Select trusted entity
 Add permissions
 Step 3
 Name, review, and create

Add permissions info

Permissions policies (1038) info

Choose one or more policies to attach to your new role.

Filter by Type
CloudWatchAgentServerPolicy All types 1 match

Policy name Type Description

CloudWatchAgentServerPolicy AWS managed Permissions required to use AmazonCl...

Set permissions boundary - *optional*

Cancel Previous Next

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Add permissions Info

Permissions policies (2/1038) Info

Choose one or more policies to attach to your new role.

Filter by Type All types 1 match

Policy name	Type	Description
AmazonEC2RoleforSSM	AWS managed	This policy will soon be deprecated. Pl...

Set permissions boundary - optional

Cancel Previous Next

Review all the sections and click on create role.

Step 1
Select trusted entity
Step 2
Add permissions
Step 3
Name, review, and create

Name, review, and create

Role details

Role name
Enter a meaningful name to identify this role.
EC2CloudWatchRole

Description
Add a short explanation for this role.
Allows EC2 Instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

```
1- {
2-   "Version": "2012-10-17",
3-   "Statement": [
4-     {
5-       "Effect": "Allow",
6-       "Action": [
7-         "sts:AssumeRole"
8-       ],
9-       "Principal": [
10-         {
11-           "Service": [
12-             "ec2.amazonaws.com"
13-           ]
14-         }
15-       ]
16-     }
]
```

Step 2: Add permissions

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonEC2RoleforSSM	AWS managed	Permissions policy
CloudWatchAgentServerPolicy	AWS managed	Permissions policy

Step 3: Add tags

Add tags - optional

No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel Previous Create role

You Can see your role was creating.

Role EC2CloudWatchRole created.

Roles (1/6)

Role name	Trusted entities	Last activity
AWSServiceRoleForRDS	AWS Service: rds (Service-Linked Role)	29 minutes ago
AWSServiceRoleForRedshift	AWS Service: redshift (Service-Linked Role)	15 days ago
AWSServiceRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSServiceRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
<input checked="" type="checkbox"/> EC2CloudWatchRole	AWS Service: ec2	-
rds-monitoring-role	AWS Service: monitoring.rds	-

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.

Instances (1/1)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
UserMetricInst...	i-067bdd8dd4ad16dd	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a

Actions

Launch instance

Connect

Instance state

Last updated less than a minute ago

Actions

Modify IAM role

Connect

View details

Manage instance state

Instance settings

Networking

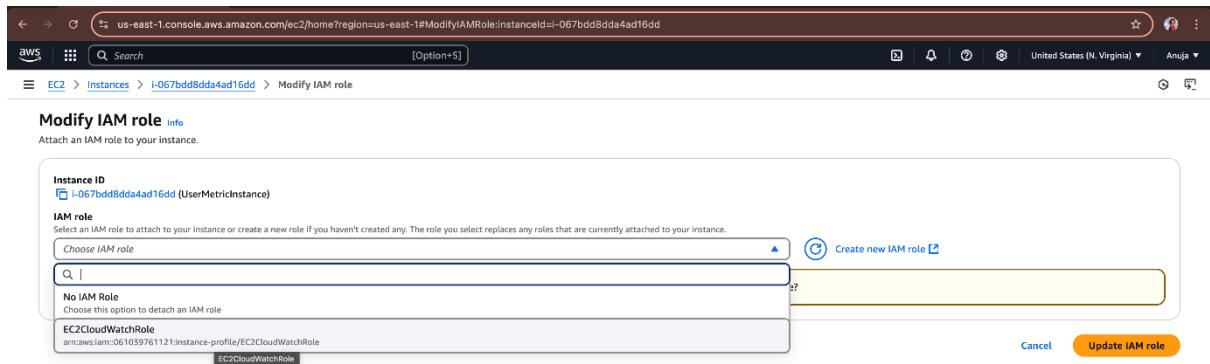
Security

Get Windows password

Image and templates

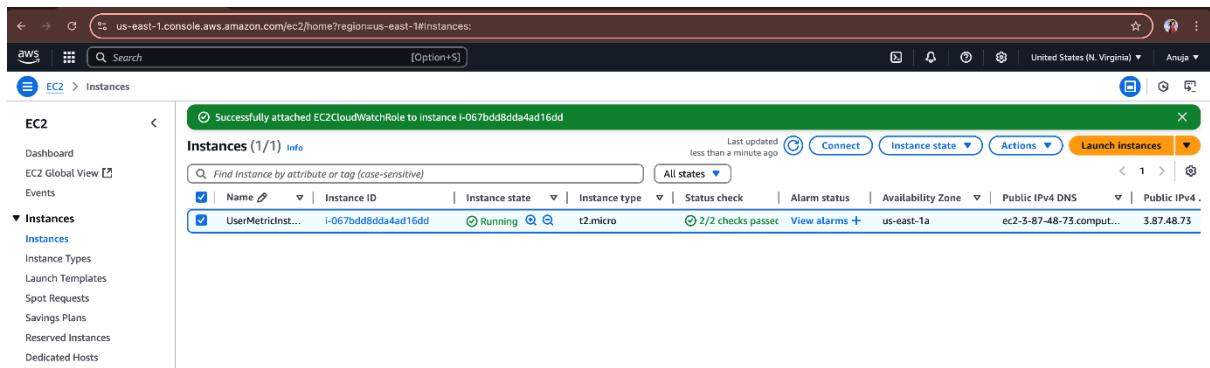
Monitor and troubleshoot

Choose your newly created IAM role and attach.



The screenshot shows the 'Modify IAM role' page in the AWS IAM console. The instance ID is I-067bdd8dda4ad16dd. The 'IAM role' section shows a dropdown menu with 'EC2CloudWatchRole' selected. The 'Update IAM role' button is visible at the bottom right.

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

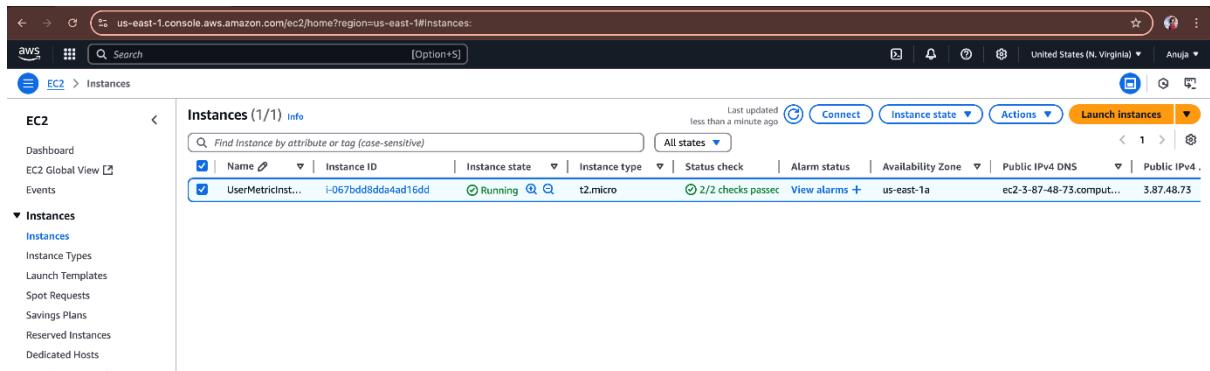


The screenshot shows the 'Instances' page in the AWS EC2 console. A green success message at the top states 'Successfully attached EC2CloudWatchRole to instance I-067bdd8dda4ad16dd'. The table below shows the instance details, including the attached role.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
UserMetricInst...	I-067bdd8dda4ad16dd	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-5-87-48-73.comput...	3.87.48.73

Step3: Pushing custom metrics to cloud watch

Go to Ec2 and connect to ssh.



The screenshot shows the 'Instances' page in the AWS EC2 console. The instance details table is identical to the previous screenshot, showing the attached EC2CloudWatchRole and its status.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IPv4
UserMetricInst...	I-067bdd8dda4ad16dd	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	ec2-5-87-48-73.comput...	3.87.48.73

Connect to your EC2 instance using terminal.

```
raj@Rajs-Laptop ~ % cd desktop
raj@Rajs-Laptop desktop % chmod 400 "cloudwatchkeypair.pem"
raj@Rajs-Laptop desktop %

[ec2-user@ip-172-31-87-219 ~]$ ssh -i "cloudwatchkeypair.pem" ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com
raj@Rajs-Laptop desktop % ssh -i "cloudwatchkeypair.pem" ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com
The authenticity of host 'ec2-3-87-48-73.compute-1.amazonaws.com (3.87.48.73)' can't be established.
ED25519 key fingerprint is SHA256:Or1eRqpnY1RzH8gwR0rW+UT282eFn3dDNbkqoscp5N0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-87-48-73.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

          #
          #_
          ~\_ #####_      Amazon Linux 2
          ~\_#####\_
          ~~ \###|      AL2 End of Life is 2026-06-30.
          ~~ \#/      ~~>
          ~~ V~'      /-->
          ~~      /      A newer version of Amazon Linux is available!
          ~~-. /      /-->
          ~~ /_ /      Amazon Linux 2023, GA and supported until 2028-03-15.
          _/m/      https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-87-219 ~]$
```

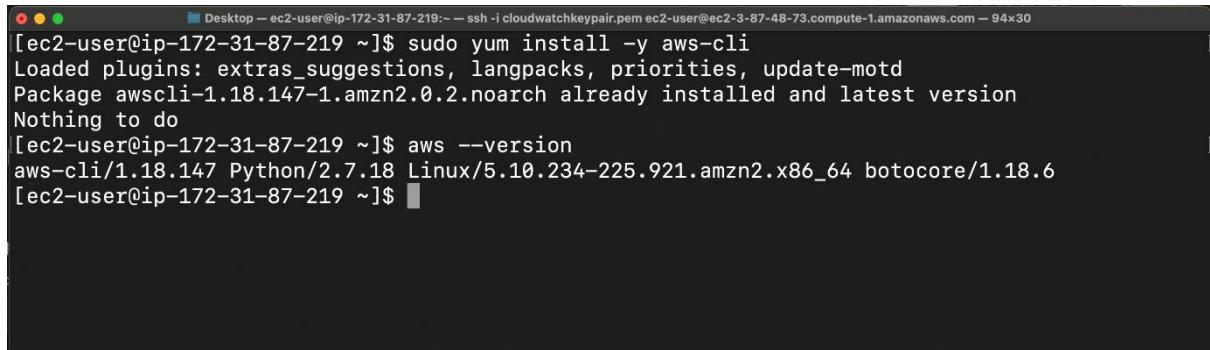
Then install all the updates using the command `yum update`.

```
[ec2-user@ip-172-31-87-219 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
[ec2-user@ip-172-31-87-219 ~]$
```

Then now need to install AWS-Cli.

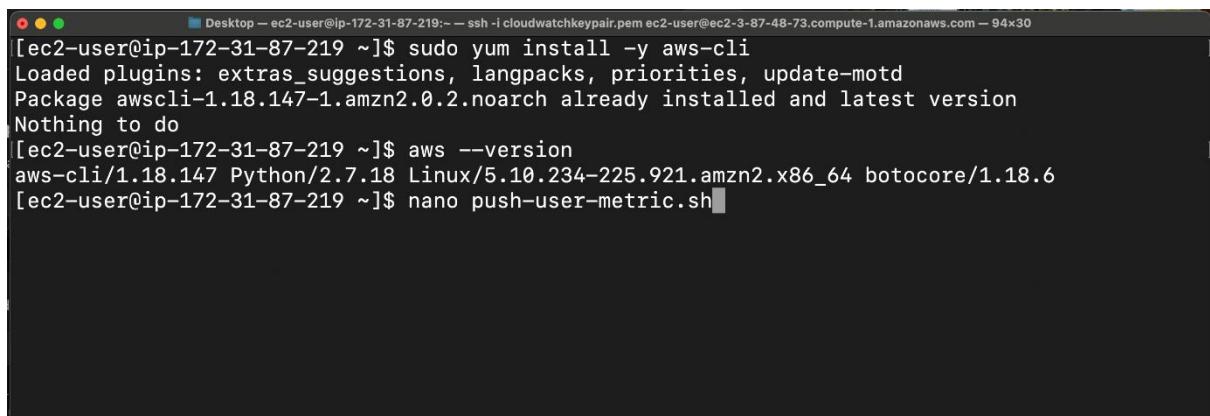
```
[ec2-user@ip-172-31-87-219 ~]$ sudo yum install -y aws-cli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package awscli-1.18.147-1.amzn2.0.2.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-219 ~]$
```

Check AWS version.



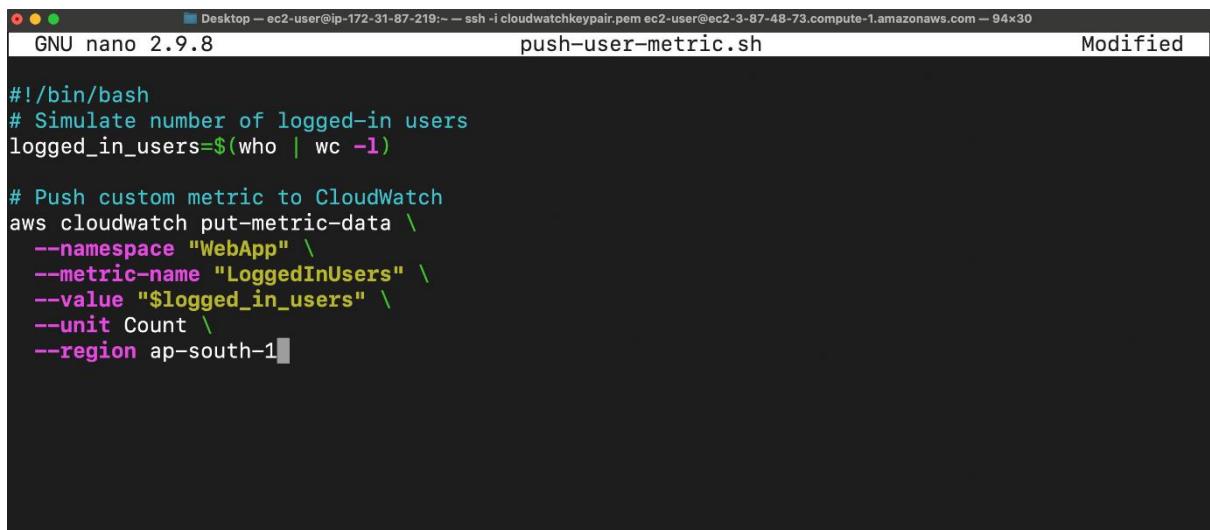
```
[ec2-user@ip-172-31-87-219 ~]$ sudo yum install -y aws-cli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package awscli-1.18.147-1.amzn2.0.2.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-219 ~]$ aws --version
aws-cli/1.18.147 Python/2.7.18 Linux/5.10.234-225.921.amzn2.x86_64 botocore/1.18.6
[ec2-user@ip-172-31-87-219 ~]$
```

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



```
[ec2-user@ip-172-31-87-219 ~]$ sudo yum install -y aws-cli
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package awscli-1.18.147-1.amzn2.0.2.noarch already installed and latest version
Nothing to do
[ec2-user@ip-172-31-87-219 ~]$ aws --version
aws-cli/1.18.147 Python/2.7.18 Linux/5.10.234-225.921.amzn2.x86_64 botocore/1.18.6
[ec2-user@ip-172-31-87-219 ~]$ nano push-user-metric.sh
```

In that file insert this code.



```
GNU nano 2.9.8          push-user-metric.sh          Modified
#!/bin/bash
# Simulate number of logged-in users
logged_in_users=$(who | wc -l)

# Push custom metric to CloudWatch
aws cloudwatch put-metric-data \
  --namespace "WebApp" \
  --metric-name "LoggedInUsers" \
  --value "$logged_in_users" \
  --unit Count \
  --region ap-south-1
```

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

```
[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 "$logged_in_users" \
--unit Count \
--region us-east-1
[ec2-user@ip-172-31-87-219 ~]$
```

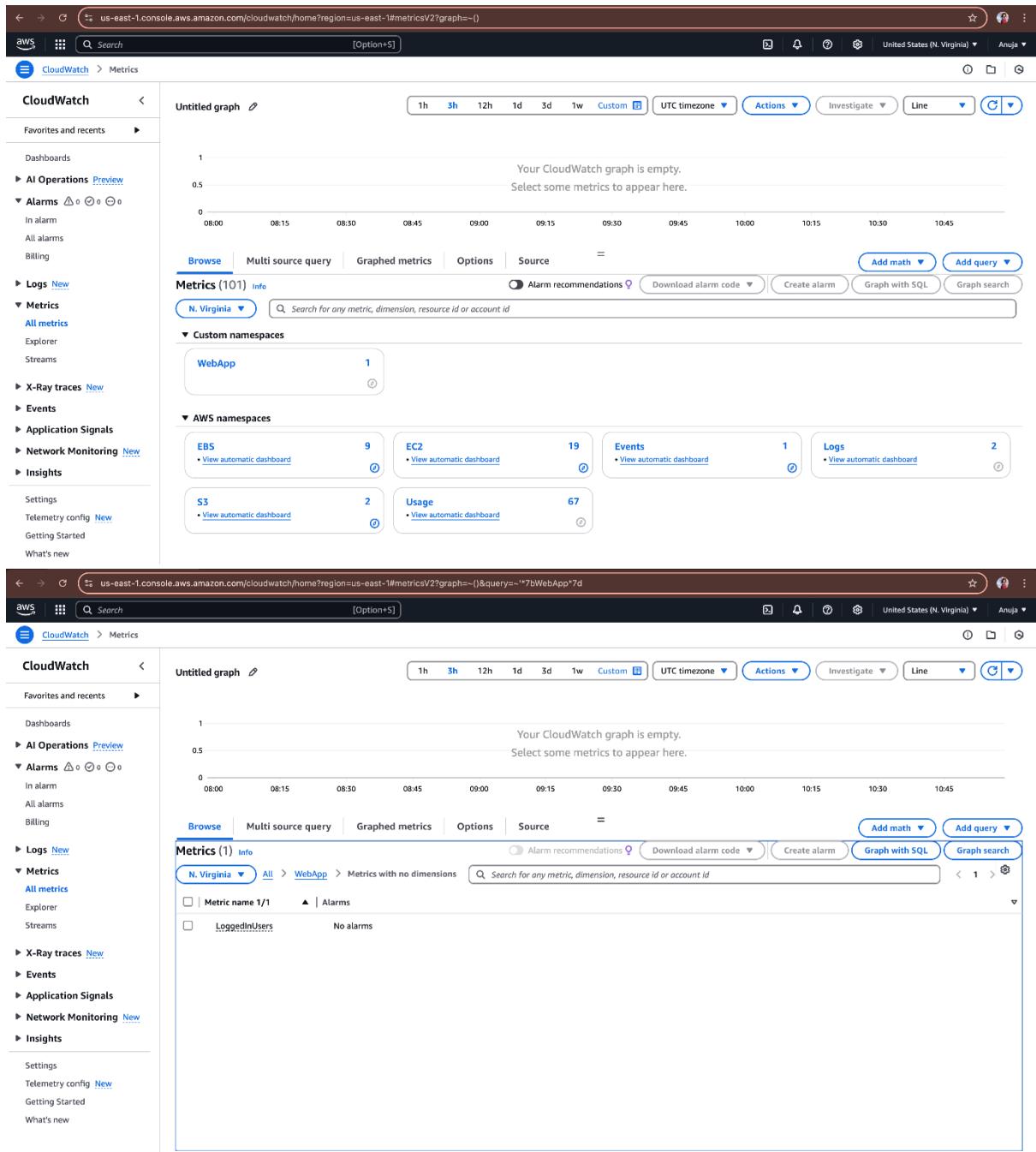
Now the permission of the new created file using the command shown.

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

Now final step need to push metrics.

```
[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.

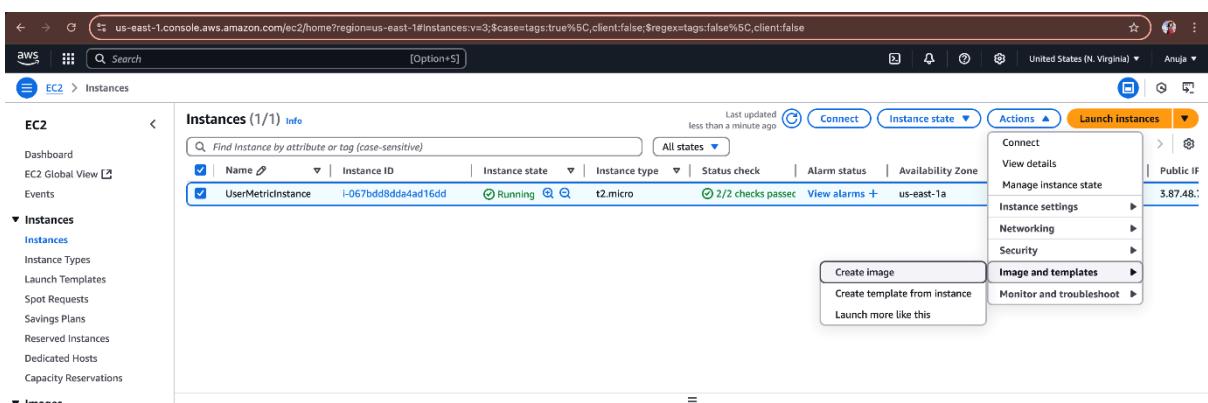
First you need to open your terminal and open crontab -e

```
Desktop — ec2-user@ip-172-31-87-219:~ — ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com — 94x30
[ec2-user@ip-172-31-87-219 ~]$ crontab -e
```

Again push your metrics.

```
Desktop — ec2-user@ip-172-31-87-219:~ — ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com — 94x30
*/5 * * * * /home/ec2-user/push-user-metric.sh
```

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



The screenshot shows the AWS EC2 Instances page. A single instance, 'UserMetricInstance' (ID: i-067bdd8dda4ad16dd), is listed. The 'Actions' menu is open, and the 'Image and templates' option is selected. A sub-menu for 'Image and templates' is displayed, with 'Create image' highlighted.

Give your image name and click on next.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateImage:instanceId=i-067bdd8dda4ad16dd

EC2 > Instances > i-067bdd8dda4ad16dd > 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-067bdd8dda4ad16dd (UserMetricInstance)

Image name
WebAppAMI

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
EBS	/dev/x...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

No tags associated with the resource.

Then click on create image.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateImage:instanceId=i-067bdd8dda4ad16dd

EC2 > Instances > i-067bdd8dda4ad16dd > Create image

Image name
WebAppAMI

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
EBS	/dev/x...	Create new snapshot from v...	8	EBS General Purpose SSD - ...	100		<input checked="" type="checkbox"/> Enable	<input type="checkbox"/> Enable

[Add volume](#)

During the image creation process, Amazon EC2 creates a snapshot of each of the above volumes.

Tags - optional
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Tag image and snapshots together
Tag the image and the snapshots with the same tag.

Tag image and snapshots separately
Tag the image and the snapshots with different tags.

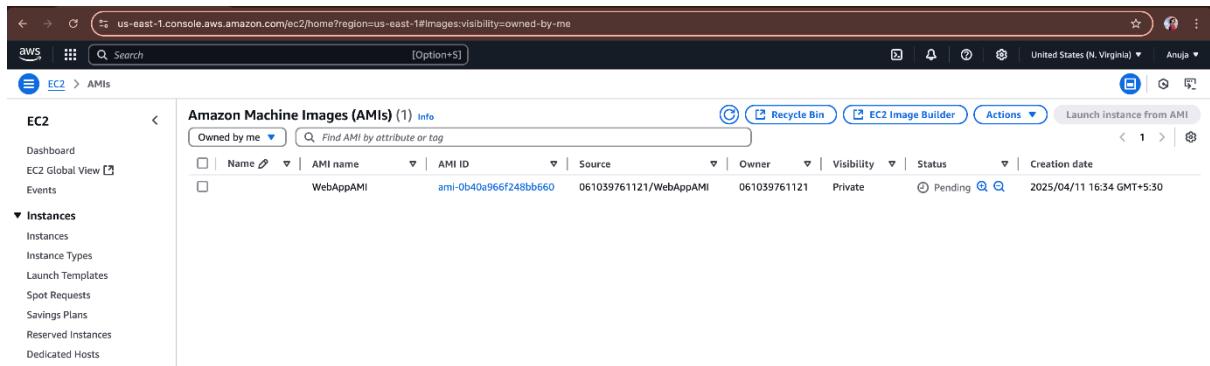
No tags associated with the resource.

[Add new tag](#)

You can add up to 50 more tags.

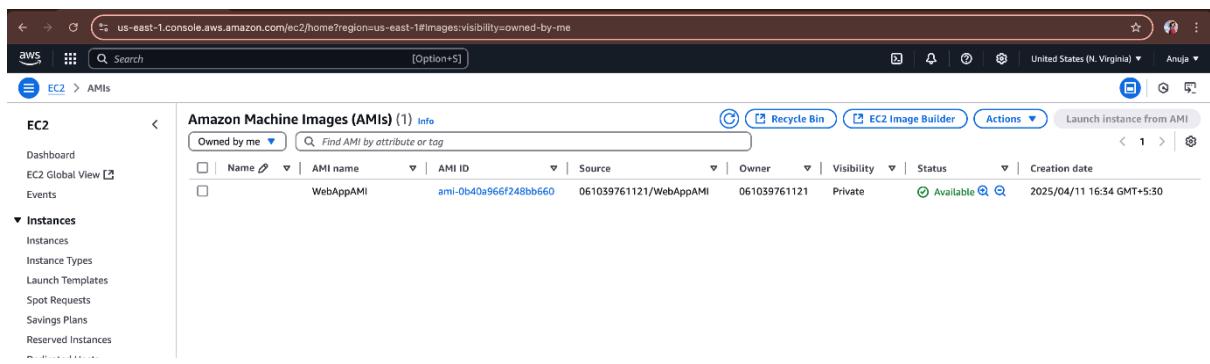
[Cancel](#) [Create Image](#)

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



The screenshot shows the AWS EC2 AMIs page. The left sidebar is collapsed. The main content area is titled "Amazon Machine Images (AMIs) (1) Info". A table lists one AMI: "WebAppAMI" (ami-0b40a966f248bb660). The table includes columns for Name, AMI name, AMI ID, Source, Owner, Visibility, Status, and Creation date. The "Status" column shows "Pending" with a "Q" icon. The "Actions" dropdown menu includes "Recycle Bin", "EC2 Image Builder", and "Launch instance from AMI". The top right corner shows "United States (N. Virginia)" and "Anuja".

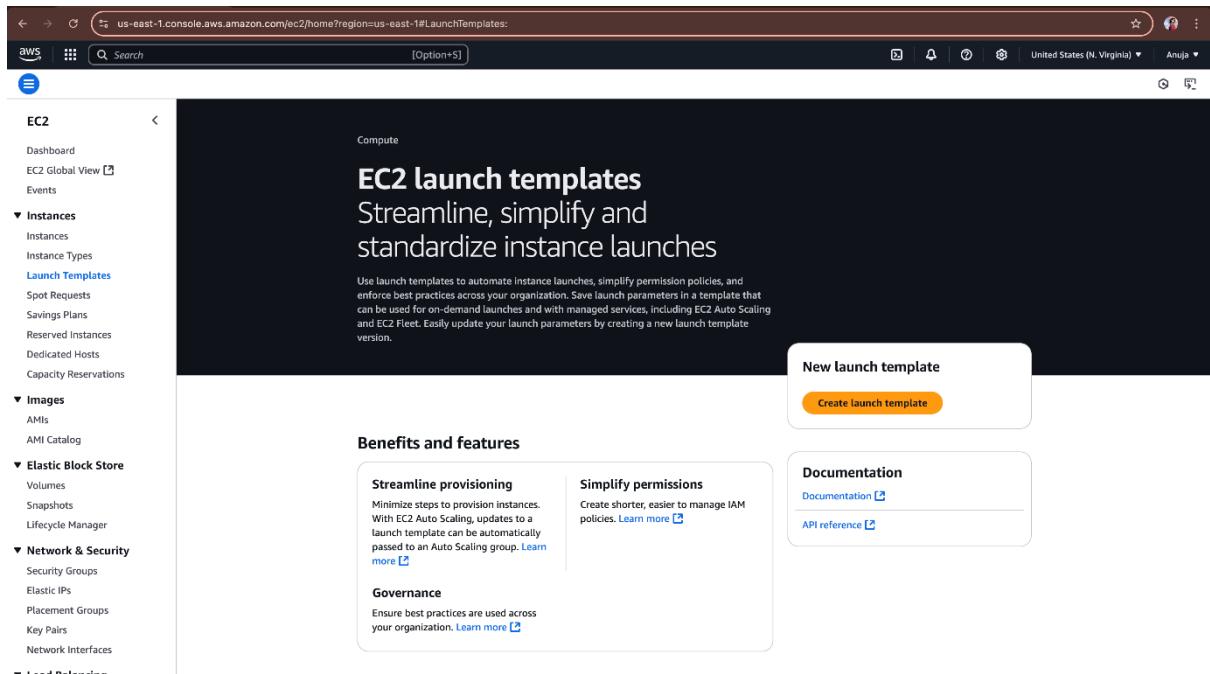
Once your image is available you can ready to use it.



The screenshot shows the AWS EC2 AMIs page. The left sidebar is collapsed. The main content area is titled "Amazon Machine Images (AMIs) (1) Info". A table lists one AMI: "WebAppAMI" (ami-0b40a966f248bb660). The table includes columns for Name, AMI name, AMI ID, Source, Owner, Visibility, Status, and Creation date. The "Status" column shows "Available" with a green checkmark icon. The "Actions" dropdown menu includes "Recycle Bin", "EC2 Image Builder", and "Launch instance from AMI". The top right corner shows "United States (N. Virginia)" and "Anuja".

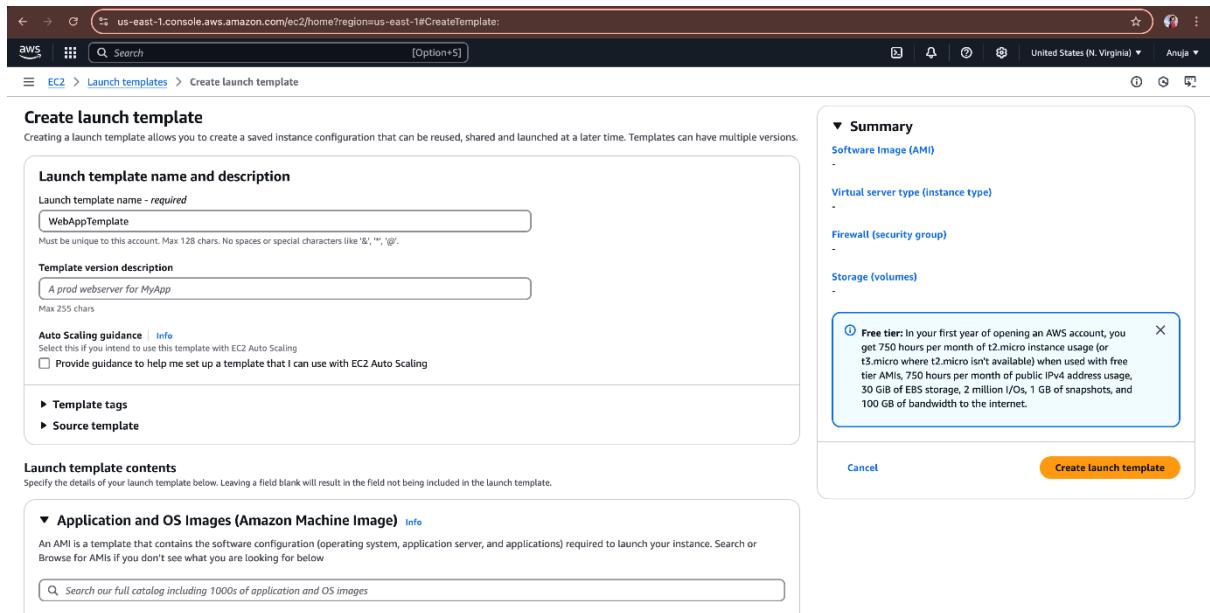
Step 5: Creating a launch template.

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



The screenshot shows the AWS EC2 Launch Templates page. The left sidebar is collapsed. The main content area is titled "Compute" and "EC2 launch templates". It says "Streamline, simplify and standardize instance launches". A callout box on the right says "New launch template" with a "Create launch template" button. Below this, there are sections for "Benefits and features", "Streamline provisioning", "Simplify permissions", "Governance", "Documentation", and "API reference".

Give your launch template a name.



Create launch template
Creating a launch template allows you to create a saved instance configuration that can be reused, shared and launched at a later time. Templates can have multiple versions.

Launch template name and description

Launch template name - required
WebAppTemplate
Must be unique to this account. Max 128 chars. No spaces or special characters like '&', '*', '@'.

Template version description
A prod webserver for MyApp
Max 255 chars

Auto Scaling guidance Info
Select this if you intend to use this template with EC2 Auto Scaling
 Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Template tags

Source template

Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Summary
Software Image (AMI)
WebAppAMI
ami-0b40a966f248bb660

Virtual server type (instance type)

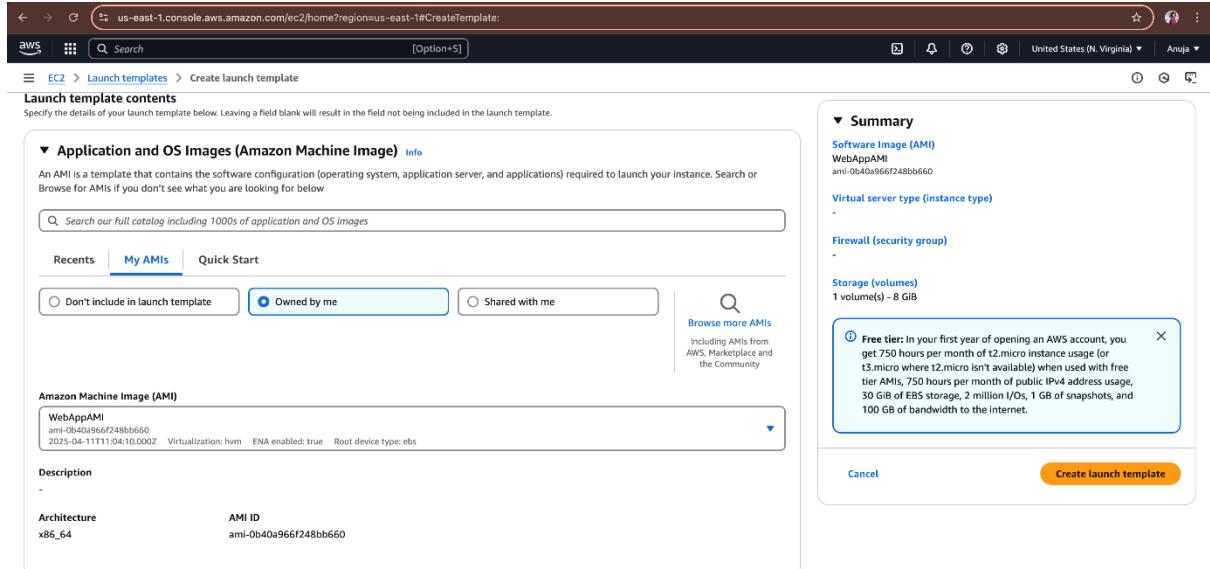
Firewall (security group)

Storage (volumes)

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Create launch template

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



Launch template contents
Specify the details of your launch template below. Leaving a field blank will result in the field not being included in the launch template.

Application and OS Images (Amazon Machine Image) Info
An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Recents **My AMIs** Quick Start

Don't include in launch template Owned by me Shared with me

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Amazon Machine Image (AMI)
WebAppAMI
ami-0b40a966f248bb660
2025-04-11T11:04:10.000Z Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Architecture x86_64 AMI ID ami-0b40a966f248bb660

Summary
Software Image (AMI)
WebAppAMI
ami-0b40a966f248bb660

Virtual server type (instance type)

Firewall (security group)

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs. 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

Create launch template

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

▼ Instance type [Info](#) | [Get advice](#)

Instance type

t2.micro
Family: t2 1 vCPU 1 GB Memory Current generation: true On-Demand Windows base pricing: 0.0162 USD per Hour
On-Demand Ubuntu Pro base pricing: 0.0134 USD per Hour On-Demand SUSE base pricing: 0.0116 USD per Hour
On-Demand RHEL base pricing: 0.026 USD per Hour On-Demand Linux base pricing: 0.0116 USD per Hour

Free tier eligible

All generations [Compare instance types](#)

Additional costs apply for AMIs with pre-installed software

▼ Key pair (login) [Info](#)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name

cloudwatchkeypair [Create new key pair](#)

▼ Network settings [...](#)

Virtual server type (instance type)
t2.micro

Firewall (security group)
-

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Create launch template](#)

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

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateTemplate:

aws [Search](#) [Option+S]

EC2 > [Launch templates](#) > Create launch template

Select existing security group Create security group

Security groups [Info](#)
Select security groups [Compare security group rules](#)

Launch-wizard-1 sg-04e1fdb5c82bd8084 [X](#)
VPC: vpc-01ea9bb028b6573725

► Advanced network configuration

▼ Storage (volumes) [Info](#)

EBS Volumes [Hide details](#)

Volume 1 (AMI Root): 8 GiB, EBS, General purpose SSD (gp2)
AMI Volumes are not included in the template unless modified

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage [X](#)

[Add new volume](#)

▼ Resource tags [Info](#)

No resource tags are currently included in this template. Add a resource tag to include it in the launch template.

[Add new tag](#)

You can add up to 50 more tags.

► Advanced details [Info](#)

Summary

Software Image (AMI)
WebAppAMI
ami-0b10a966f248b8660

Virtual server type (instance type)
t2.micro

Firewall (security group)
Launch-wizard-1

Storage (volumes)
1 volume(s) - 8 GiB

Free tier: In your first year of opening an AWS account, you get 750 hours per month of t2.micro instance usage (or t3.micro where t2.micro isn't available) when used with free tier AMIs, 750 hours per month of public IPv4 address usage, 30 GiB of EBS storage, 2 million I/Os, 1 GB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Create launch template](#)

You can see you have launched your template.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LaunchTemplates:

aws [Search](#) [Option+S]

EC2

Launch Templates (1) [Info](#)

Actions [Create launch template](#)

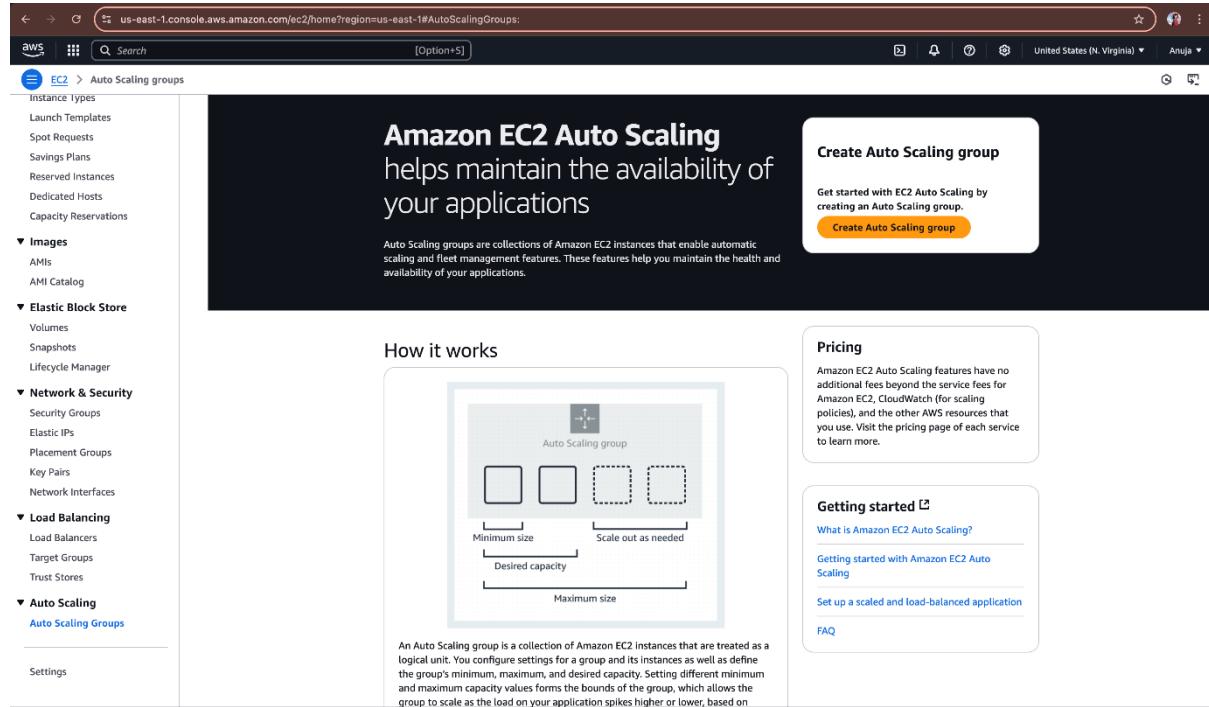
Launch Template ID Launch Template Name Default Version Latest Version Create Time Created By Manager

lt-093ac1629edd8e259 WebAppTemplate 1 1 2025-04-11T11:12:53.000Z arn:aws:iam::061039761121:root false

Select a launch 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.



Amazon EC2 Auto Scaling helps maintain the availability of your applications

Auto Scaling groups are collections of Amazon EC2 instances that enable automatic scaling and fleet management features. These features help you maintain the health and availability of your applications.

Create Auto Scaling group

Get started with EC2 Auto Scaling by creating an Auto Scaling group.

Create Auto Scaling group

How it works

An Auto Scaling group is a collection of Amazon EC2 instances that are treated as a logical unit. You configure settings for a group and its instances as well as define the group's minimum, maximum, and desired capacity. Setting different minimum and maximum capacity values forms the bounds of the group, which allows the group to scale as the load on your application spikes higher or lower, based on

Pricing

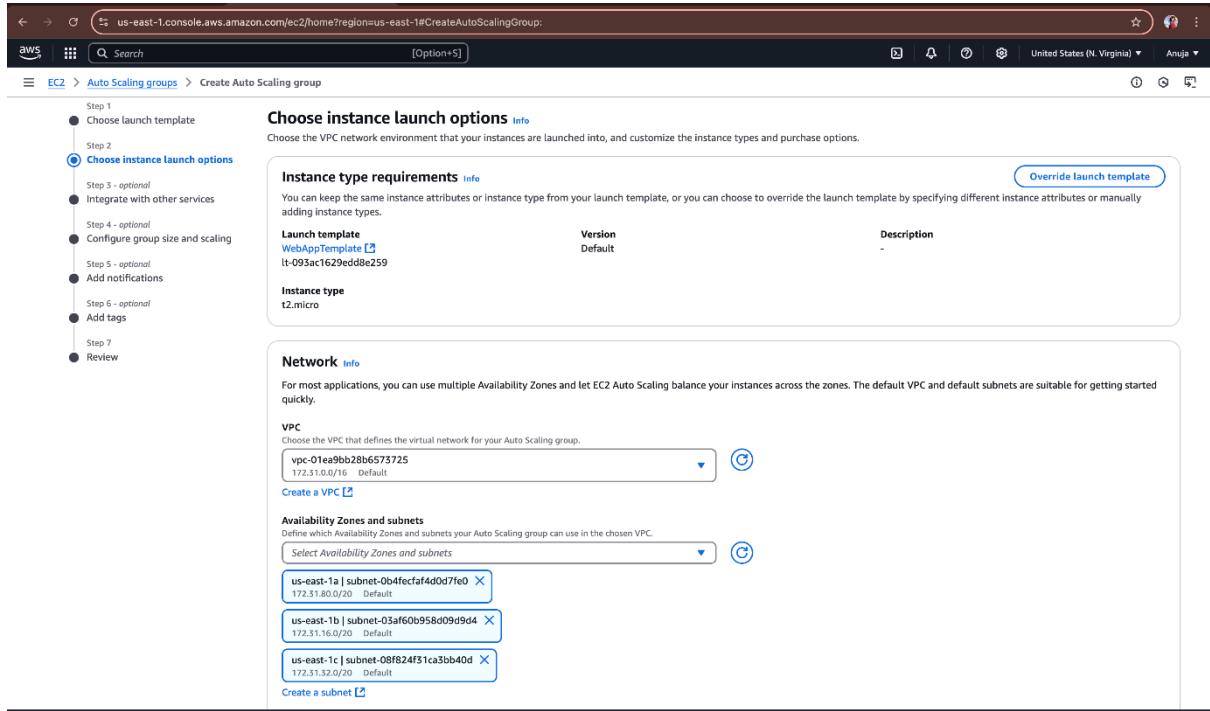
Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

Getting started

What is Amazon EC2 Auto Scaling?
Getting started with Amazon EC2 Auto Scaling
Set up a scaled and load-balanced application
FAQ

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.



Choose instance launch options Info
Choose the VPC network environment that your instances are launched into, and customize the instance types and purchase options.

Instance type requirements Info
You can keep the same instance attributes or instance type from your launch template, or you can choose to override the launch template by specifying different instance attributes or manually adding instance types.

Launch template	Version	Description
WebAppTemplate <small>l-095ac1629edd8e259</small>	Default	-

Instance type Info
t2.micro

Override launch template

Network Info
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-01e9b028b6573725
172.31.0.0/16 Default

Create a VPC Info

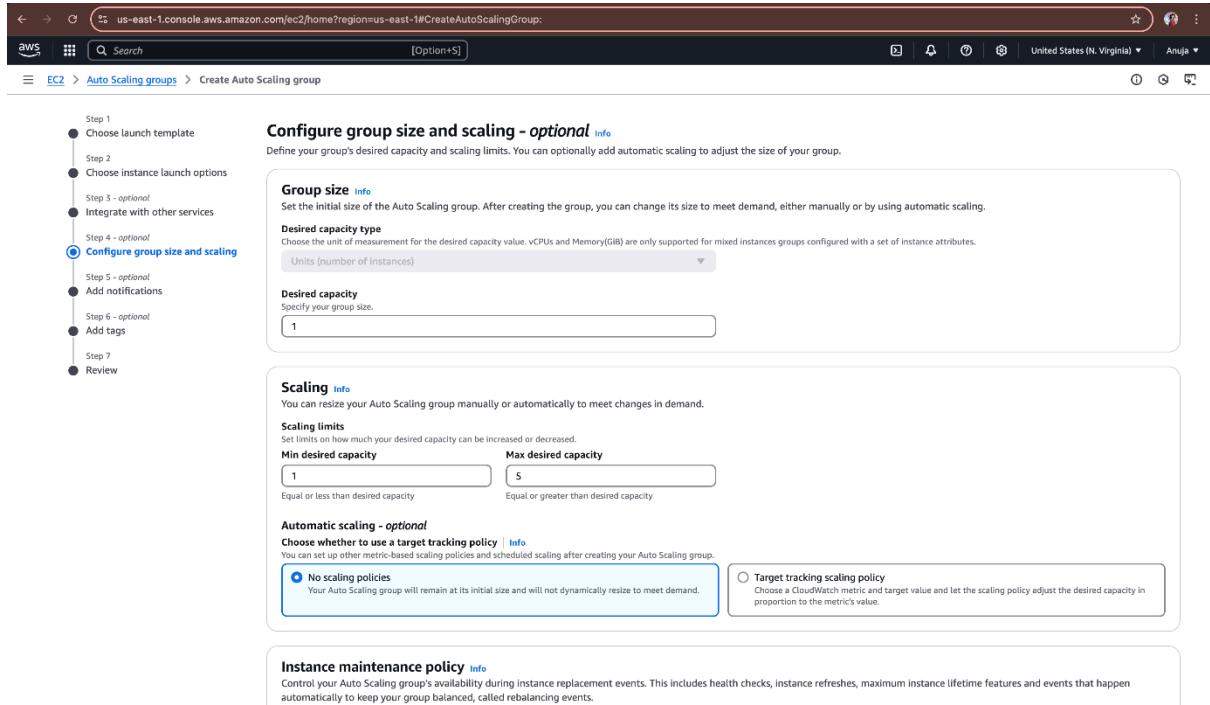
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.

Select Availability Zones and subnets Info

us-east-1a subnet-0b4fecf44d0d7fe0 172.31.80.0/20 Default
us-east-1b subnet-03af60b958d09d964 172.31.16.0/20 Default
us-east-1c subnet-08f824f31ca3bb40d 172.31.32.0/20 Default

Create a subnet Info

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.



Configure group size and scaling - optional Info
Define your group's desired capacity and scaling limits. You can optionally add automatic scaling to adjust the size of your group.

Group size Info
Set the initial size of the Auto Scaling group. After creating the group, you can change its size to meet demand, either manually or by using automatic scaling.

Desired capacity type Info
Choose the unit of measurement for the desired capacity value. vCPUs and Memory(GiB) are only supported for mixed instances groups configured with a set of instance attributes.

Units (number of instances)

Desired capacity Info
Specify your group size.
1

Scaling Info
You can resize your Auto Scaling group manually or automatically to meet changes in demand.

Scaling limits
Set limits on how much your desired capacity can be increased or decreased.

Min desired capacity Info
1

Max desired capacity Info
5

Equal or less than desired capacity

Equal or greater than desired capacity

Automatic scaling - optional
Choose whether to use a target tracking policy Info
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

No scaling policies
Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

Target tracking scaling policy
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Instance maintenance policy Info
Control your Auto Scaling group's availability during instance replacement events. This includes health checks, instance refreshes, maximum instance lifetime features and events that happen automatically to keep your group balanced, called rebalancing events.

Review all your details and press create auto scaling group.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

aws Search [Option+S]

EC2 > Auto Scaling groups > Create Auto Scaling group

Review Info

Step 1: Choose launch template

Step 2: Choose instance launch options

Step 3 - optional: Integrate with other services

Step 4 - optional: Configure group size and scaling

Step 5 - optional: Add notifications

Step 6 - optional: Add tags

Step 7: Review

Step 1: Choose launch template

Group details

Auto Scaling group name: WebApp-ASG

Launch template

Launch template	Version	Description
WebAppTemplate Edit	lt-093ac1629ed8e259	

Step 2: Choose instance launch options

Network

VPC: vpc-01ea9bb28b5573725 [Edit](#)

Availability Zones and subnets

Availability Zone	Subnet	Subnet CIDR range
us-east-1a	subnet-0b4fecfaf4d0d7fe0 Edit	172.31.80.0/20
us-east-1b	subnet-03af60b958d09d9d4 Edit	172.31.16.0/20
us-east-1c	subnet-08f824f31ca3bb40d Edit	172.31.32.0/20

Availability Zone distribution

Balanced best effort

Instance type requirements

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup:

aws Search [Option+S]

EC2 > Auto Scaling groups > Create Auto Scaling group

Instance maintenance policy

Replacement behavior	Min healthy percentage	Max healthy percentage
No policy	-	-

Additional settings

Instance scale-in protection	Monitoring	Default instance warmup
Disabled	Disabled	Disabled

Capacity Reservation preference

Preference	Capacity Reservation IDs	Resource Groups
Default	-	-

Step 5: Add notifications

Notifications

No notifications

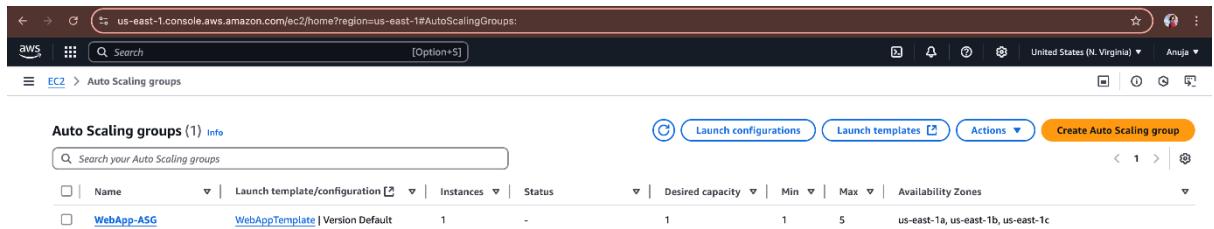
Step 6: Add tags

Tags (0)

Key	Value	Tag new instances
No tags		

[Preview code](#) Cancel Previous **Create Auto Scaling group**

You can see your auto scaling group was created.

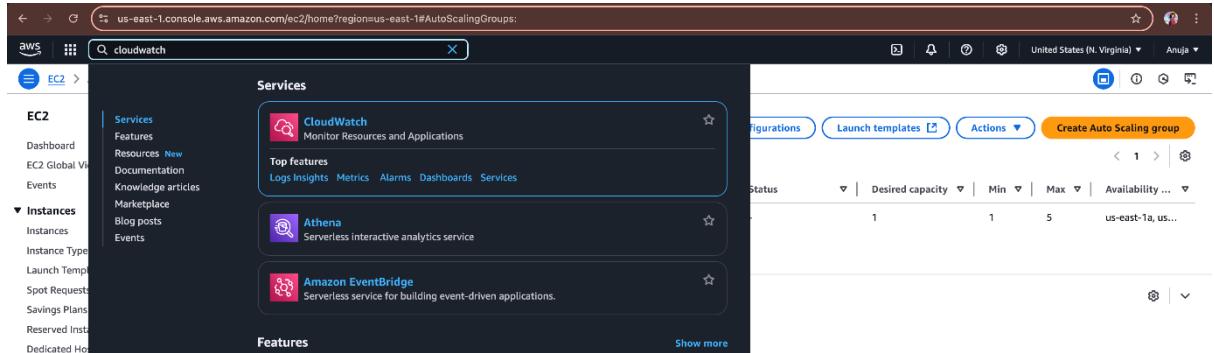


The screenshot shows the AWS EC2 Auto Scaling Groups page. At the top, there is a search bar with the placeholder 'Search' and a dropdown menu 'Option+5'. The main content area is titled 'Auto Scaling groups (1) Info'. It shows a table with one row for 'WebApp-ASG'. The table columns are: Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. The 'Availability Zones' column shows 'us-east-1a, us-east-1b, us-east-1c'. The 'Actions' menu at the top right includes 'Launch configurations', 'Launch templates', 'Actions', and 'Create Auto Scaling group'.

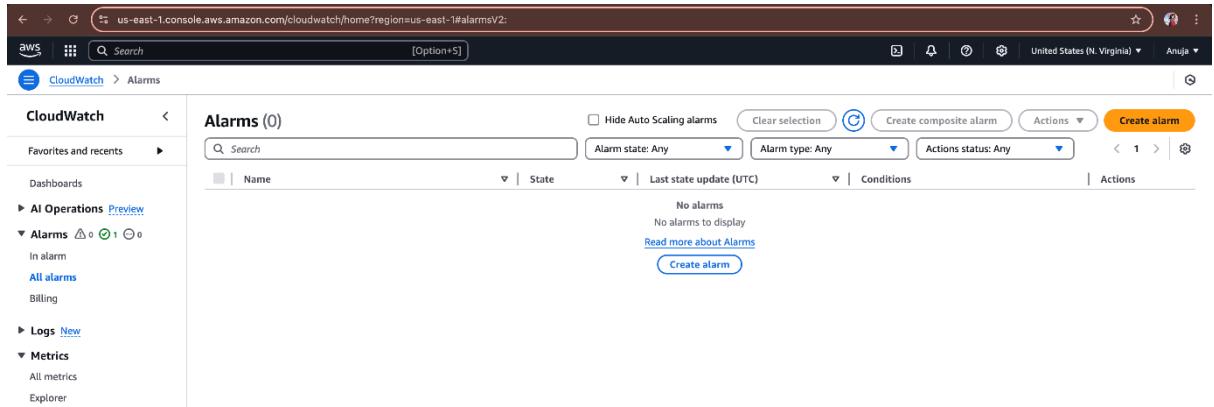
Step 7: Creating 2 alarms for scaling in and scaling out of instance in cloud watch.

Alarm 1:

Go to cloud watch and click on create alarms.



The screenshot shows the AWS CloudWatch Services page. The search bar at the top contains 'cloudwatch'. The 'Services' section is expanded, showing 'CloudWatch' (selected), 'Features', 'Resources New', 'Documentation', 'Knowledge articles', 'Marketplace', 'Blog posts', and 'Events'. Below this, the 'Features' section is partially visible. The main content area shows a table for 'CloudWatch' with columns: Status, Desired capacity, Min, Max, and Availability ... (with values 1, 1, 5, and 'us-east-1a, us...'). The 'Actions' menu at the top right includes 'Launch configurations', 'Launch templates', 'Actions', and 'Create Auto Scaling group'.



The screenshot shows the AWS CloudWatch Alarms page. The search bar at the top contains 'Search' and 'Option+5'. The left sidebar shows 'CloudWatch' (selected), 'Favorites and recents', 'Dashboards', 'AI Operations', 'Alarms' (with 1 item), 'All alarms', 'Billing', 'Logs', 'Metrics', and 'Explorer'. The main content area is titled 'Alarms (0)'. It includes filters for 'Hide Auto Scaling alarms', 'Clear selection', 'Create composite alarm', 'Actions', and 'Create alarm'. A search bar and a table header with columns: Name, State, Last state update (UTC), Conditions, and Actions. The table body shows 'No alarms' and 'No alarms to display'. A 'Create alarm' button is located at the bottom.

Now select your metrics which you created earlier.

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?-(Page-MetricSelection-AlarmType-MetricAlarm-AlarmData-(Metrics-(-)-AlarmName-~-AlarmDescription-~-ActionsEn...)

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

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

Your CloudWatch graph is empty.
Select some metrics to appear here.

1h 3h 12h 1d 3d 1w Custom UTC timezone Line

Browse Multi source query Graphed metrics Options Source

Metrics (176)

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

Your CloudWatch graph is empty.
Select some metrics to appear here.

1h 3h 12h 1d 3d 1w Custom UTC timezone Line

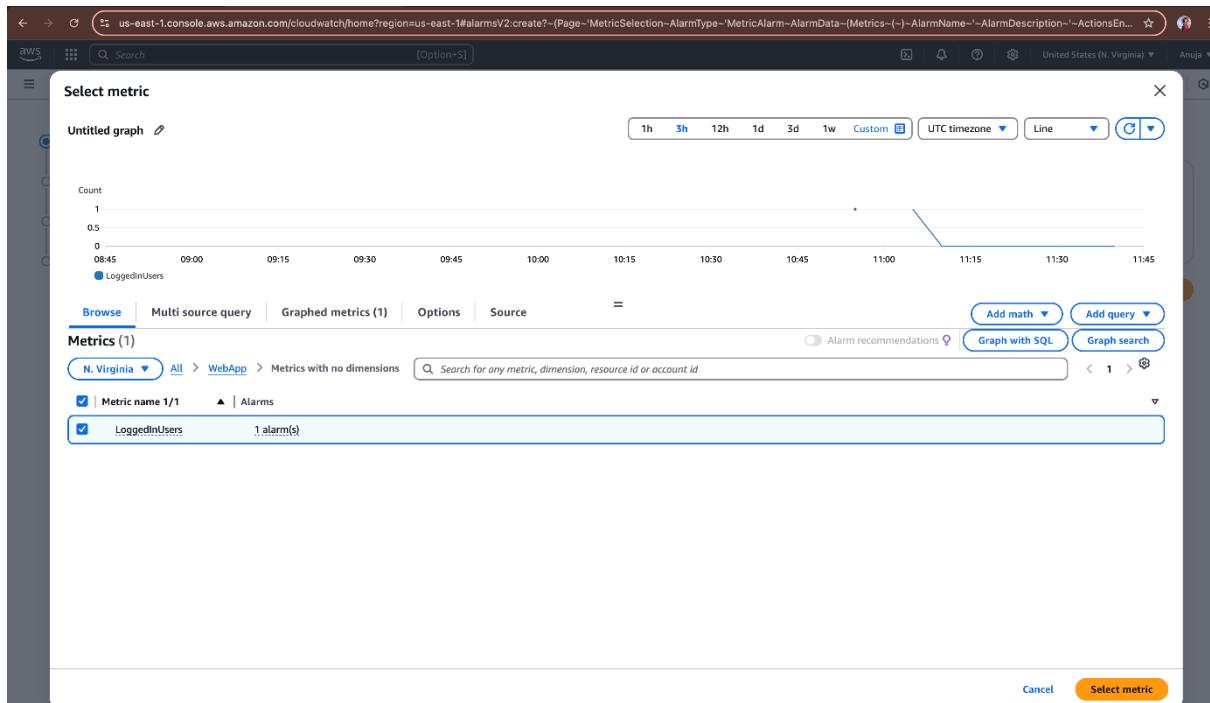
Browse Multi source query Graphed metrics Options Source

Metrics (1)

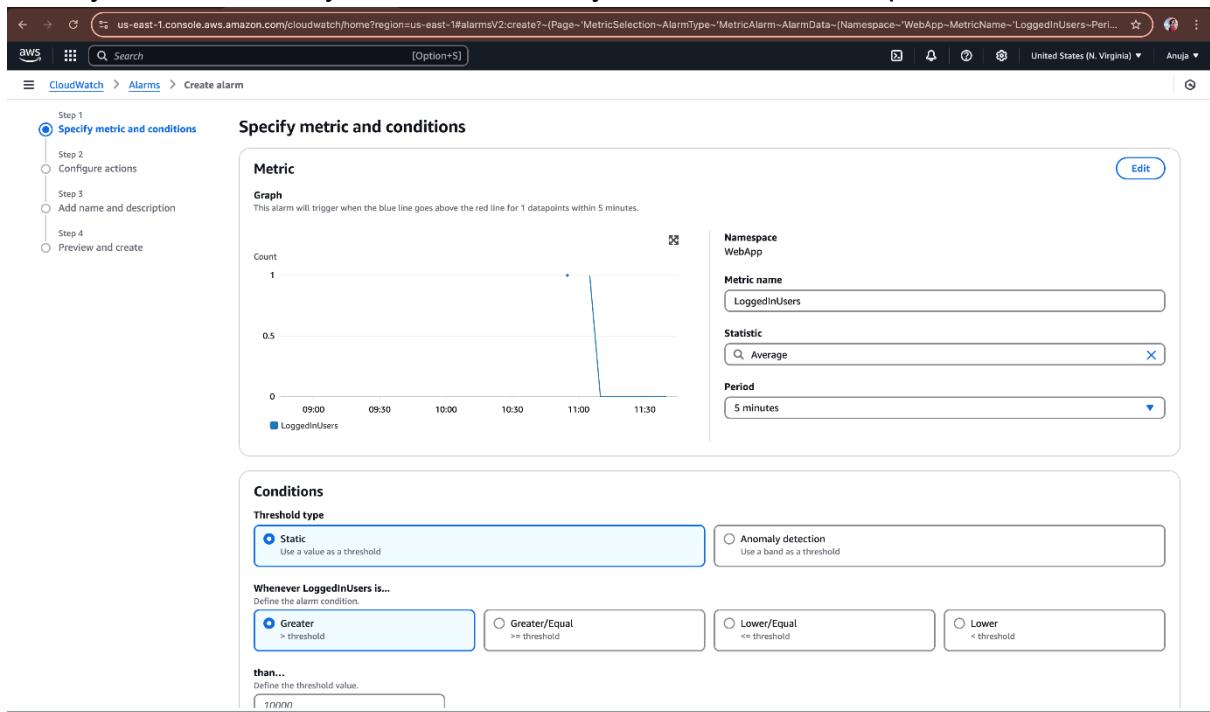
N. Virginia

WebApp

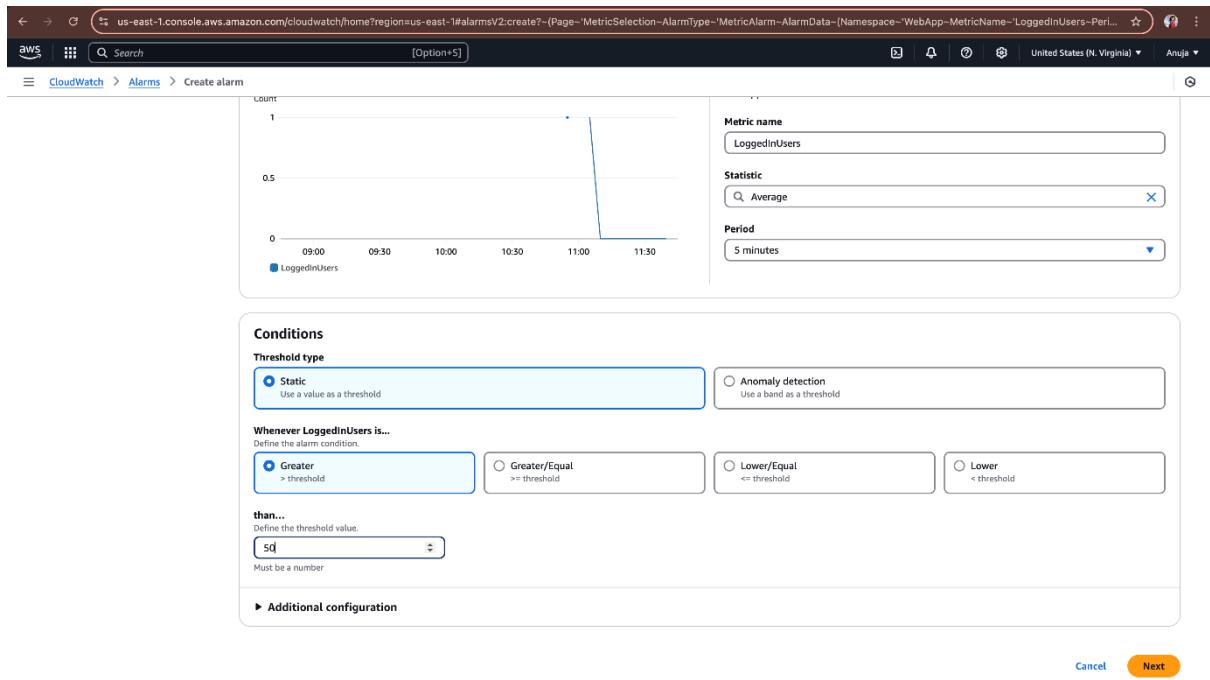
Metrics with no dimensions 1



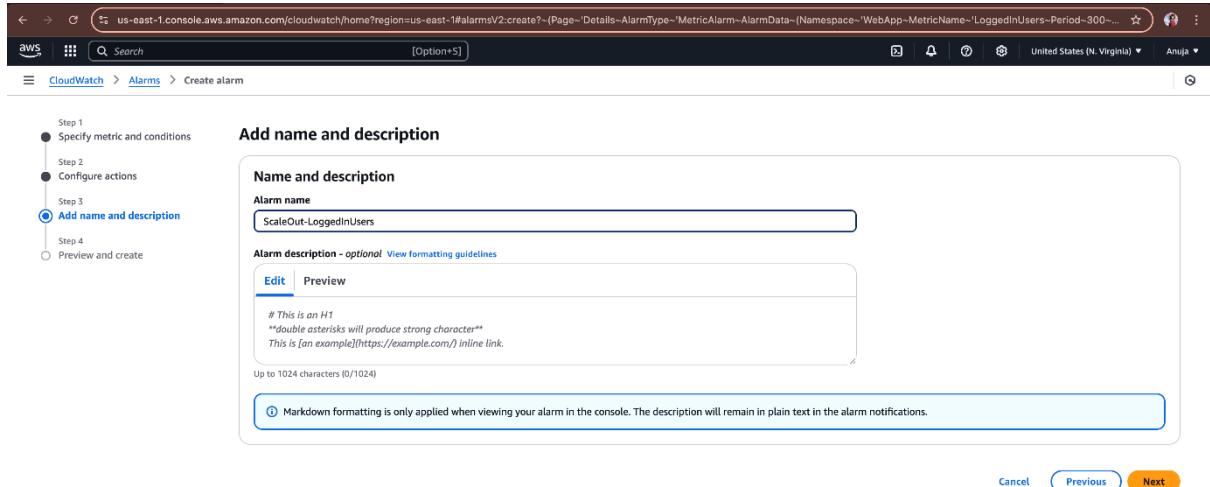
Then you can see the you have selected your desire metrices path.



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 preview all your data and create the alarm.

Step 3
Add name and description
Step 4
Preview and create

Metric

Graph
This alarm will trigger when the blue line goes above the red line for 1 datapoints within 5 minutes.

Count

50

25

0

09:00 09:30 10:00 10:30 11:00 11:30

LoggedinUsers

Namespace
WebApp

Metric name
LoggedinUsers

Statistic
Average

Period
5 minutes

Conditions

Threshold type
Static

Whenever LoggedinUsers is
Greater (>)

than...
50

► Additional configuration

Step 2: Configure actions

Actions

You can see your alarm was created.

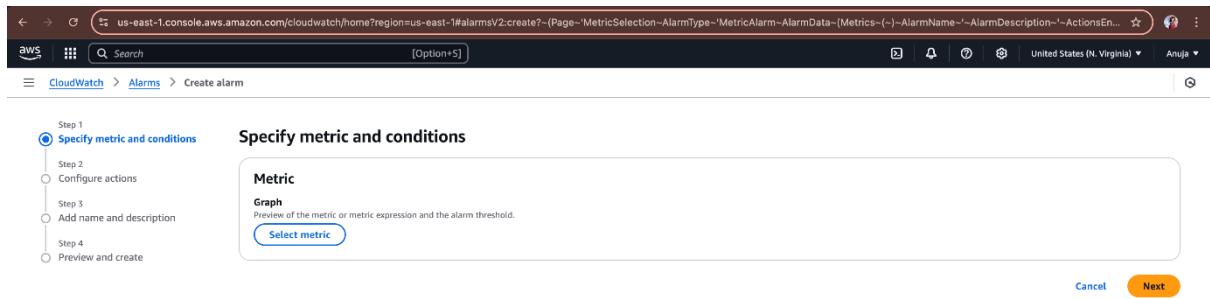
Successfully created alarm ScaleOut-LoggedinUsers.

Alarms (1)

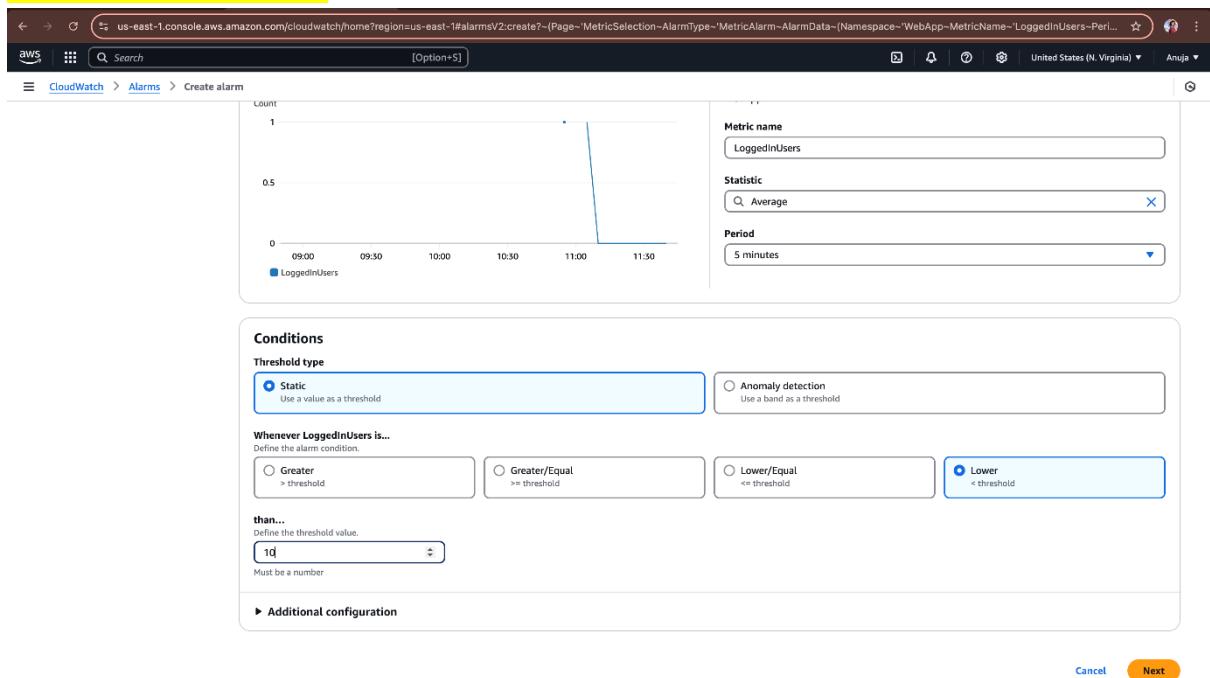
Name	State	Last state update (UTC)	Conditions	Actions
ScaleOut-LoggedinUsers	Insufficient data	2025-04-11 11:49:17	LoggedinUsers > 50 for 1 datapoints within 5 minutes	No actions

Create alarm

Same way go and create one more alarm.



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



Give alaram name as scale in logged in users.

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~Details~AlarmType~MetricAlarm~AlarmData~Namespace~'WebApp~MetricName~'LoggedInUsers~Period~300~)

CloudWatch > Alarms > Create alarm

Step 3
Specify metric and conditions
Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create

Add name and description

Name and description

Alarm name
ScaleIn-LoggedInUsers

Alarm description - optional [View formatting guidelines](#)

Edit **Preview**

This is an H1
double asterisks will produce strong character
This is [an example](https://example.com/) inline link.

Up to 1024 characters (0/1024)

Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications.

Cancel **Previous** **Next**

Preview all the details and create the second alaram.

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~Details~AlarmType~MetricAlarm~AlarmData~Namespace~'WebApp~MetricName~'LoggedInUsers~Period~300~)

CloudWatch > Alarms > Create alarm

Step 2
Configure actions
Step 3
Add name and description
Step 4
Preview and create

Step 1: Specify metric and conditions

Metric

Graph
This alarm will trigger when the blue line goes below the red line for 1 datapoints within 5 minutes.

Count

10 10

5

0

09:00 09:30 10:00 10:30 11:00 11:30

LoggedInUsers

Namespace
WebApp

Metric name
LoggedInUsers

Statistic
Average

Period
5 minutes

Conditions

Threshold type
Static

Whenever LoggedInUsers is
Lower (<) than...
10

Additional configuration

Step 2: Configure actions

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:

CloudWatch > Alarms

Alarms (2)

Successfully created alarm ScaleIn-LoggedInUsers.

Actions

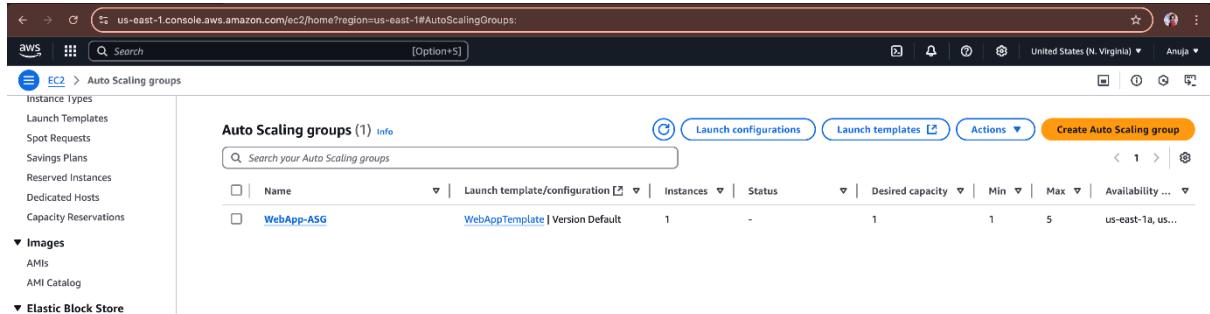
Hide Auto Scaling alarms Clear selection Create composite alarm Actions Create alarm

Search Alarm state: Any Alarm type: Any Actions status: Any

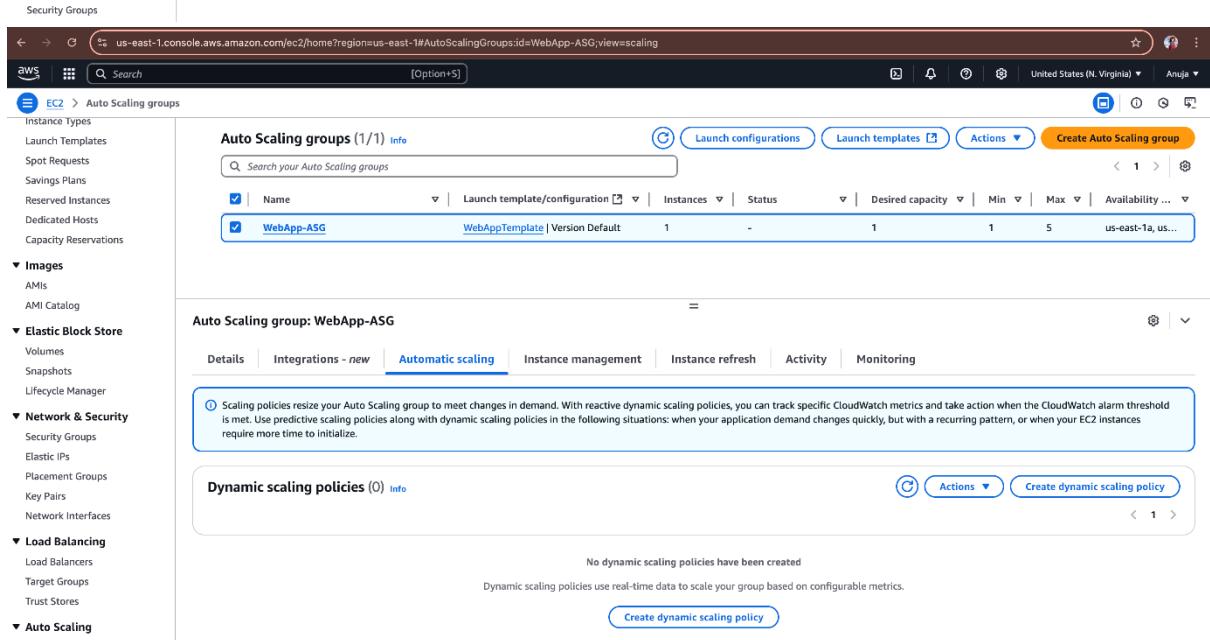
Name	State	Last state update (UTC)	Conditions	Actions
ScaleIn-LoggedInUsers	Insufficient data	2025-04-11 11:51:32	LoggedInUsers < 10 for 1 datapoints within 5 minutes	No actions
ScaleOut-LoggedInUsers	OK	2025-04-11 11:50:02	LoggedInUsers > 50 for 1 datapoints within 5 minutes	No actions

Step 8: Now both the alarms were created now we need to attach these alarms to Auto scaling group.

Go to auto scaling group then click on auto scaling.



The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar is collapsed. The main content area shows a table for 'Auto Scaling groups (1)'. The table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability. One row is visible for 'WebApp-ASG' with a status of 'WebAppTemplate | Version Default' and 1 instance.



The screenshot shows the details page for the 'WebApp-ASG' Auto Scaling group. The left sidebar is collapsed. The main content area shows the 'Auto Scaling group: WebApp-ASG' details. It includes tabs for Details, Integrations - new, Automatic scaling (which is selected), Instance management, Instance refresh, Activity, and Monitoring. A note about scaling policies is displayed: 'Scaling policies resize your Auto Scaling group to meet changes in demand. With reactive dynamic scaling policies, you can track specific CloudWatch metrics and take action when the CloudWatch alarm threshold is met. Use predictive scaling policies along with dynamic scaling policies in the following situations: when your application demand changes quickly, but with a recurring pattern, or when your EC2 instances require more time to initialize.' Below this, a section for 'Dynamic scaling policies (0)' is shown with a note: 'No dynamic scaling policies have been created. Dynamic scaling policies use real-time data to scale your group based on configurable metrics.' A 'Create dynamic scaling policy' button is available.

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

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#DynamicScalingPolicy:id=WebApp-ASG

AWS Search [Option+S] United States (N. Virginia) Anuja

EC2 > Auto Scaling groups > WebApp-ASG > Dynamic scaling policy

Instance Types

- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes
- Snapshots
- Lifecycle Manager

Network & Security

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers
- Target Groups
- Trust Stores

Auto Scaling

- Auto Scaling Groups

Create dynamic scaling policy

Policy type: Step scaling

Scaling policy name: ScaleOutPolicy

CloudWatch alarm: Choose an alarm that can scale capacity whenever: ScaleOut-LoggedInUsers

Take the action: Add

1 capacity units when LoggedInUsers <= 50

300 seconds

Cancel Create

Click on create.

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups:id=WebApp-ASG:view=scaling

AWS Search [Option+S] United States (N. Virginia) Anuja

EC2 > Auto Scaling groups

Instance Types

- Launch Templates
- Spot Requests
- Savings Plans
- Reserved Instances
- Dedicated Hosts
- Capacity Reservations

Images

- AMIs
- AMI Catalog

Elastic Block Store

- Volumes
- Snapshots
- Lifecycle Manager

Network & Security

- Security Groups
- Elastic IPs
- Placement Groups
- Key Pairs
- Network Interfaces

Load Balancing

- Load Balancers
- Target Groups
- Trust Stores

Auto Scaling

Dynamic scaling policy created or edited successfully.

Auto Scaling groups (1/1) Info

Launch configurations Launch templates Actions Create Auto Scaling group

Search your Auto Scaling groups

WebApp-ASG WebAppTemplate | Version Default 1 - 1 1 5 us-east-1a, us...

Auto Scaling group: WebApp-ASG

ScaleOutPolicy

Policy type: Step scaling

Enabled or disabled: Enabled

Execute policy when: No alarm selected

Take the action: Add 1 capacity units when 0 <= Metric name < +infinity

Instances need: 300 seconds to warm up after each step

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

EC2 > Auto Scaling groups > WebApp-ASG > Dynamic scaling policy

Create dynamic scaling policy

Policy type: Step scaling

Scaling policy name: ScaleInPolicy

CloudWatch alarm: Choose an alarm that can scale capacity whenever: ScaleIn-LoggedInUsers

Take the action: Remove

1 capacity units when 10 >= LoggedInUsers > -infinity

Create

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

Dynamic scaling policy created or edited successfully.

Auto Scaling groups (1/1)

Auto Scaling group: WebApp-ASG

ScaleInPolicy

Policy type: Step scaling

Enabled or disabled: Enabled

Execute policy when: No alarm selected

Take the action: Remove 1 capacity units when +infinity <= Metric name < 0

ScaleOutPolicy

Policy type: Step scaling

Enabled or disabled: Enabled

Execute policy when: ScaleOut-LoggedInUsers

Take the action: Add 1 capacity units when 50 <= LoggedInUsers < +infinity

Instances need: 300 seconds to warm up after each step

Predictive scaling policies (0)

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

```
Desktop - ec2-user@ip-172-31-87-219:~ - ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-8
[ec2-user@ip-172-31-87-219 ~]$ nano push-user-metric.sh
```

```
Desktop -- ec2-user@ip-172-31-87-219:~ -- ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com -- 94x30
[ec2-user@ip-172-31-87-219 ~]$ nano push-user-metric.sh
[ec2-user@ip-172-31-87-219 ~]$ cat push-user-metric.sh
#!/bin/bash

# Simulate number of logged-in users manually
logged_in_users=$1

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

```
Desktop -- ec2-user@ip-172-31-87-219:~ -- ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com -- 94x30
GNU nano 2.9.8                                         push-user-metric.sh
[ec2-user@ip-172-31-87-219 ~]$ !/bin/bash

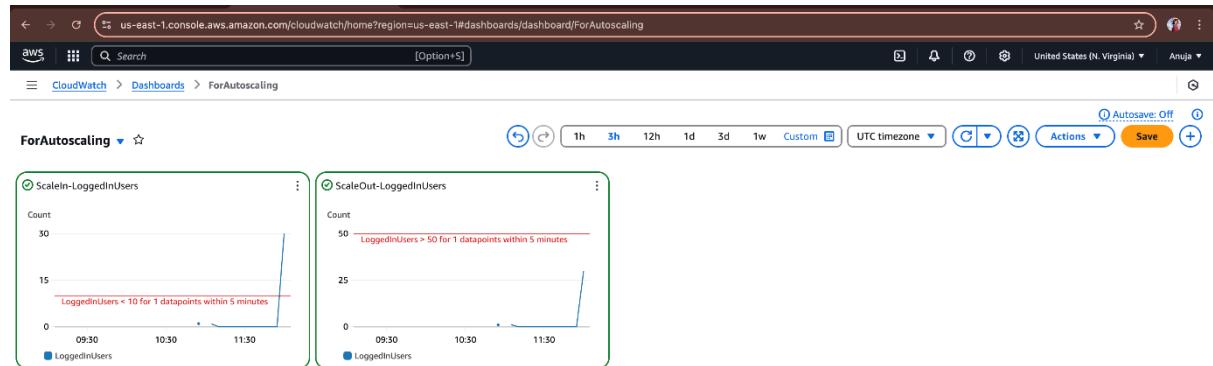
# Simulate number of logged-in users manually
logged_in_users=$1

# Push to CloudWatch
aws cloudwatch put-metric-data \
--namespace "WebApp" \
--metric-name "LoggedInUsers" \
--value "$logged_in_users" \
--unit Count \
--region us-east-1

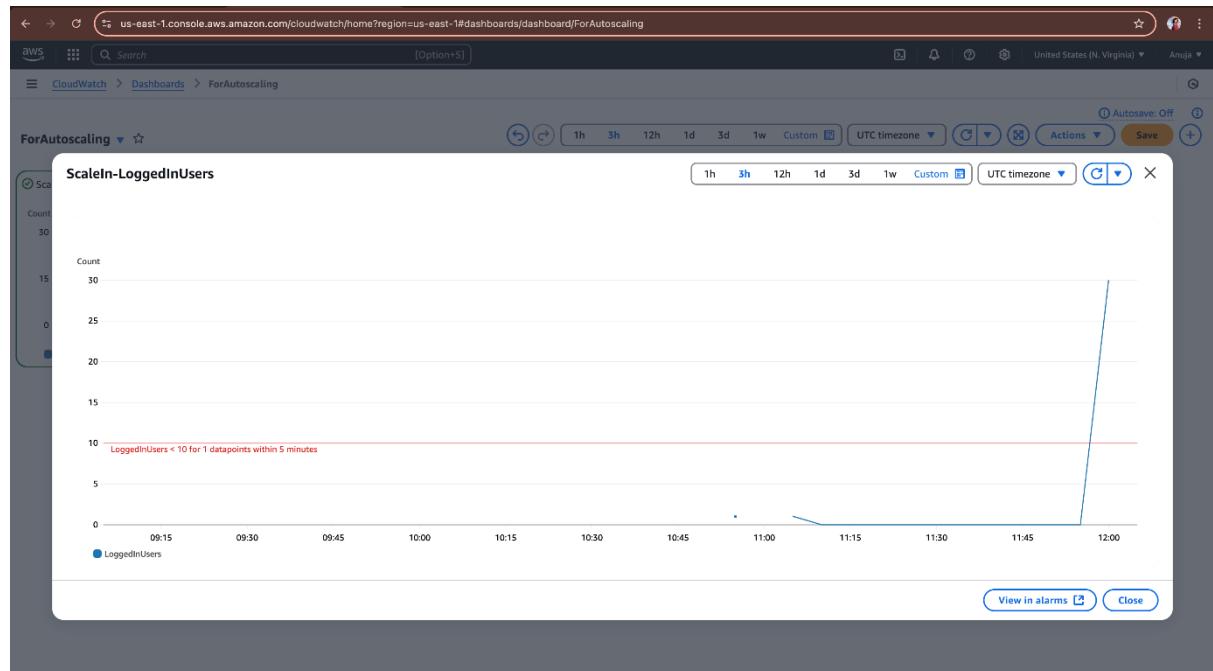
[ Read 12 lines ]
^G Get Help      ^O Write Out     ^W Where Is      ^K Cut Text      ^J Justify      ^C Cur Pos
^X Exit          ^R Read File     ^\ Replace       ^U Uncut Text    ^T To Linter    ^_ Go To Line
```

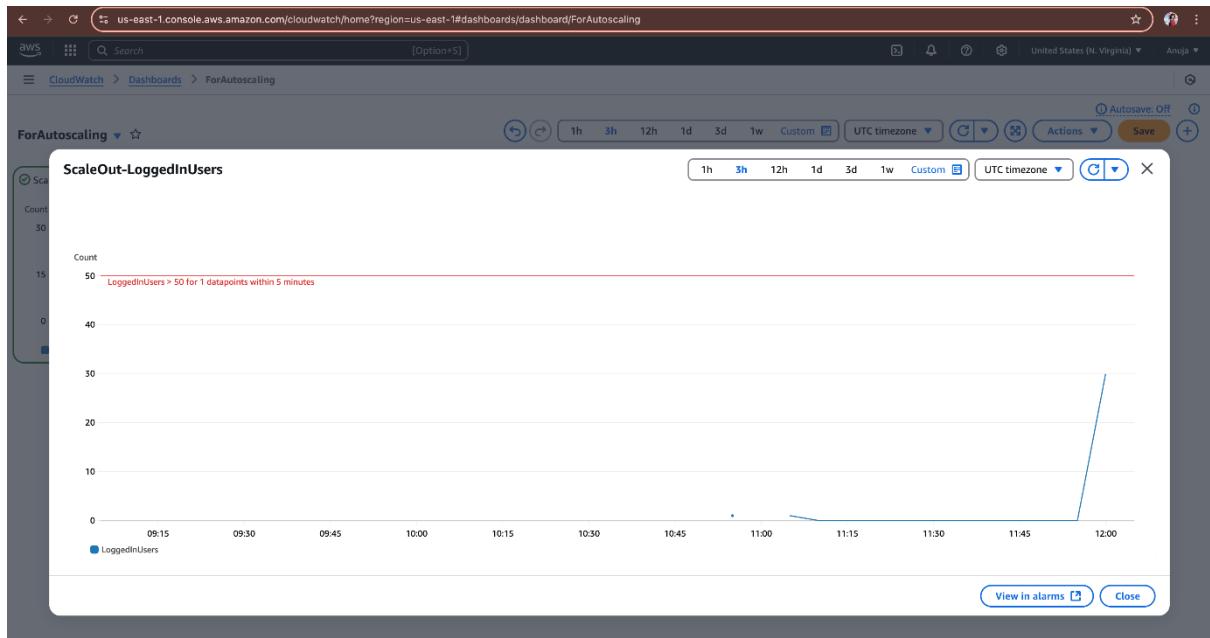
```
Desktop -- ec2-user@ip-172-31-87-219:~ -- ssh -i cloudwatchkeypair.pem ec2-user@ec2-3-87-48-73.compute-1.amazonaws.com -- 94x30
[ec2-user@ip-172-31-87-219 ~]$ chmod +x push-user-metric.sh
[ec2-user@ip-172-31-87-219 ~]$
```

We have created dashboards for autoscaling group.



You can see your dashboards.





You can see one EC2 instance is automatically launched.

