

# WE ARE CREATING AUTOSCALING

## ➤ In Aws Autoscaling is 2 Types

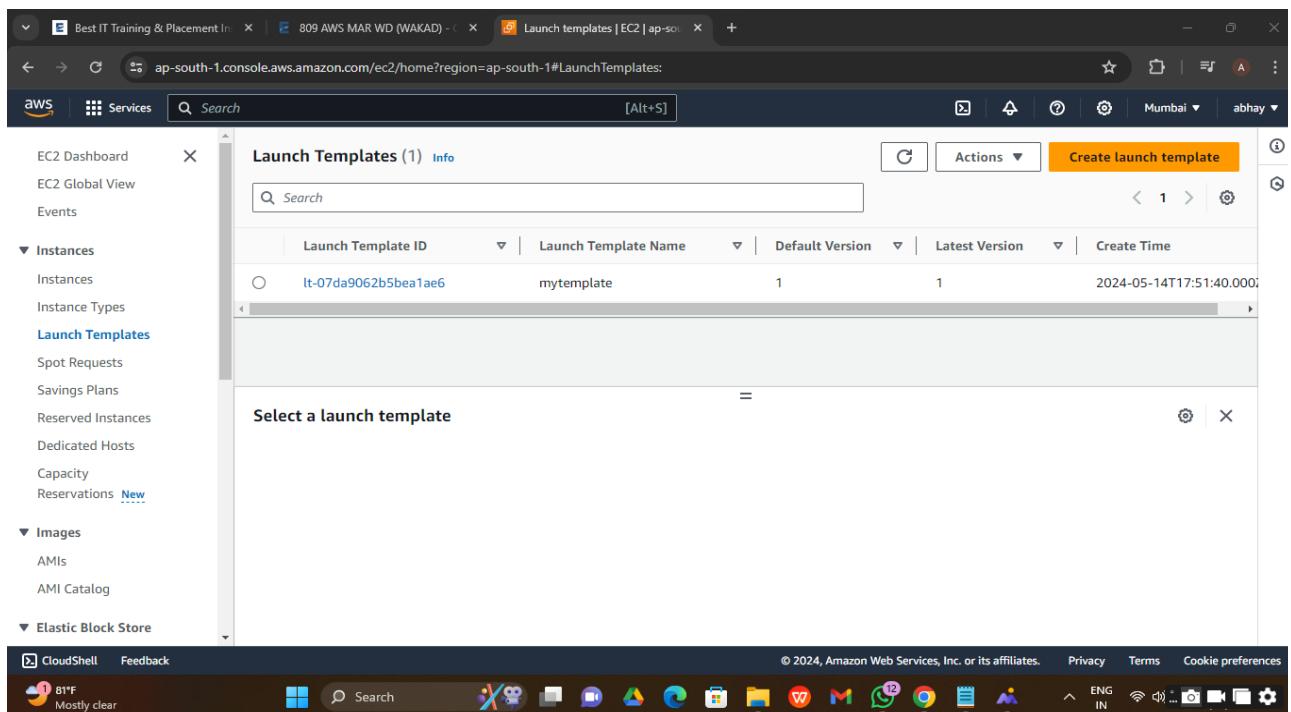
1. Manual
2. Automatic

## ➤ In Automatic is 3 Types

1. Scheduled Actions
2. Dynamic Policies
3. Predictive Policies

## ➤ In Dynamic Policies

1. Targete Scaling Policies
  2. Simple Scaling Policies
  3. Step Scaling Policies
- To implement autoscaling we first need to make configuration ready.
  - When a server will go down and when a new server will come for it.
  - it will have the same configuration, if suppose Linux server is down and next server will come and his same configuration, so system how can understand that time system using present(predefine) configuration.
  - In aws predefine configure is launch templates.
  - Launch templates is storing storing steps of Ec2 when we are creating ec2.
  - goto EC2.
  - goto launch templates--and create launch templates.



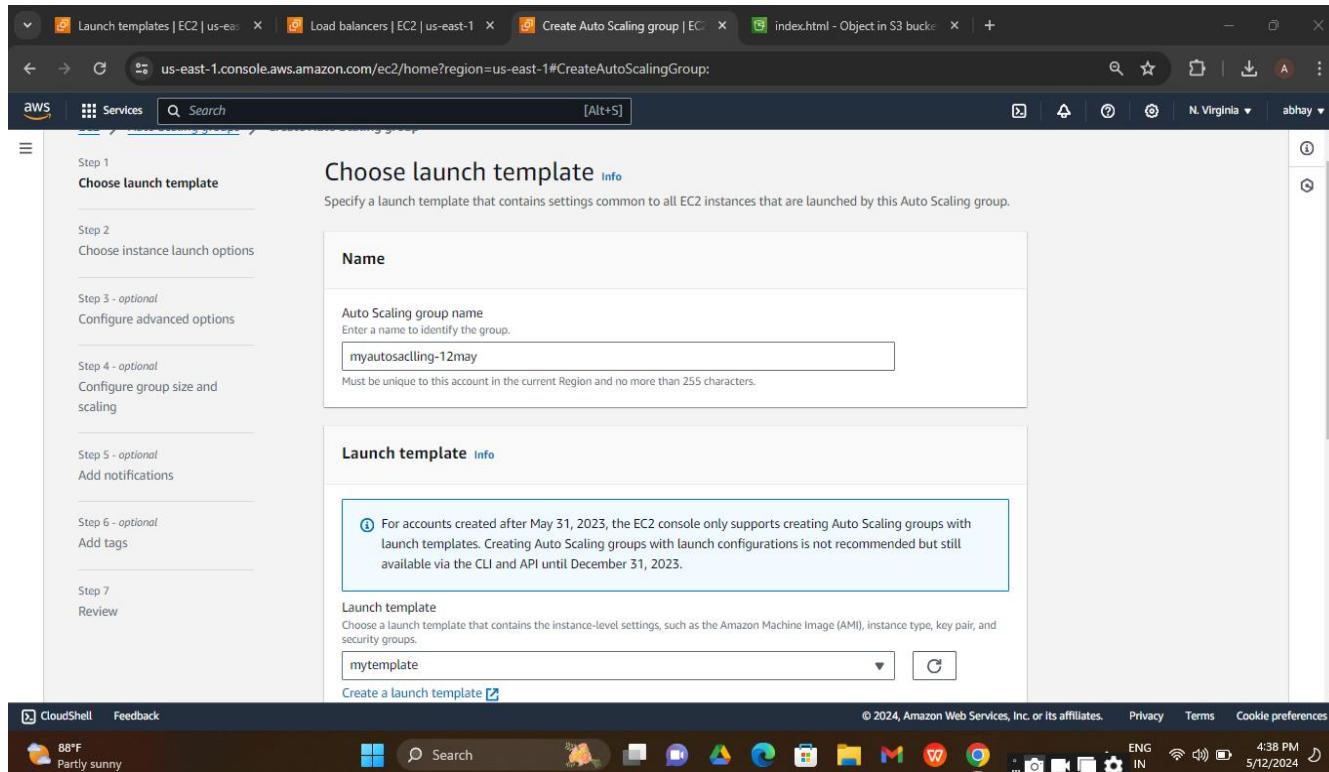
- We creating Classic Load Balancer for autoscaling.
- Classic load balancer is simple load balancer.
- check all Availability Zone because server will go any availability zone.
- same security group as in templates.
- listener and routing HTTP 80.
- health check index.html.
- create load balancer.

The screenshot shows the AWS Cloud Console interface for managing load balancers. The left sidebar navigation includes EC2 Dashboard, EC2 Global View, Events, Console-to-Code (Preview), Instances (with sub-options like Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area is titled "Load balancers (1)". It displays a table with the following data:

Name	DNS name	State	VPC ID	Availability Zones	Type
myclb	myclb-1687412193.us-east-1.elb.amazonaws.com	Active	vpc-06711da3af3b555...	6 Availability Zones	classic

Below the table, a message states "0 load balancers selected" and "Select a load balancer above." The bottom of the screen shows the AWS navigation bar with CloudShell, Feedback, a weather icon (88°F Partly sunny), a search bar, and various AWS service icons (S3, Lambda, CloudWatch, etc.). The status bar at the bottom right indicates the date (5/12/2024), time (4:37 PM), and location (IN, ENG).

- Goto Autoscaling group.
- Create autoscaling.
- We attached screen shot of the next process.



The screenshot shows the AWS CloudShell interface at the bottom, featuring a dark theme with various icons for different services like CloudWatch, Lambda, and S3. The main area displays the AWS Lambda console for the 'us-east-1' region. The URL in the browser bar is 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup'. The page is titled 'Create Auto Scaling group | EC2' and shows the 'Step 2 - optional' configuration section. It includes a dropdown for 'VPC' set to 'vpc-06711da3af3b55566' and a list of subnets from the selected VPC. The subnets listed are:

- us-east-1a | subnet-0b7a8072a4e1501e1 | 172.31.32.0/20 | Default
- us-east-1b | subnet-0da123a304dd8face | 172.31.0.0/20 | Default
- us-east-1c | subnet-0a11676bc73bdbe6f | 172.31.80.0/20 | Default
- us-east-1d | subnet-0498e8a018462dd4a | 172.31.16.0/20 | Default
- us-east-1e | subnet-065755aa653ade756 | 172.31.48.0/20 | Default
- us-east-1f | subnet-050b1096ad5ce566c | 172.31.64.0/20 | Default

Below the subnet list is a 'Create a subnet' button. At the bottom of the page are buttons for 'Cancel', 'Skip to review', 'Previous', and 'Next'.

This screenshot shows the continuation of the 'Create Auto Scaling group' wizard. The URL in the browser bar remains the same: 'us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateAutoScalingGroup'. The page title is now 'Step 4 - optional'.

The left sidebar lists optional steps:

- Configure group size and scaling
- Add notifications
- Add tags
- Review

The main content area is titled 'Configure advanced options' and contains the following sections:

- Step 4 - optional**: 'Configure group size and scaling'
- Step 5 - optional**: 'Add notifications'
- Step 6 - optional**: 'Add tags'
- Step 7**: 'Review'

**Configure advanced options**

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer  
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer  
Choose from your existing load balancers.

Attach to a new load balancer  
Quickly create a basic load balancer to attach to your Auto Scaling group.

**Attach to an existing load balancer**

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups  
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

**Classic Load Balancers**

Select Classic Load Balancers

myclb

Classic Load Balancer

**VPC Lattice integration options**

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

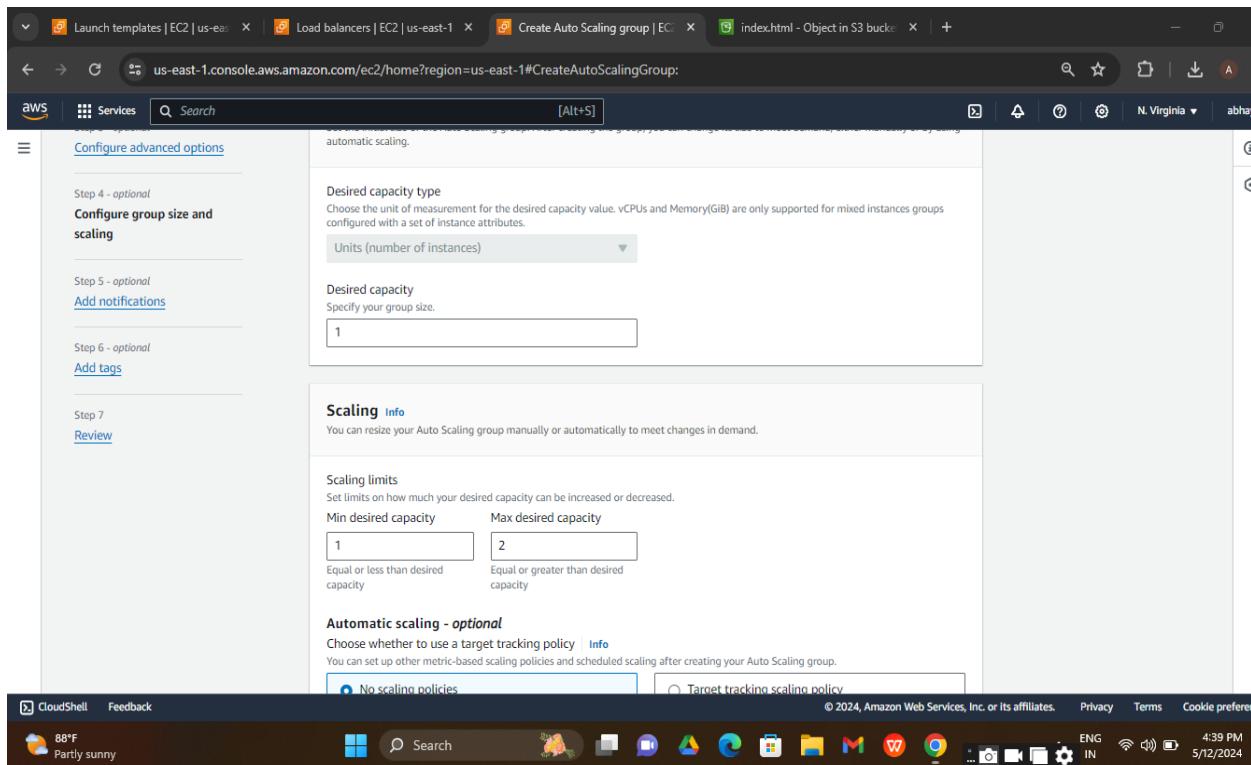
The CloudShell interface at the bottom is identical to the one in the first screenshot.

## ➤ Group size

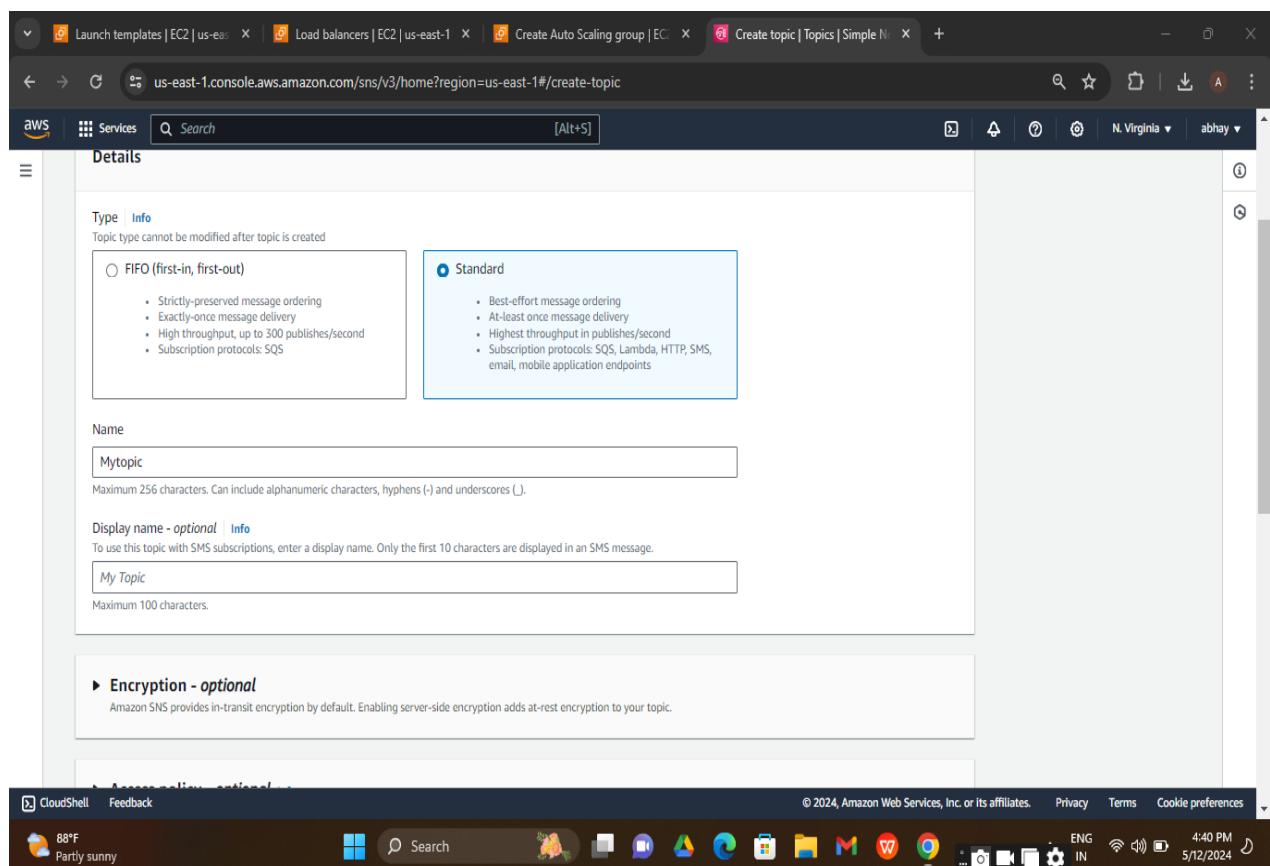
- Coming of servers in load balancer what is desired capacity.
- We decide desired capacity is (1)
- Min desired capacity is (1)
- max desired capacity is (2)
- We cannot scaling in this window because this window scaling policies options is limited.
- When complete this configuration then we are doing the all scaling policies.

## ➤ Instance scale-in protection

- means if my system scaled out for reason is server load is high then system create new server means scaled out. then server load will low.
- So there is options we check this options server will not scale in. and these is not good for cost optimization.
- So we uncheck this option.



- Next step is add notification Goto SNS service. SNS service is sending message from 1 service to another service.
- Means sending message from producer to consumer.
- SNS service also sending the message for you from email and text message also.
- There are 2 ways to sending the messages First is Standard and Second is FIFO.
- In case of standard is --unordered delivery and do not have sure shot delivery.
- Standard delivery is fast compare to FIFO and standard is low cost.
- In case of FIFO is --ordered delivery and sure shot delivery.
- FIFO is slow and High cost.
- We create 1 placeholder.
- Goto topics and create topics.
- We are create topic and subscribe on email

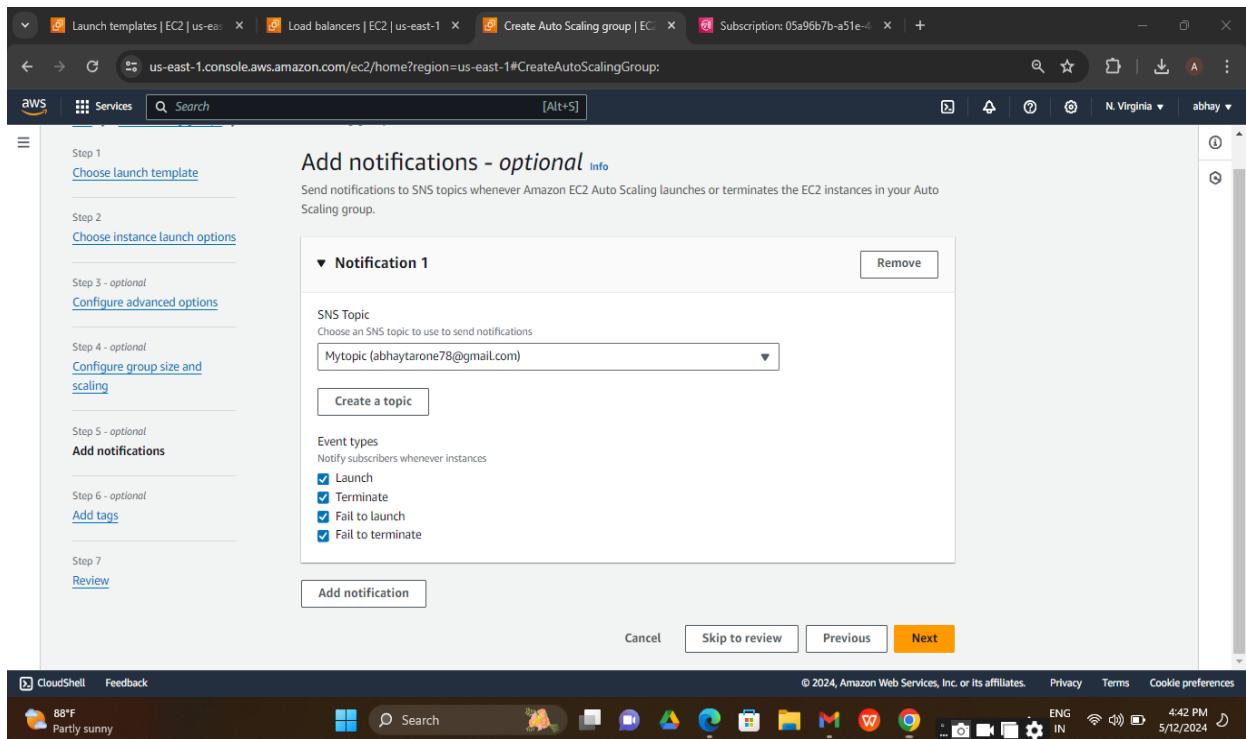


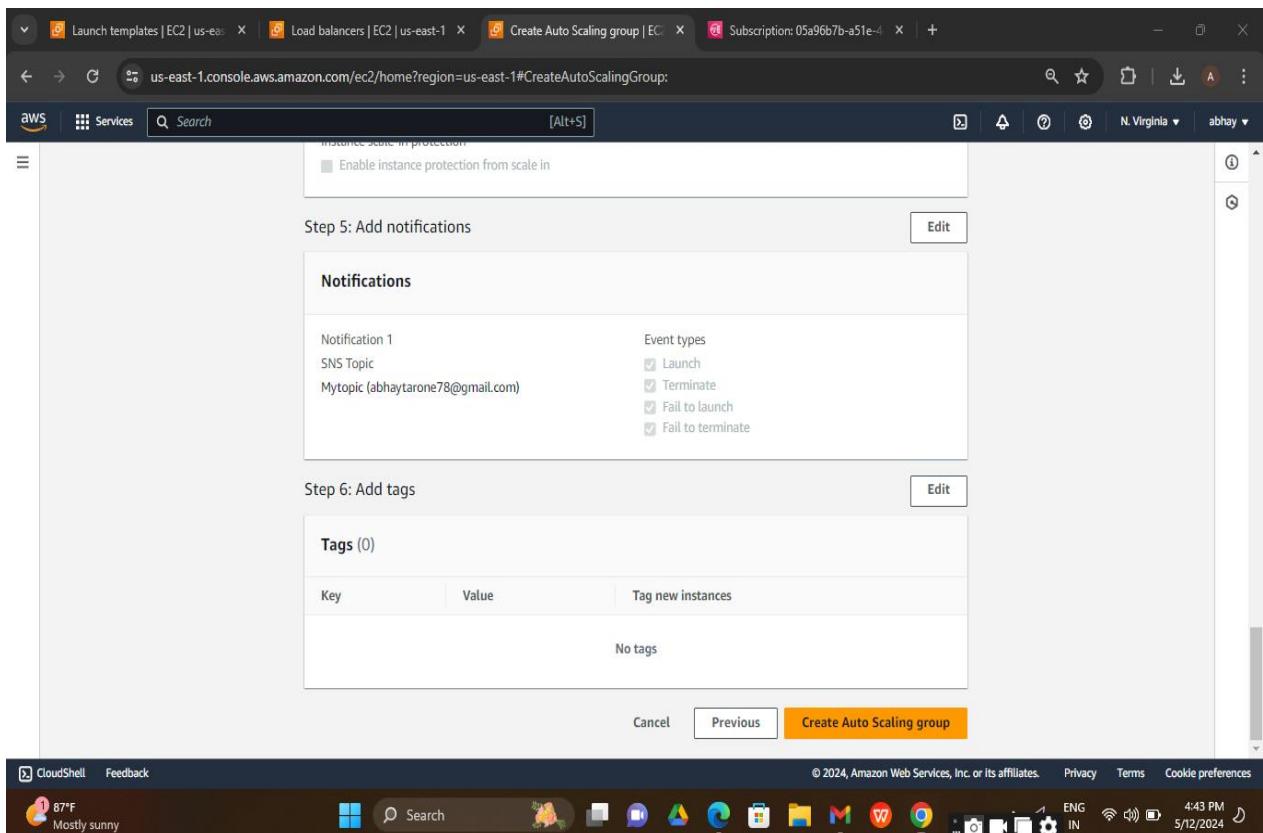
The screenshot shows the 'Create subscription' page in the AWS SNS console. The 'Topic ARN' field contains 'arn:aws:sns:us-east-1:851725280943:Mytopic'. The 'Protocol' dropdown is set to 'Email'. The 'Endpoint' field contains 'abhyatrone78@gmail.com'. A note at the bottom states: 'After your subscription is created, you must confirm it.'

The screenshot shows the 'Subscription: 05a96b7b-a51e-4472-aa95-18b79da01344' details page in the AWS SNS console. The ARN is 'arn:aws:sns:us-east-1:851725280943:Mytopic:05a96b7b-a51e-4472-aa95-18b79da01344'. The status is 'Confirmed'. The protocol is 'EMAIL'. The topic is 'Mytopic'. The subscription principal is 'arn:aws:iam::851725280943:root'. The navigation bar on the left includes 'Dashboard', 'Topics', 'Subscriptions', 'Mobile', 'Push notifications', 'Text messaging (SMS)', and 'Origination numbers'.

## ➤ SNS topic is (my topic)

- I should get a notification-
  1. when my server is launch.
  2. when my server is terminated.
  3. fail to launch or fail to terminate
- I get all this notification on my email.





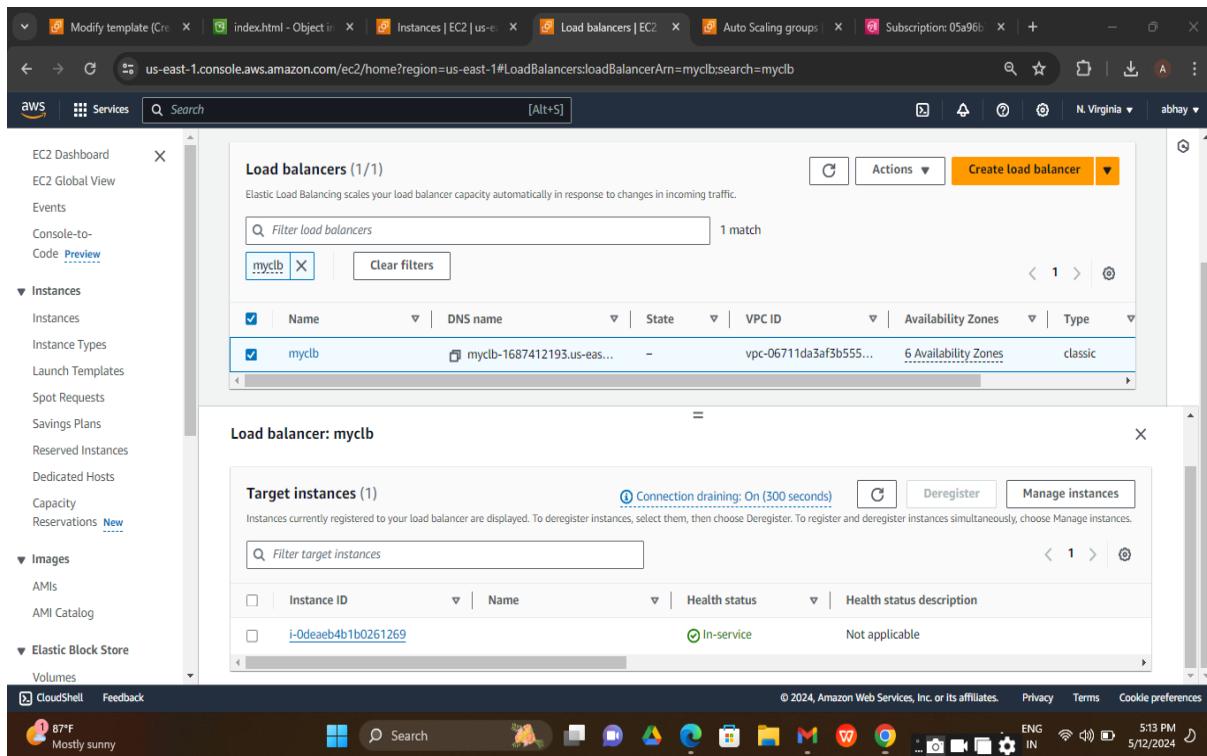
- Autoscaling is created.
- Desired capacity is 1.
- Instance will be 1.

The screenshot shows the AWS Auto Scaling Groups details page. At the top, there are two success messages: "Auto Scaling group updated successfully" and "Myautoscalling, 1 Notification created successfully". Below this, the "Auto Scaling groups (1/1) Info" section displays a table with one row for "Myautoscalling". The table columns include Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Avg. The "Name" column shows "Myautoscalling", "Launch template/configuration" shows "Mytemplate | Version Default", "Instances" shows "1", "Desired capacity" shows "1", and "Min" and "Max" both show "1". The "Avg" column shows "2".

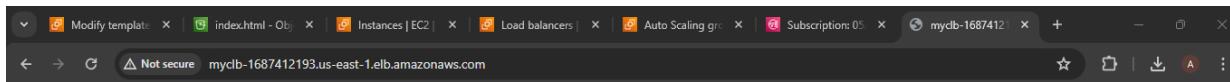
Below the table, the "Auto Scaling group: Myautoscalling" section is expanded. It has tabs for Details, Activity, Automatic scaling, Instance management, Monitoring, and Instance refresh. The Details tab is selected. Under "Group details", the "Auto Scaling group name" is "Myautoscalling", "Desired capacity" is "1", "Desired capacity type" is "Units (number of instances)", and the "Amazon Resource Name (ARN)" is "arn:aws:autoscaling:us-east-1:8517252809:autoScalingGroup:72dd51-77r8-4b0f-a68".

The bottom of the screen shows the Windows taskbar with various pinned icons and system status indicators.

- We are seeing that the server is behind the load balancer using autoscaling.



- Copy loadable DNA and paste browser.
- Now we see our page is resolved from s3.



**hi this webserver resloved from s3**



- WE are increasing capacity manual.
- Select autoscaling goto action and edit.
- we are select desired capacity 2, minimum 1, and maximum 2.

The screenshot shows the AWS CloudWatch Metrics Insights interface. A search bar at the top contains the query: "AWS CloudWatch Metrics Insights". Below the search bar, there are two tabs: "Metrics" and "Logs". Under the "Metrics" tab, there is a table with three columns: "Metric Name", "Dimensions", and "Value". The first row shows "AWS CloudWatch Metrics Insights" with dimensions "Region:us-east-1" and a value of 1. The second row shows "AWS CloudWatch Metrics Insights" with dimensions "Region:us-west-2" and a value of 1. The third row shows "AWS CloudWatch Metrics Insights" with dimensions "Region:us-west-1" and a value of 1. At the bottom of the interface, there is a "Run" button.

The screenshot shows the AWS Management Console interface for editing an Auto Scaling group. The top navigation bar includes tabs for Modify template, index.html - Object in, Instances | EC2 | us-east-1, Load balancers | EC2, Edit Auto Scaling group, and Subscription: 05a96b. The URL in the address bar is us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#EditAutoScalingGroupid=Myautoscaling. The top header bar has tabs for AWS, Services, and Search, along with a [Alt+S] keyboard shortcut. The top right corner shows N. Virginia and a user profile for abhay.

A green success banner at the top states: "Auto Scaling group updated successfully".

The main configuration section starts with "Desired capacity type" which notes that vCPUs and Memory(GiB) are only supported for mixed instance groups. It includes a dropdown menu for "Units (number of instances)" currently set to 2.

The "Desired capacity" section specifies a group size of 2. Below it, the "Scaling limits" section sets both min and max desired capacity to 2, with the note "Equal or less than desired capacity" next to the min value and "Equal or greater than desired capacity" next to the max value.

A "Launch template" section contains a note: "For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023."

The bottom navigation bar includes CloudShell, Feedback, and links for Privacy, Terms, and Cookie preferences. It also shows system status icons for battery, signal, and network, along with the date and time (5/12/2024, 5:14 PM).

- You can see 2 servers are running.

The screenshot shows the AWS EC2 Instances page. The left sidebar navigation includes: EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instances Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations (New), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area displays a table titled "Instances (1/5) Info" with the following data:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
	i-08252d5feb69c662c	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1b	ec2-44-222-2
	i-07b57963eb12e5cdd	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1d	-
<input checked="" type="checkbox"/>	i-Odeaeb4b1b0261269	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1e	ec2-100-25-1
	i-0ab792547709ccaa8	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-
	i-0e9eddab0535f0c87	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-

Below the table, the details for instance **i-Odeaeb4b1b0261269** are shown. The "Details" tab is selected, displaying the following information:

- Instance summary**: Instance ID: i-Odeaeb4b1b0261269, Public IPv4 address: 100.25.158.240, Instance state: Running.
- Public IPv4 address**: 100.25.158.240 | [open address](#)
- Private IPv4 addresses**: 172.31.59.103
- Public IPv4 DNS**: ec2-100-25-158-240.compute-1.amazonaws.com

The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray indicating the date and time (5/12/2024, 5:17 PM).

- Get the public IP of server and login to the server.

The screenshot shows the AWS Management Console interface for the EC2 service. The left sidebar navigation includes 'EC2 Global View', 'Events', 'Console-to-Code Preview', 'Instances' (selected), 'Instance Types', 'Launch Templates', 'Spot Requests', 'Savings Plans', 'Reserved Instances', 'Dedicated Hosts', 'Capacity', and 'Reservations'. The main content area displays the 'Instances (1/5)' list with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 D
	i-08252d5feb69c662c	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1b	ec2-44-222-2
	i-07b57963eb12e5cdd	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1d	-
<input checked="" type="checkbox"/>	i-0deaeb4b1b0261269	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1e	ec2-100-25-1
	i-0ab792547709ccaa8	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-
	i-0e9eddab0535f0c87	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-

Below the list, the details for the selected instance 'i-0deaeb4b1b0261269' are shown:

**i-0deaeb4b1b0261269**

**Details** | Status and alarms [New](#) | Monitoring | Security | Networking | Storage | Tags

**Instance summary** [Info](#)

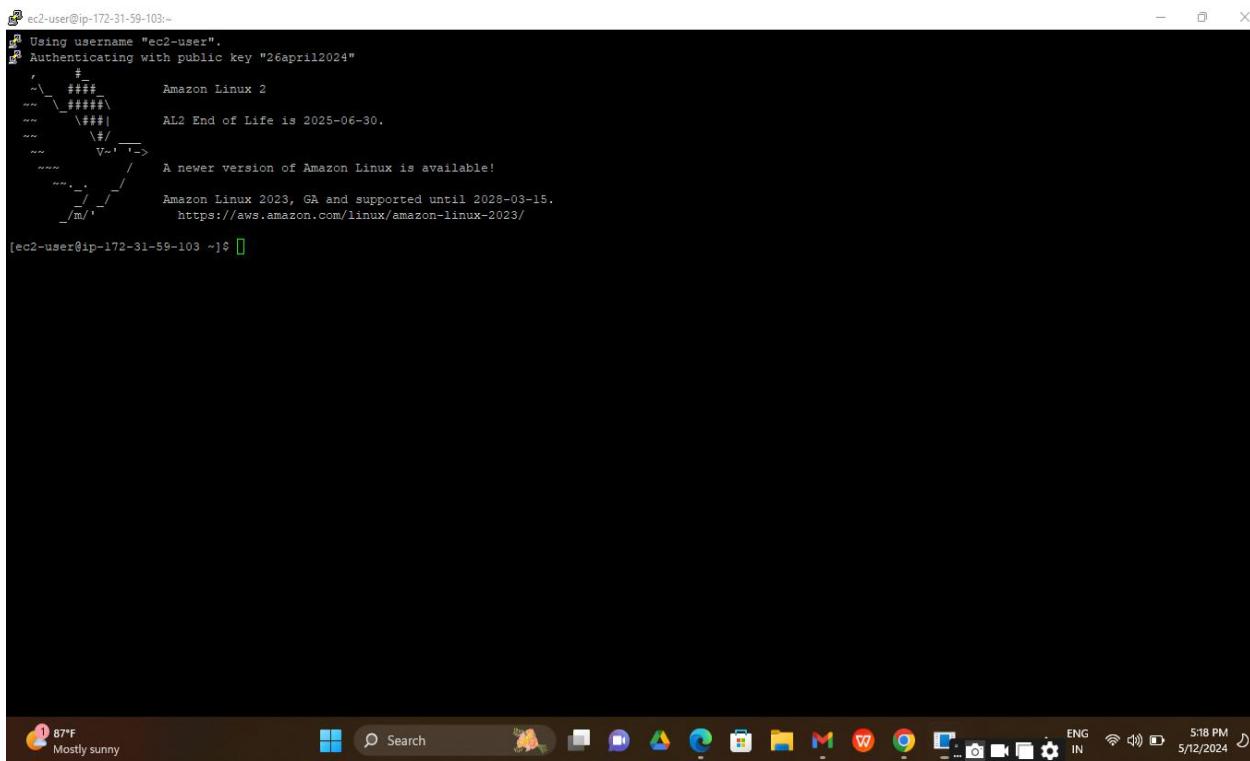
Instance ID i-0deaeb4b1b0261269	Public IPv4 address 100.25.158.240   <a href="#">open address</a>	Private IPv4 addresses 172.31.59.103
IPv6 address -	Instance state Running	Public IPv4 DNS ec2-100-25-158-240.compute-1.amazonaws.com

At the bottom of the screen, there is a taskbar with various icons and system status information.

- **Fire the command-** `cd /var/www/html`  
`sudo vi index.html`
- **insert**

```
</html><h1>hi this is webserver resolved from s3
V1.0</h1></html>
```

  - We fired this command to see.
  - All the server behind the load balancer will segregate the traffic.
  - In the following screenshot we can see that the traffic is

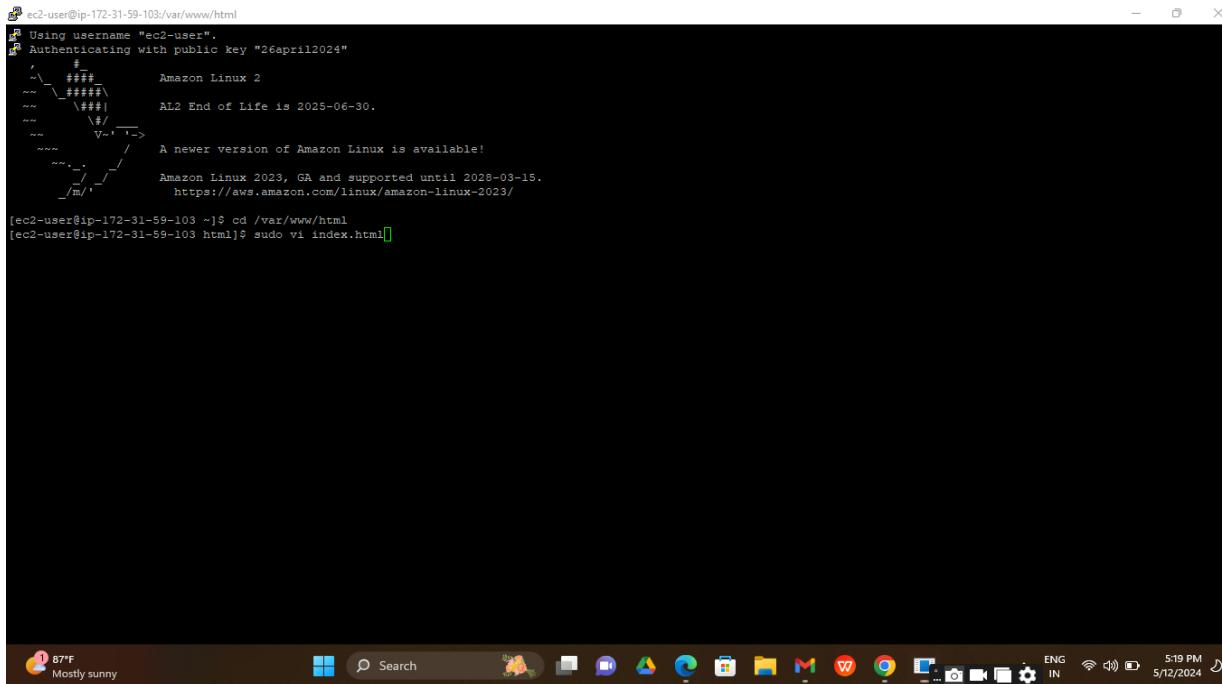


The screenshot shows a terminal window with the following text:

```
ec2-user@ip-172-31-59-103:~$ 
Using username "ec2-user".
Authenticating with public key "26april2024"
          _#_#
         /###\   Amazon Linux 2
        /###\|   AL2 End of Life is 2025-06-30.
         \#/   V-->
          /   A newer version of Amazon Linux is available!
         /   Amazon Linux 2023, GA and supported until 2028-03-15.
         /   https://aws.amazon.com/linux/amazon-linux-2023/
/m/
```

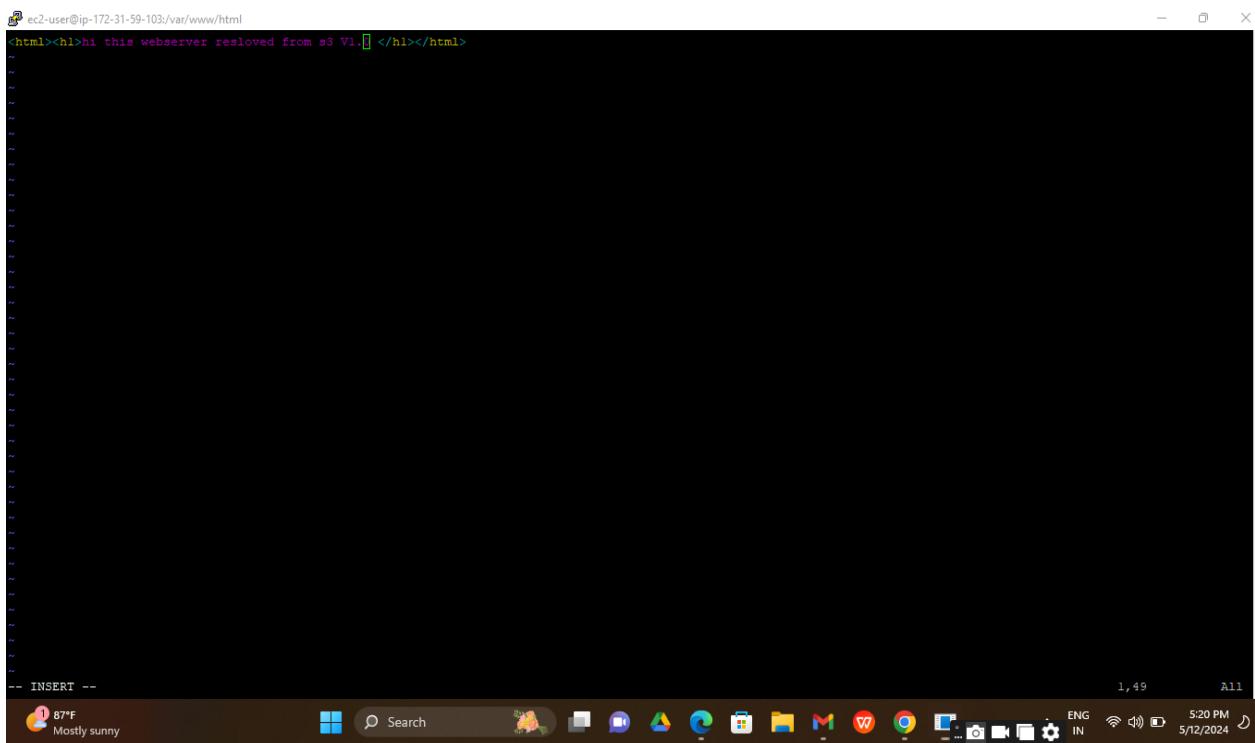
The terminal prompt is `[ec2-user@ip-172-31-59-103 ~]$`. The desktop environment at the bottom includes a weather icon (87°F, Mostly sunny), a search bar, and various application icons.

```
ec2-user@ip-172-31-59-103:~$ cd /var/www/html
[ec2-user@ip-172-31-59-103 ~]$ cat index.html
<html><h1>hi this webserver resolved from s3 V1.</h1></html>
[ec2-user@ip-172-31-59-103 ~]$ sudo vi index.html
```

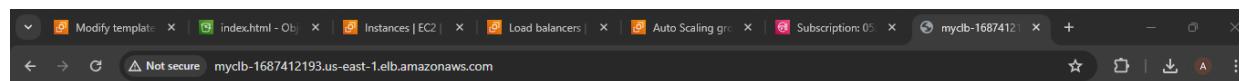
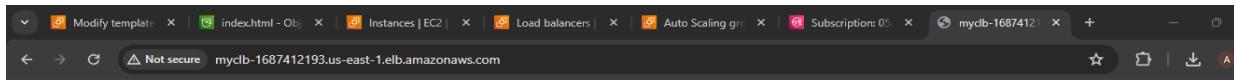


This screenshot shows a Windows desktop environment with a terminal window open. The terminal window displays a message about the end-of-life of Amazon Linux 2, followed by a command to edit a file. Below the terminal is a file editor window showing the same HTML content. The desktop taskbar at the bottom includes icons for various applications like File Explorer, Google Chrome, and Microsoft Word.

```
ec2-user@ip-172-31-59-103:~$ cd /var/www/html
[ec2-user@ip-172-31-59-103 ~]$ cat index.html
<html><h1>hi this webserver resolved from s3 V1.</h1></html>
```



This screenshot shows a Windows desktop environment with a terminal window open, displaying the same HTML content as the previous screenshot. A file editor window is also present. The desktop taskbar at the bottom is identical to the first screenshot, showing various application icons.



➤ In automatic scaling-

- We are going to scheduled action.
- Create scheduled action.
- Name -Rule 01
- Desired capacity -2
- Min -1
- maxi-2
- Recurrence- Once
- Time zone-- Asia/Kolkata
- Specific start time - 2024/05/12
- 17:26 (Asia/Kolkata)

Create.

The screenshot shows the AWS EC2 Auto Scaling Groups page. The left sidebar navigation includes: EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances (selected), Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity Reservations (New), Images (AMIs, AMI Catalog), and Elastic Block Store (Volumes). The main content area displays the 'Auto Scaling groups (1/1)' section. A table lists one group: 'Myautoscalling' (Mytemplate | Version Default) with 1 instance, a status of -, a desired capacity of 1, and a minimum of 1. Below this, a modal window titled 'Auto Scaling group: Myautoscalling' shows a 'Scheduled actions (0)' section with a 'Create scheduled action' button. The bottom of the screen shows the Windows taskbar with various pinned icons and the system tray.

The screenshot shows the AWS Management Console interface for creating a scheduled action within an Auto Scaling group. The left sidebar navigation includes 'EC2 Dashboard', 'EC2 Global View', 'Events', 'Console-to-Code Preview', 'Instances' (selected), 'Images', and 'Elastic Block Store'. The main content area is titled 'Auto Scaling group' and shows a configuration dialog for a new scheduled action named 'Rule 01'. The dialog fields include:

- Name:** Rule 01
- Desired capacity:** 2
- Min:** 1
- Max:** 2
- Recurrence:** Once
- Time zone:** Asia/Kolkata
- Specific start time:** 2024/05/12 at 17:26 (Asia/Kolkata)

At the bottom right of the dialog are 'Cancel' and 'Create' buttons. The status bar at the bottom of the screen displays the date and time as '5/12/2024 5:24 PM'.

- Now we can see that the instance has become two.

Scheduled action created or edited successfully

[EC2](#) > Auto Scaling groups

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<a href="#">Myautoscaling</a>	<a href="#">Mytemplate</a>   Version Default	2	-	2	1

0 Auto Scaling groups selected

Select an Auto Scaling group

CloudShell Feedback

87°F Mostly sunny

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

ENG IN 5:27 PM 5/12/2024

- 2 servers now created and running state.

The screenshot shows the AWS EC2 Instances page with the following details:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
	i-08252d5feb69c662c	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a> +	us-east-1b	ec2-44-222-2
	i-0cdc8428e17a4481a	Running	t2.micro	Initializing	<a href="#">View alarms</a> +	us-east-1d	ec2-100-27-2
	i-07b57963eb12e5cdd	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1d	-
<input checked="" type="checkbox"/>	i-0deaeb4b1b0261269	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-
	i-0ab792547709ccaa8	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-
	i-0e9eddab0535fc87	Terminated	t2.micro	-	<a href="#">View alarms</a> +	us-east-1e	-

Details for the selected instance (i-0deaeb4b1b0261269):

- Instance summary:
 

Instance ID: i-0deaeb4b1b0261269	Public IPv4 address: -	Private IPv4 addresses: -
IPv6 address: -	Instance state: Terminated	Public IPv4 DNS: -

- 2 Server is behind load balancer.

The screenshot shows the AWS EC2 Load Balancers console. The main pane displays a table titled "Load balancers (1/1)" with one entry: "myclb". The details for "myclb" are shown in the table below:

Name	DNS name	State	VPC ID	Availability Zones	Type
myclb	myclb-1687412193.us-eas...	-	vpc-06711da3af3b555...	6 Availability Zones	classic

Below this, a modal window titled "Load balancer: myclb" shows the registered instances:

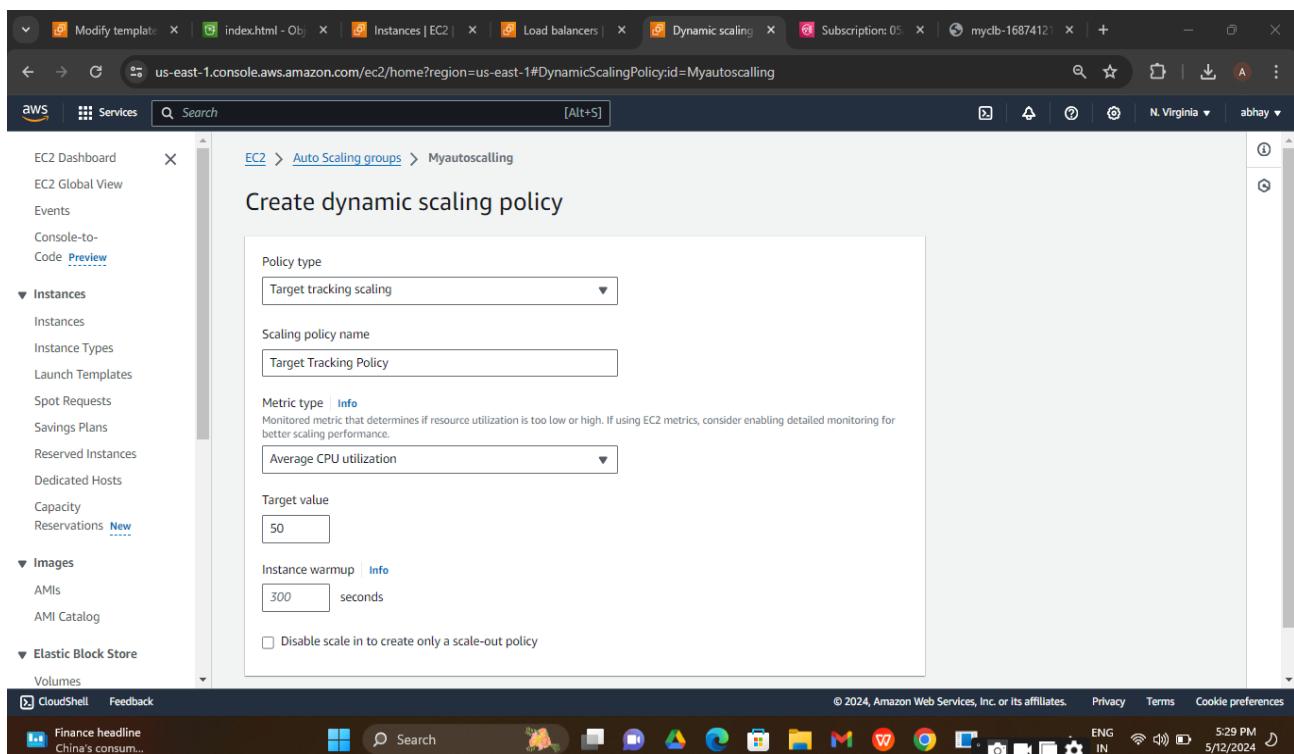
Instance ID	Name	Health status	Health status descri...	Security groups
i-08252d5feb69c662c		<span>In-service</span>	Not applicable	launch-wizard-2
i-0cdc8428e17a4481a		<span>In-service</span>	Not applicable	launch-wizard-2

➤ **We are going to dynamic scaling policies**

- In dynamic scaling policies 3 types.
  1. Target scaling policies
  2. Simple scaling policies
  3. Step scaling policies
- We go to the simple scaling policies.
- We do simple scaling policy so we have to go to CloudWatch service. CloudWatch is doing our odd jobs.
- We go to ---> alarm --create alarms--select metric- Ec2 -- By autoscaling group-- select Autoscaling-12may CPUUtilization (I want to see the capacity) --select matrix-- If utilization period of time 1 minute is going to higher than 60% then get notification of my email using(mytopic).

➤ **Add name and description**

- Alarm name (Alarm 01) --next--create alarm.
- We create alarm.
- Go back to the simple scaling policies.
- Policy type- simple scaling policies.
- Scaling policy name-(Rule 01).
- Alarm 01.
- Take the action.
- Add 1 server capacity unit.
- create.



The screenshot shows the AWS CloudWatch Alarms page. The left sidebar is collapsed, and the main area displays a table titled "Alarms (0)". The table has columns for Name, State, Last state update (UTC), Conditions, and Actions. A message at the top right says "No alarms" and "No alarms to display". Below this is a link "Read more about Alarms" and a large orange "Create alarm" button.

The screenshot shows the "Select metric" dialog box as part of the "Create alarm" wizard. The dialog is titled "Step 1: Specify" and has tabs for Step 2: Configure, Step 3: Add name, and Step 4: Preview. The "Configure" tab is selected. The dialog shows a list of metrics under "Graphed metrics (1)". One metric, "CPUUtilization", is selected and highlighted with a blue border. Other metrics listed include "DiskReadBytes", "NetworkOut", "NetworkIn", "CPUUtilization" (selected), "NetworkPacketsOut", and "DiskReadOps". At the bottom right of the dialog are "Cancel" and "Select metric" buttons.

Screenshot of the AWS CloudWatch Metrics Alarm creation wizard - Step 2: Configure actions.

The "Metric" section shows a graph of CPUUtilization over time (09:30 to 11:30). A blue line represents the metric value, which crosses a red threshold line at approximately 0.367. The graph indicates that the alarm will trigger when the blue line goes above the red line for 1 datapoint within 1 minute.

**Metric**

Graph: This alarm will trigger when the blue line goes above the red line for 1 datapoint within 1 minute.

Percent

0.367

0.35

0.334

09:30 10:30 11:30

CPUUtilization

Namespace: AWS/EC2

Metric name: CPUUtilization

AutoScalingGroupName: myautoscaling-12may

Statistic: Average

Period: 1 minute

**Conditions**

Threshold type: 1 minute

Screenshot of the AWS CloudWatch Metrics Alarm creation wizard - Step 3: Add name and description.

The "Conditions" section is displayed. The "Threshold type" is set to "Static" (radio button selected). The condition is defined as "Whenever CPUUtilization is... Greater > threshold" with a value of "60".

**Conditions**

Threshold type: 1 minute

Whenever CPUUtilization is...

Define the alarm condition.

Static  
Use a value as a threshold

Anomaly detection  
Use a band as a threshold

Greater  
> threshold

Greater/Equal  
>= threshold

Lower/Equal  
<= threshold

Lower  
< threshold

than...

Define the threshold value.

60

Must be a number

**Additional configuration**

Cancel Next

Screenshot of the AWS CloudWatch Metrics console showing the 'Create alarm' wizard - Step 2: Configure actions.

The 'Notification' section is displayed, showing the 'Alarm state trigger' configuration. It includes three options:

- In alarm: The metric or expression is outside of the defined threshold.
- OK: The metric or expression is within the defined threshold.
- Insufficient data: The alarm has just started or not enough data is available.

Below this, there are sections for sending notifications to an SNS topic ('Mytopic') and an email endpoint ('abhyatrone78@gmail.com').

At the bottom right of the main panel, there is a 'Remove' button.

The navigation sidebar on the left shows steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create).

The browser status bar at the bottom shows: CloudShell Feedback, 86°F Mostly sunny, Search, various icons, ENG IN, 5:35 PM, 5/12/2024.

Screenshot of the AWS CloudWatch Metrics console showing the 'Create alarm' wizard - Step 3: Add name and description.

The 'Name and description' section is displayed, containing the following fields:

- Alarm name:** alarm 01
- 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)

A note at the bottom states: "Markdown formatting is only applied when viewing your alarm in the console. The description will remain in plain text in the alarm notifications."

At the bottom right, there are 'Cancel', 'Previous', and 'Next' buttons.

The navigation sidebar on the left shows steps: Step 1 (Specify metric and conditions), Step 2 (Configure actions), Step 3 (Add name and description), and Step 4 (Preview and create).

The browser status bar at the bottom shows: CloudShell Feedback, 86°F Mostly sunny, Search, various icons, ENG IN, 5:36 PM, 5/12/2024.

us-east-1.console.aws.amazon.com/cloudwatch/home?region=us-east-1#alarmsV2:create?~(Page~>Preview~AlarmType~MetricAlarm~AlarmData~(Name...)

Whenever CPUUtilization is Greater (>) than... 60

▶ Additional configuration

**Step 2: Configure actions**

**Actions**

Notification  
When In alarm, send a notification to "Mytopic"

**Step 3: Add name and description**

**Name and description**

Name  
alarm 01

Description

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 86°F Mostly sunny 5:37 PM 5/12/2024 ENG IN

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

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

**Instances**

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

**Images**

- AMIs
- AMI Catalog

**Elastic Block Store**

- Volumes

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 86°F Mostly sunny 5:38 PM 5/12/2024 ENG IN

**Policy type**  
Simple scaling

**Scaling policy name**  
Rule 01

**CloudWatch alarm**  
Choose an alarm that can scale capacity whenever:  
alarm 01 [Create a CloudWatch alarm](#)  
breaches the alarm threshold: CPUUtilization > 60 for 1 consecutive periods of 60 seconds for the metric dimensions:  
AutoScalingGroupName = myautoscaling-12may

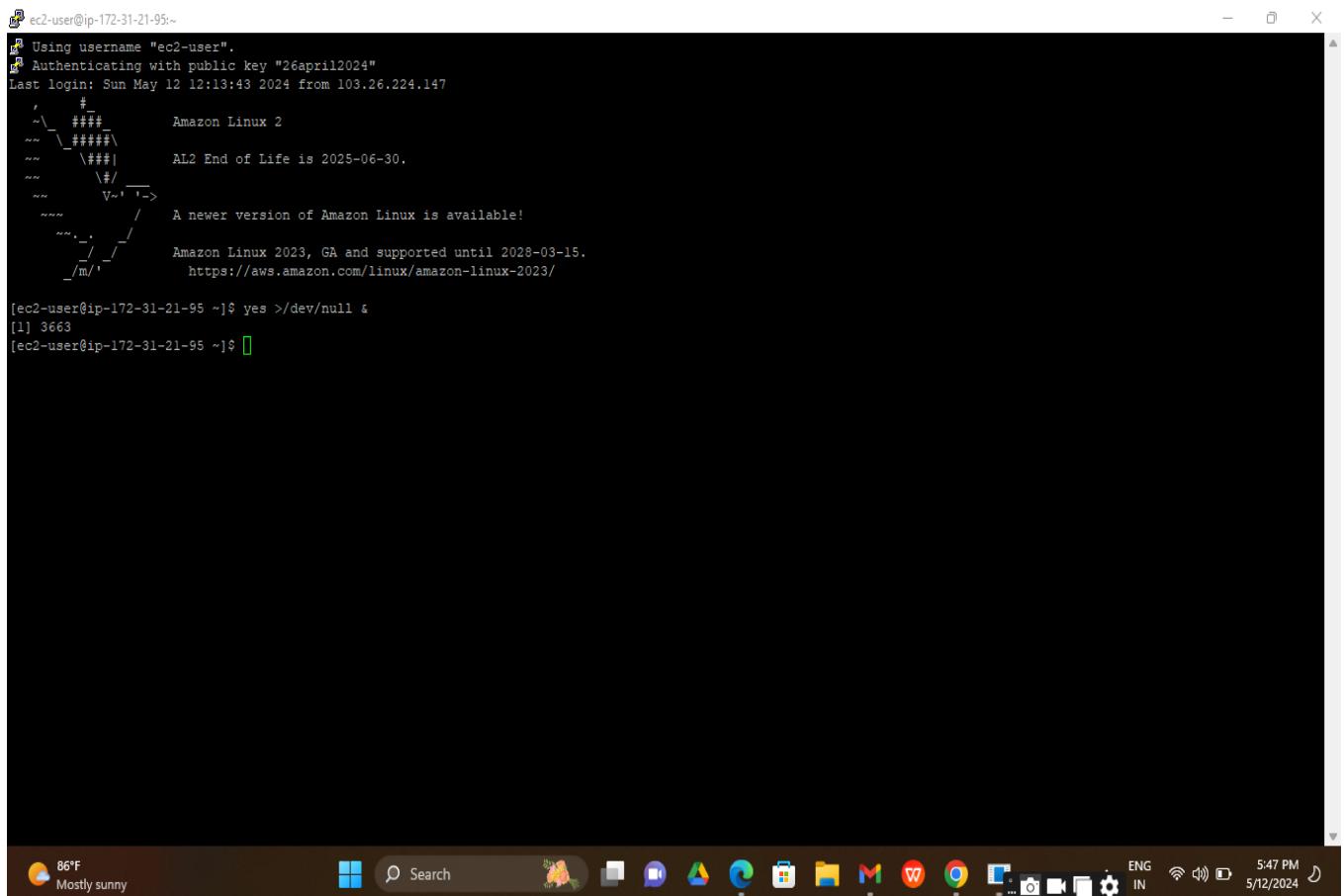
**Take the action**  
Add 1 capacity units

**And then wait**  
300 seconds before allowing another scaling activity

Create

➤ **Login to server.**

- We had said that if our server utilization goes up to 60% so new server should be created.
- We can increase server utilization for using this command- yes >/dev/null &
- Meaning of this command is - yes is loop and we running this loop continuously.
- Our CPU in consume
- We never do that in production



A screenshot of a Windows desktop environment. In the center is a terminal window titled 'cmd' with the command 'yes >/dev/null &'. The terminal output shows the user logging in as 'ec2-user' on an Amazon Linux 2 system, which is end-of-life as of June 30, 2025. It also mentions a newer version of Amazon Linux available and links to the Amazon Linux 2023 documentation. The taskbar at the bottom shows various pinned icons including File Explorer, Microsoft Edge, and several Microsoft Office applications like Word, Excel, and PowerPoint. The system tray shows the date as 5/12/2024, the time as 5:47 PM, and battery status.

```
ec2-user@ip-172-31-21-95:~  
Using username "ec2-user".  
Authenticating with public key "26april2024"  
Last login: Sun May 12 12:13:43 2024 from 103.26.224.147  
      _#_  
     /###/  
    /####\ Amazon Linux 2  
   /##| AL2 End of Life is 2025-06-30.  
  /#/  
 / V~! -->  
 / A newer version of Amazon Linux is available!  
 / /  
 /m/ Amazon Linux 2023, GA and supported until 2028-03-15.  
 https://aws.amazon.com/linux/amazon-linux-2023/  
[ec2-user@ip-172-31-21-95 ~]$ yes >/dev/null &  
[1] 3663  
[ec2-user@ip-172-31-21-95 ~]$
```

```

ec2-user@ip-172-31-21-95:~ 
Using username "ec2-user".
Authenticating with public key "26april2024"
Last login: Sun May 12 12:10:49 2024 from 103.26.224.147
  _#_
 /###\ Amazon Linux 2
 /##\ AL2 End of Life is 2025-06-30.
 \#/
 \# V- ' ->
 / A newer version of Amazon Linux is available!
 /m/ Amazon Linux 2023, GA and supported until 2028-03-15.
 https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-21-95 ~]$ yes >/dev/null &
[1] 3663
[ec2-user@ip-172-31-21-95 ~]$ top

```

```

ec2-user@ip-172-31-21-95:~ 
top - 12:19:08 up 22 min, 1 user, load average: 0.80, 0.39, 0.16
Tasks: 108 total, 2 running, 64 sleeping, 0 stopped, 0 zombie
%Cpu(s): 98.3 us, 1.3 sy, 0.0 ni, 0.0 id, 0.0 wa, 0.0 hi, 0.3 si, 0.0 st
KiB Mem : 975572 total, 257344 free, 93896 used, 624332 buff/cache
KiB Swap: 0 total, 0 free, 0 used. 741516 avail Mem

      PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
 3663 ec2-user  20   0 114648  728  668 R 99.3  0.1 1:18.06 yes
 3664 ec2-user  20   0 168948 4288 3736 R  0.3  0.4 0:00.01 top
  1 root    20   0 123708  5672 3976 S  0.0  0.6 0:02.20 systemd
  2 root    20   0     0     0  0 S  0.0  0.0 0:00.00 kthreadd
  3 root    0 -20    0     0  0 I  0.0  0.0 0:00.00 rcu_gp
  4 root    0 -20    0     0  0 I  0.0  0.0 0:00.00 rcu_par_gp
  6 root    0 -20    0     0  0 I  0.0  0.0 0:00.00 kworker/7:0H-ev
  7 root    20   0     0     0  0 I  0.0  0.0 0:00.02 kworker/u30:0-x
  8 root    0 -20    0     0  0 I  0.0  0.0 0:00.00 mm_percpu_wq
  9 root    20   0     0     0  0 S  0.0  0.0 0:00.00 rcu_tasks_rude_
 10 root   20   0     0     0  0 S  0.0  0.0 0:00.00 rcu_tasks_trace_
 11 root   20   0     0     0  0 S  0.0  0.0 0:00.03 ksoftirqd/0
 12 root   20   0     0     0  0 I  0.0  0.0 0:00.12 rcu_sched
 13 root   rt   0     0     0  0 S  0.0  0.0 0:00.00 migration/0
 15 root   20   0     0     0  0 S  0.0  0.0 0:00.00 cpuhp/0
 17 root   20   0     0     0  0 S  0.0  0.0 0:00.00 kdevtmpfs
 18 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 netns
 21 root   20   0     0     0  0 S  0.0  0.0 0:00.01 kaudited
 299 root   20   0     0     0  0 S  0.0  0.0 0:00.00 khungtaskd
300 root   20   0     0     0  0 S  0.0  0.0 0:00.00 com_reaper
301 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 writeback
303 root   20   0     0     0  0 S  0.0  0.0 0:00.02 kcompactd0
304 root   25   5     0     0  0 S  0.0  0.0 0:00.00 ksmd
305 root   39  19     0     0  0 S  0.0  0.0 0:00.00 khugepaged
361 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 integrityd
363 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 kblockd
364 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 blkcg_punt_bio
716 root   20   0     0     0  0 S  0.0  0.0 0:00.00 xen-balloon
722 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 tpm_dev_wq
728 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 md
731 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 edac-poller
736 root   -51   0     0     0  0 S  0.0  0.0 0:00.00 watchdogd
834 root   0 -20    0     0  0 I  0.0  0.0 0:00.05 kworker/0:1H-kb
886 root   20   0     0     0  0 S  0.0  0.0 0:00.01 kswapd0
888 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 xfsalloc
889 root   0 -20    0     0  0 I  0.0  0.0 0:00.00 xfs_mru_cache

```

- We can see that instance count is 2.
- Now simple scaling policies is done.

The screenshot shows a browser window for the AWS Management Console at [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#AutoScalingGroups). The left sidebar is collapsed. The main content area displays a green success message: "Auto Scaling group updated successfully". Below it, the "Auto Scaling groups" section shows a table with one item:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
<a href="#">Myautoscalling</a>	<a href="#">Mytemplate</a>   Version Default	2	-	2	1

Below the table, a section titled "0 Auto Scaling groups selected" with the sub-instruction "Select an Auto Scaling group" is visible. The bottom of the screen shows the standard AWS navigation bar with links like CloudShell, Feedback, and various service icons.

The screenshot shows the AWS EC2 Instances page with the following details:

**Instances (4) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
	i-08252d5feb69c662c	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1b	-
	i-0cdc8428e17a4481a	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1d	ec2-100-
	i-0deaeb4b1b0261269	Terminated	t2.micro	-	<a href="#">View alarms</a>	us-east-1e	-
	i-04cd21ae972ec646c	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1e	ec2-34-

**Select an instance**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 86°F Mostly sunny Search ENG IN 6:06 PM 5/12/2024

