

AWS AUTOSCALING

What is Autoscaling Groups in AWS?

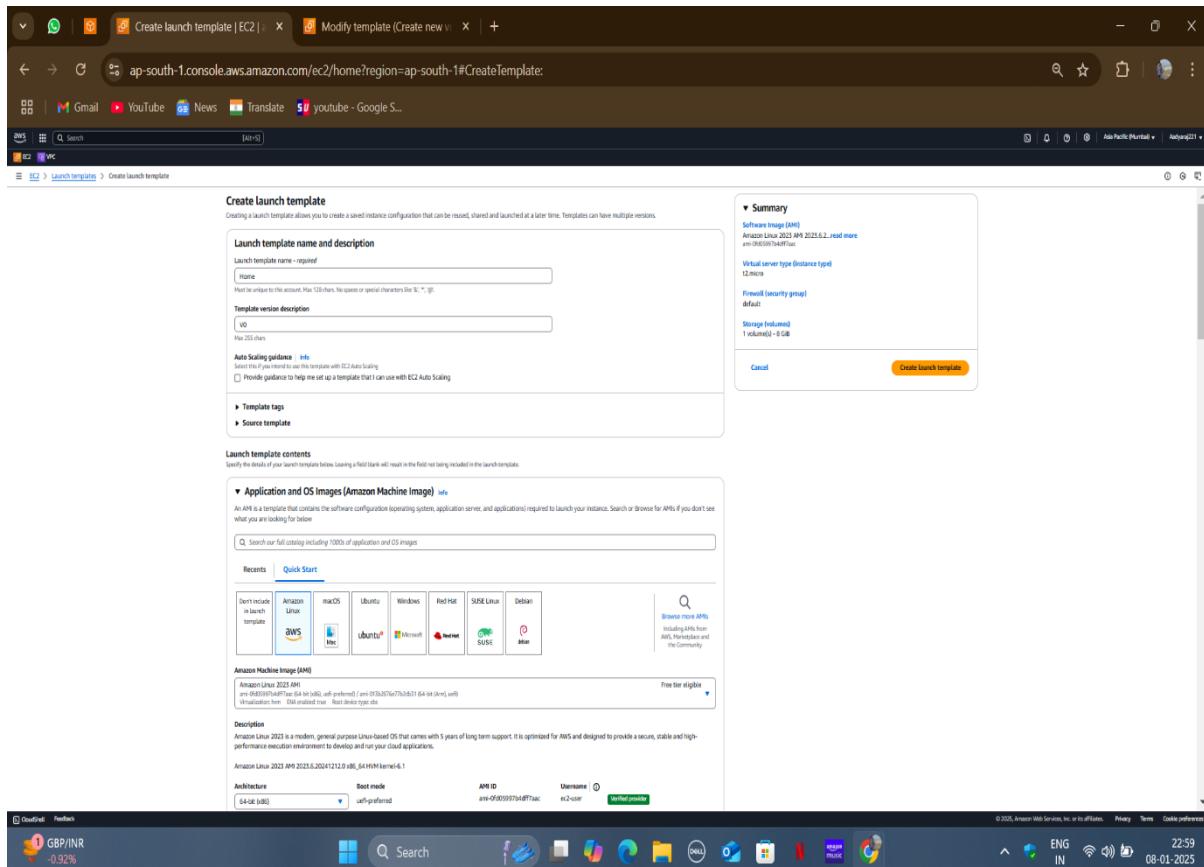
AWS Auto Scaling Group's is an automated way to horizontally scale up your instances with the help of templates, ASG will scale up instances on the basis of the incoming traffic, while it will use the configurations of templates to create multiple copies.

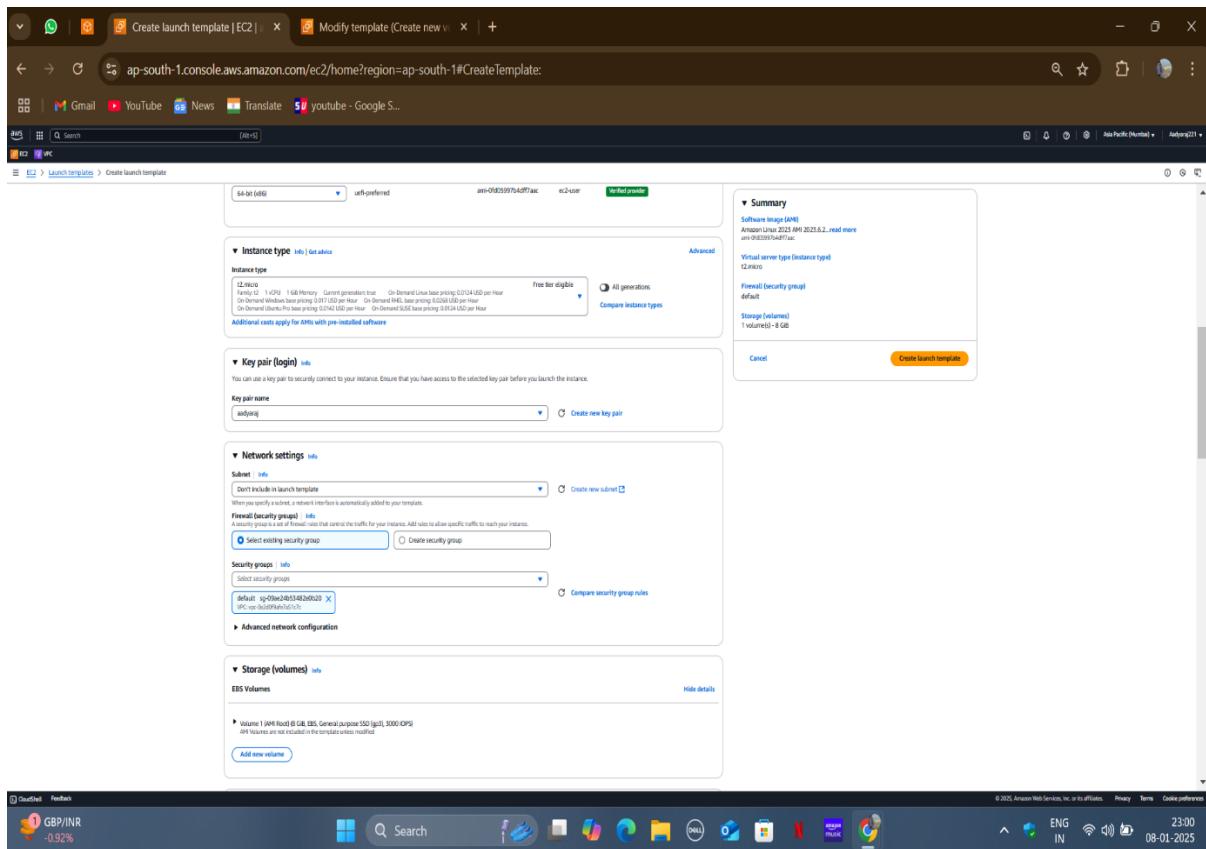
Autoscaling types:

- Dynamic Scaling:** Automatically adjust capacity based on traffic patterns.
- Predictive Scaling:** Anticipate and prepare for future traffic using machine learning models.
- Reactive Scaling:** As workloads increase, the system responds by bringing more servers online.
- Scheduled Scaling:** Adjusts or scales resources based on schedule.

How Auto Scaling Works:

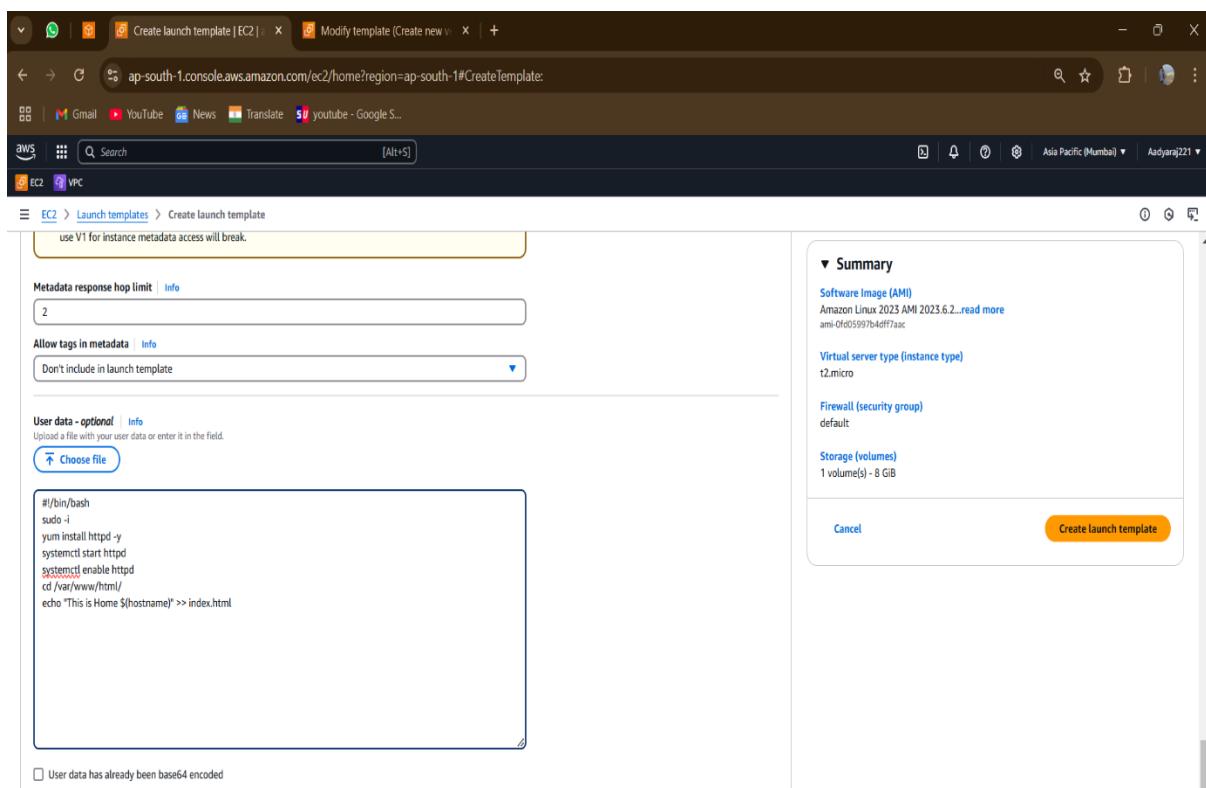
Step1: create 1st Home template-





The screenshot shows the AWS Cloud9 IDE interface. The code editor contains the following Python script:

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html/
echo "This is Home ${hostname}" >> index.html
```



The screenshot shows the AWS Cloud9 IDE interface. The code editor contains the same Python script as the previous screenshot.



The screenshot shows the AWS Cloud9 IDE interface. The code editor contains the same Python script as the previous screenshots.

Step2: create 2nd Laptop template-

The screenshot shows the AWS Lambda console with the following details:

- Function name:** HelloWorld
- Runtime:** Python 3.8
- Description:** A simple Lambda function that prints "Hello World" to the CloudWatch logs.
- Code entry type:** Lambda@Edge
- Code:** A sample Lambda function code in Python.
- Test:** A CloudWatch Logs test log group is selected.
- Logs:** Shows the execution log for the function.
- Metrics:** Shows CloudWatch Metrics data for the function.
- Deployment:** Shows deployment history and metrics.

The screenshot shows the AWS Lambda console with the following details:

- Function name:** HelloWorld
- Runtime:** Python 3.8
- Description:** A simple Lambda function that prints "Hello World" to the CloudWatch logs.
- Code entry type:** Lambda@Edge
- Code:** A sample Lambda function code in Python.
- Test:** A CloudWatch Logs test log group is selected.
- Logs:** Shows the execution log for the function.
- Metrics:** Shows CloudWatch Metrics data for the function.
- Deployment:** Shows deployment history and metrics.

The screenshot shows the AWS CloudShell interface at the bottom, displaying various AWS services like Lambda, S3, and CloudWatch. The main browser window is titled "Create launch template | EC2" and shows the "Create new v..." step. It includes sections for "User data - optional" containing a shell script to start an Apache server, and a "Summary" section with details like Software Image (AMI), Virtual server type (t2.micro), and Storage (volumes). A "Create launch template" button is visible.

Step3: create 3rd Mobile template-

The screenshot shows the AWS CloudShell interface at the bottom, displaying various AWS services like Lambda, S3, and CloudWatch. The main browser window is titled "Create launch template | EC2" and shows the "Create new v..." step. It includes sections for "Launch template name and description" (with "Mobile" entered), "Template version description" (with "v0"), and "Launch template contents" (showing the "Amazon Machine Image (AMI)" section with "Amazon Linux 2023 AMI" selected). A "Create launch template" button is visible.

Instance type

Key pair (log in)

Network settings

Storage (volumes)

Resource tags

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6... [read more](#)
ami-0fd05997b4dff7aac

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

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

[Create launch template](#)

Allow tags in metadata

User data - optional

```
#!/bin/bash
sudo -i
yum install httpd -y
systemctl start httpd
systemctl enable httpd
mkdir /var/www/html/Mobile/
cd /var/www/html/Mobile/
echo "This is Mobile ${hostname}" >> index.html
```

User data has already been base64 encoded

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6... [read more](#)
ami-0fd05997b4dff7aac

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

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

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

CloudShell Feedback

Search

EC2 VPC

21°C Clear

Summary

Software Image (AMI)
Amazon Linux 2023 AMI 2023.6... [read more](#)
ami-0fd05997b4dff7aac

Virtual server type (instance type)
t2.micro

Firewall (security group)
default

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

[Create launch template](#)

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
ENG IN 23:07 08-01-2025

Step4: create an Application Load Balancer-

The screenshot shows the 'Create Application Load Balancer' wizard on the AWS console. The current step is 'Basic configuration'. The 'Load balancer name' field contains 'lb1'. The 'Scheme' section is set to 'Internal', which is described as handling traffic from private IP addresses. The 'Load balancer IP address type' dropdown is set to 'IPv4'. Under 'Network mapping', the 'Subnet' dropdown lists 'subnet-01d99fd7d13679aa' and 'subnet-0f981fa792b16f5'. The 'Availability Zone' dropdown lists 'ap-south-1c (ap-s1-a2)'. The 'IPv4 address' dropdown shows 'Assigned by AWS'. The 'Mapping' dropdown lists 'ap-south-1b (ap-s1-a2)' and 'ap-south-1d (ap-s1-a2)'. The browser status bar at the bottom indicates it's 21°C Clear.

The screenshot shows the 'Create Application Load Balancer' wizard on the AWS console. The current step is 'Network mapping'. It shows the selected subnets: 'subnet-01d99fd7d13679aa' and 'subnet-0f981fa792b16f5'. The 'Security groups' section shows the 'default' security group selected. The 'Listeners and routing' section shows a single listener for 'HTTP' on port 80, with the target group 'Selected' and '1-6000'. The 'Load balancer tags - optional' section is present. The 'Optimize with service integrations - optional' section is also present. The browser status bar at the bottom indicates it's 21°C Clear.

Step5: create a 1st target group (Home) to attach it to a Load Balancer-

The screenshot shows the AWS CloudShell interface with the following details:

- Step 1: Specify group details**
- Basic configuration**: Settings in this section can't be changed after the target group is created.
- Choose a target type**:
 - Instances** (selected):
 - Supports load balancing to Instances within a specific VPC.
 - Facilitates the use of [Amazon EC2 Auto Scaling](#) to manage and scale your EC2 capacity.
 - IP addresses**:
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservice based architectures, simplifying inter-application communication.
 - Supports IPv6 targets, enabling end-to-end IPv6 communication, and (IPv4-to-IPv6 NAT).
 - Lambda function**:
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
 - Application Load Balancer**:
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.
- Target group name**: TgHome
- Protocol : Port**:
 - Protocol: HTTP
 - Port: 80

The screenshot shows the AWS CloudShell interface with the following details:

- IP address type**: Only instances with the indicated IP address type can be registered to this target group.
 - IPv4** (selected): An instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.
 - IPv6**: An instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)
- VPC**: Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.
 - vpc-0c0d9afe7a51c7c
IPv4 CIDR: 172.31.0.0/16
- Protocol version**:
 - HTTP1** (selected): Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.
 - HTTP2**: Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.
 - gRPC**: Send requests to targets using gRPC. Supported when the request protocol is gRPC.
- Health checks**: The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.
- Health check protocol**: HTTP
- Health check path**: Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred. /
- Attributes**

This screenshot shows the AWS CloudShell interface with the following details:

- URL:** ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup:protocol=HTTP:vpc=vpc-0e2d09afe7a51c7c
- Region:** Asia Pacific (Mumbai)
- EC2 Target Groups:** Step 2 Create target group
- Available instances (0):** No instances.
- Ports for the selected instances:** 80
- Review targets:** Targets (0) - No instances added yet.
- Buttons:** Remove all pending, Cancel, Previous, Create target group.

Step6: continue creating Load Balancer by attaching 1st target group-

This screenshot shows the AWS CloudShell interface with the following details:

- URL:** ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateALBWizard:
- Region:** Asia Pacific (Mumbai)
- EC2 Load balancers:** Create Application Load Balancer
- Security groups:** Select up to 5 security groups - default (sg-09ee240534820c020 VPC: vpc-0e2d09afe7a51c7c)
- Listeners and routing info:**
 - Listener HTTP:80:** Protocol: HTTP, Port: 80, Forward to: TgHome, Target type: Instance, IPv4, Default action: HTTP.
 - Create target group:** Create target group
 - Listener tags - optional:** Add listener tag
 - Load balancer tags - optional:** Consider adding tags to your load balancer. Tags enable you to categorize your AWS resources so you can more easily manage them. The 'Key' is required, but 'Value' is optional. For example, you can have Key = production-webserver, or Key = webserver, and Value = production.
- Optimize with service integrations - optional:** Amazon CloudFront + AWS Web Application Firewall (WAF) - new
- CloudShell Feedback:** GBP/INR -0.95%

The screenshot shows the AWS CloudWatch Metrics interface. A metric named "TgHealth" is displayed, showing a constant value of 100% across all dimensions. The chart has a single data series labeled "TgHealth" and a legend indicating "1 rule".

Load balancer details | EC2 | ap-south-1

Target group details | EC2 | ap-south-1

lb1

Details

Status: Provisioning

VPC: vpc-0e2b0f9afe7a51c7c

Hosted zone: ZP97RAFLXTNZK

Availability Zones: subnet-0f8141a7972616f5 ap-south-1c (ap-s1-a2z), subnet-042374a853a6999a76 ap-south-1a (ap-s1-a1z), subnet-01699cf0d733619a5 ap-south-1b (ap-s1-a3z)

Date created: January 8, 2025, 23:23 (UTC+05:30)

Listeners and rules (1)

Protocol/Port: HTTP:80

Forward to target group: TgHome (100%), Target group stickiness: Off

DNS name info: lb1-1522302404.ap-south-1.elb.amazonaws.com (A Record)

Listeners and rules (1) info

A listener checks for connection requests on its configured protocol and port. Traffic received by the listener is routed according to the default action and any additional rules.

Protocol/Port: HTTP:80

Default action: Forward to target group (TgHome, 100%)

Rules: 1 rule

ARN: Not applicable

Security policy: Not applicable

Default SSL/TLS certificate: mTLS

Tags: Not applicable

Step7: create 2nd target group (/Laptop)-

The screenshot shows the AWS CloudWatch Metrics interface. A metric named "TgHealth" is displayed, showing a constant value of 100% across all dimensions. The chart has a single data series labeled "TgHealth" and a legend indicating "1 rule".

Step 1 Create target group | EC2 | ap-south-1

Target group details | EC2 | ap-south-1

Create target group

Specify group details

Basic configuration

Choose a target type

- Instances
 - Supports load balancing to instances within a specific VPC.
 - Facilitates the use of Amazon EC2 Auto Scaling to manage and scale your EC2 capacity.
- IP addresses
 - Supports load balancing to VPC and on-premises resources.
 - Facilitates routing to multiple IP addresses and network interfaces on the same instance.
 - Offers flexibility with microservices-based architectures, simplifying intra-application communication.
 - Supports IP targets, enabling end-to-end IP communication, and IPv4-to-IPv6 NAT.
- Lambda function
 - Facilitates routing to a single Lambda function.
 - Accessible to Application Load Balancers only.
- Application Load Balancer
 - Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
 - Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name: TgLaptop

Protocol : Port: HTTP : 80

Protocol : Port: Choose a protocol for your target group that corresponds to the Load Balancer type that will route traffic to it. Some protocols now include anomaly detection for the targets and you can set mitigation options once your target group is created. This choice cannot be changed after creation.

Step 1 Create target group | EC2 Target group details | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup

Gmail YouTube News Translate youtube - Google S...

EC2 VPC

EC2 Target groups Create target group

IP address type

Only targets with the indicated IP address type can be registered to this target group.

IPv4
Each instance has a default network interface (eth0) that is assigned the primary private IPv4 address. The instance's primary private IPv4 address is the one that will be applied to the target.

IPv6
Each instance you register must have an assigned primary IPv6 address. This is configured on the instance's default network interface (eth0). [Learn more](#)

VPC

Select the VPC with the instances that you want to include in the target group. Only VPCs that support the IP address type selected above are available in this list.

vpc-0e2d9f5af07a31c7c
(IPv4 CIDR: 172.31.0.0/16)

Protocol version

HTTP1
Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

HTTP2
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

gRPC
Send requests to targets using gRPC. Supported when the request protocol is gRPC.

Health checks

The associated load balancer periodically sends requests, per the settings below, to the registered targets to test their status.

Health check protocol

HTTP

Health check path

Use the default path of "/" to perform health checks on the root, or specify a custom path if preferred.

/Laptop

Up to 1024 characters allowed.

[Advanced health check settings](#)

Attributes

CloudShell Feedback

SH114 / universit... Closed road

Search

Dell

ENG IN 23:25 08-01-2025

Step 2 Create target group | EC2 Target group details | EC2 | ap-south-1

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup

Gmail YouTube News Translate youtube - Google S...

CloudShell Feedback

aws EC2 VPC

EC2 Target groups Create target group

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Register targets

Available instances (0)

No instances

0 selected

Ports for the selected instances

Ports for routing traffic to the selected instances.

80
1-65535 (separate multiple ports with commas)

[Include as pending below](#)

Review targets

Targets (0)

No instances added yet

Specify instances above, or leave the group empty if you prefer to add targets later.

0 pending

Cancel Previous Create target group

CloudShell Feedback

Upcoming Earnings

Search

Dell

ENG IN 23:27 08-01-2025

Step8: create 3rd target group (/Mobile)-

The screenshot shows the AWS CloudShell interface with a browser window titled "Step 1 Create target group | EC2 | ap-south-1 | Target group details | EC2 | ap-south-1#CreateTargetGroup". The browser address bar shows the URL: "ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#CreateTargetGroup". The page is titled "Specify group details" and describes the target group as routing requests to targets and performing health checks. It shows the "Basic configuration" section with "Instances" selected as the target type. A "Target group name" input field contains "TgMobile". The "Protocol & Port" section shows "HTTP" selected. The bottom status bar indicates "CloudShell Feedback", the date "08-01-2025", and the time "23:28".

The screenshot continues the target group creation process. The browser title is now "Step 1 Create target group | EC2 | ap-south-1 | Target group details | EC2 | ap-south-1#CreateTargetGroup". The "Protocol version" section shows "HTTP1" selected. The "Health checks" section includes a "Health check protocol" dropdown set to "HTTP" and a "Health check path" input field containing "/Mobile". The "Attributes" section notes that certain default attributes will be applied. The bottom status bar remains the same as the previous screenshot.

Screenshot of the AWS Lambda console showing the creation of a new function named "HelloWorld".

The "Code" tab is selected, showing the Lambda@Edge code:

```
function handler(event, context) {  
    const response = {  
        statusCode: 200,  
        headers: {  
            'Content-Type': 'text/plain'  
        },  
        body: 'Hello from Lambda!'  
    };  
  
    context.succeed(response);  
}
```

The "Test" tab shows a successful execution with the output:

```
Execution Succeeded  
Execution ID: 12345678901234567890123456789012  
Function: HelloWorld  
Region: us-east-1  
Event Type: CloudFront  
Request ID: 12345678901234567890123456789012  
Timestamp: 2023-08-01T10:00:00Z  
Status: SUCCEEDED  
Duration: 10ms  
Memory: 128MB  
Logs:  
[{"log_line": "2023-08-01T10:00:00.000Z INFO Hello from Lambda!"}]
```

Screenshot of the AWS Lambda console showing the creation of a new function named "HelloWorld".

The "Code" tab is selected, showing the Lambda@Edge code:

```
function handler(event, context) {  
    const response = {  
        statusCode: 200,  
        headers: {  
            'Content-Type': 'text/plain'  
        },  
        body: 'Hello from Lambda!'  
    };  
  
    context.succeed(response);  
}
```

The "Test" tab shows a successful execution with the output:

```
Execution Succeeded  
Execution ID: 12345678901234567890123456789012  
Function: HelloWorld  
Region: us-east-1  
Event Type: CloudFront  
Request ID: 12345678901234567890123456789012  
Timestamp: 2023-08-01T10:00:00Z  
Status: SUCCEEDED  
Duration: 10ms  
Memory: 128MB  
Logs:  
[{"log_line": "2023-08-01T10:00:00.000Z INFO Hello from Lambda!"}]
```

Step9: attach created target group in Load Balancers rules-

The screenshot shows the AWS Lambda console interface. On the left, there's a sidebar with navigation links like 'Images', 'Elastic Block Store', 'Network & Security', 'Load Balancing', 'Auto Scaling', and 'Settings'. The main area is titled 'Load balancers (1/1)' and shows a table with one item: 'lb1'. Below it, the 'Load balancer: lb1' details are shown, specifically the 'Listeners and rules' section. It lists a single rule for 'HTTP:80' that forwards traffic to the 'TgHome' target group.

This screenshot shows the AWS Lambda console again, but this time the 'Listener details' page for the 'HTTP:80' listener of the 'lb1' load balancer. It displays the 'Details' section with the protocol and port information, and the 'Default actions' section which shows the 'Forward to target group' rule. Below that, the 'Listener rules' section is shown, which contains a single rule for the 'Default' condition that forwards traffic to the 'TgHome' target group.

Screenshot of the AWS CloudFront console showing the "Edit listener" page for an Application Load Balancer (ALB). The listener ARN is arn:aws:elasticloadbalancing:ap-south-1:438465154284:listener/app/lb1/8aa36a07bb1ef2cd/05dbb1ea848a95d. The protocol is set to HTTP and port 80. The default action is "Forward to target group". There are three target groups: TgHome, TgLaptop, and TgMobile, each with a weight of 1 and a percent of 33.33%. The "Target group stickiness" checkbox is unchecked.

Screenshot of the AWS CloudFront console showing the "Listener details" page for an Application Load Balancer (ALB). The listener ARN is arn:aws:elasticloadbalancing:ap-south-1:438465154284:loadbalancer/app/lb1/8aa36a07bb1ef2cd/05dbb1ea848a95d. The protocol is set to HTTP:80. The "Forward to target group" default action is selected, pointing to the load balancer lb1. The "Listener rules" section shows one rule named "Default" with the condition "If no other rule applies". The rule points to the target group TgHome with a weight of 1 and a percent of 33.33%. The "Target group stickiness" checkbox is unchecked.

Step10: create 1st Autoscaling group (Home) by its specific template-

Choose launch template

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name

Enter a name to identify the group.

aglHome

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.

Home

Description

VO

AMI ID

ami-0fd05997b4df7aac

Key pair name

sadhyaraj

Launch template

Home

Version

Default (1)

Security groups

-

Security group IDs

sg-09eab24b53482c0b20

Instance type

t2.micro

Request Spot Instances

No

Additional details

Choose instance launch options

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

Instance type requirements

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

Home

Version

Default

Description

VO

VPC

Choose the VPC that defines the virtual network for your Auto Scaling group.

vpc-02e0f9fb9a23a7c7c

172.31.0.1/16 Default

Create a VPC

Availability Zones and subnets

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

Select Availability Zones or subnets

ap-south-1a [Subnet: 042174b5beff89a79] 172.31.0.1/20 Default

ap-south-1b [Subnet: 0149b5d5d735017ea] 172.31.0.2/20 Default

ap-south-1c [Subnet: 0ff18141a792616f9] 172.31.1/20 Default

Create a subnet

Availability Zone distribution - new

Auto Scaling automatically balances instances across Availability Zones. If launch failures occur in a zone, select a strategy.

Balanced best effort: If launches fail in one Availability Zone, Auto Scaling will attempt to launch in another healthy Availability Zone.

Balanced only: If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the remaining Availability Zone to prevent balanced distribution.

Your requested instance type (t2.micro) is not available in 1 Availability Zone. You may need to change the instance type or choose other Availability Zones for better redundancy.



Attach to an existing load balancer and select 1st target group:

Integrate with other services - optional Info
Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift. You can also customize health check replacements and monitoring.

Load balancing Info
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

Existing load balancer target groups
Only load balancers belonging to the same VPC as your Auto Scaling group are available for selection.

Select target groups

tjHome | HTTP Application Load Balancer: lb-1

VPC Lattice integration options Info
To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

No VPC Lattice service
VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service
Incoming requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group.

Create new VPC Lattice service

Select desired capacity of scaling instances:

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
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
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 Max desired capacity
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.

Choose a replacement behavior depending on your availability requirements

Step11: create 2nd AutoScaling group (Laptop) by its specific template-

The screenshot shows the 'Choose launch template' step of the AWS Auto Scaling wizard. The 'Launch template' dropdown is set to 'Launch'. Other settings shown include:

- Auto Scaling group name:** asgLaptop
- AMI ID:** ami-0f053917d5df7faec
- Security groups:** -
- Instance type:** t2.micro
- Request Spot Instances:** No
- Storage (volume):** -
- Date created:** Wed Jan 06 2022 22:09:25 (IST+0530/India Standard Time)



The screenshot shows the 'Choose instance launch options' step of the AWS Auto Scaling wizard. The 'Network' section is visible, showing the chosen VPC as 'vpc-02b05a95a7a751c8' and the 'Balanced best effort' distribution strategy. Other settings shown include:

- Launch template:** Launch (ami-0f053917d5df7faec)
- Version:** Default
- Description:** V1
- Instance type:** t2.micro
- Availability Zone distribution:** Balanced best effort (selected)
- Note:** Your requested instance type (t2.micro) is not available in 1 Availability Zone. You may need to change the instance type or choose other Availability Zones for better resilience. [Learn more](#)



Attach to an existing load balancer and select 2nd target group:

The screenshot shows the 'Create Auto Scaling group' wizard at Step 3: 'Integrate with other services - optional'. The 'Attach to an existing load balancer' option is selected. A dropdown menu shows 'Tg-ap-sa1 (HTTP Application Load Balancer)' as the chosen target group.

Select desired capacity of scaling instances:

The screenshot shows the 'Create Auto Scaling group' wizard at Step 4: 'Configure group size and scaling - optional'. The 'Max desired capacity' field is set to 2. The 'Automatic scaling - optional' section shows 'No scaling policies' selected.

Step12: create 3rd Autoscaling group (Mobile) by its specific template-

Choose launch template

Name: `asgmobile`

Launch template: `Mobile`

Description: `V0`

AMI ID: `ami-0505907a0ff7ac`

Key pair name: `andykey`

Security group IDs: `sg-45e245346ebc20`

Instance type: `t2.micro`

Request Spot Instances: `No`

Date created: `Wed Jan 08 2023 23:07:45 GMT+0530 (India Standard Time)`

Next Step



Choose instance launch options

Instance type requirements: `t2.micro`

VPC: `vpc-0d20f0f97c123456`

Network

Availability Zone distribution - new: `Balanced best effort`

Next Step



Attach to an existing load balancer and select 3rd target group:

Integrate with other services - optional Info

Use a load balancer to distribute network traffic across multiple servers. Enable service-to-service communications with VPC Lattice. Shift resources away from impaired Availability Zones with zonal shift.

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

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

Attach to a new load balancer Create a basic load balancer to attach to your Auto Scaling group

Load balancing Info

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

Existing load balancer target groups Only instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups This option allows you to attach Application, Network, or Gateway Load Balancers.

Select target groups Select target groups

TargetGroups [HTTP] applicationloadbalancer-1

VPC Lattice integration options Info

To improve networking capabilities and scalability, integrate your Auto Scaling group with VPC Lattice. VPC Lattice facilitates communications between AWS services and helps you connect and manage your applications across compute services in AWS.

Select VPC Lattice service to attach No VPC Lattice service

VPC Lattice will not manage your Auto Scaling group's network access and connectivity with other services.

Attach to VPC Lattice service Assigning requests associated with specified VPC Lattice target groups will be routed to your Auto Scaling group

Create new VPC Lattice service

Application Recovery Controller (ARC) zonal shift - now Info

During an Availability Zone impairment, target instance launches towards other healthy Availability Zones.

Enable zonal shift New instance launches will be redirected towards healthy Availability Zones until the zonal shift is canceled.

Health checks

Health checks increase availability by replacing unhealthy instances. When you use multiple health checks, all are evaluated, and if at least one fails, instance replacement occurs.

EC2 health checks Always enabled

Additional health check types - optional Info

Select desired capacity of scaling instances:

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

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

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

Max desired capacity

1

2

Equal or less than desired capacity

Equal or greater than desired capacity

Automatic scaling - optional Info

Choose whether to use a target tracking policy Info

Your Auto Scaling group will remain at its initial size and will not dynamically resize to meet demand.

No scaling policies

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.

Choose a replacement behavior depending on your availability requirements

Created all Autoscaling groups:

The screenshot shows the AWS Auto Scaling Groups page. At the top, there are tabs for 'Launch configurations' (highlighted in blue), 'Launch templates', 'Actions', and 'Create Auto Scaling group'. Below the tabs, a search bar and a table list three Auto Scaling groups:

Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
asg3mobile	Mobile Version Default	0	Updating capacity...	1	1	2	ap-south-1c, ap-south-1b, ap-south-1a
asg2laptop	Laptop Version Default	1	-	1	1	2	ap-south-1c, ap-south-1b, ap-south-1a
asg1home	Home Version Default	1	-	1	1	2	ap-south-1c, ap-south-1b, ap-south-1a

At the bottom left, it says '0 Auto Scaling groups selected'. The browser status bar at the bottom indicates the date as 08-01-2025.

Step13: instances created by Autoscaling groups-

The screenshot shows the AWS Instances page. The left sidebar includes 'Dashboard', 'EC2 Global View', 'Events', 'Instances' (selected), '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', and 'Network & Security'. The main content area displays three instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public IP
i-09befa818b725936	i-09befa818b725936	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	ec2-13-201-36-251.ap...	13.201.36...
i-0eabb68d4284bb106	i-0eabb68d4284bb106	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-0-131-89.ap-so...	65.0.131...
i-0383d2f3da48d9e93	i-0383d2f3da48d9e93	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	ec2-65-2-123-75.ap-so...	65.2.123...

The browser status bar at the bottom indicates the date as 08-01-2025.

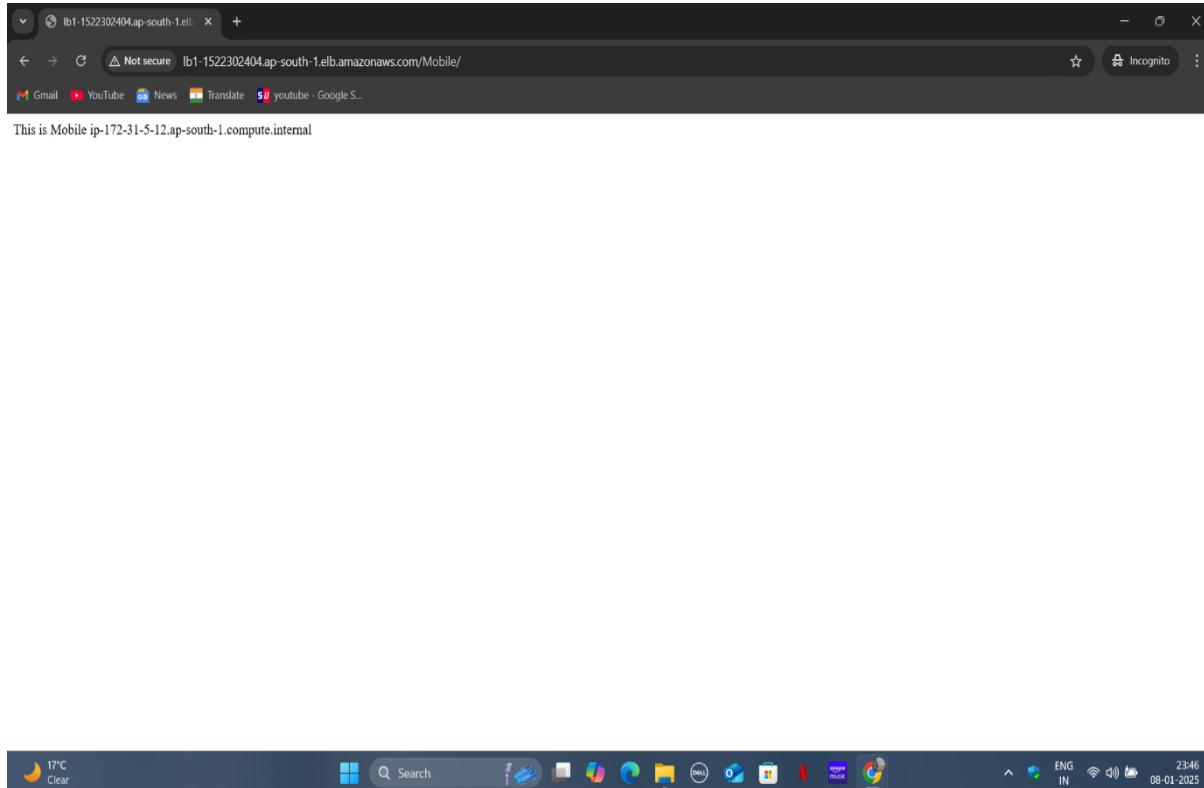
Step14: accessing 1st instance created by ASG1 through Load Balancers DNS-



Step15: accessing 2nd instance created by ASG2 through LoadBalancers DNS with /Laptop-



Step16: accessing 3rd instance created by ASG3 through LoadBalancers DNS with /Mobile-



Step17: creating dynamic scaling property for Home asg to check autoscaling-

asg1home Capacity overview

Desired capacity: 1 Scaling limits (Min - Max): 1 - 2 Desired capacity type: Units (number of instances) Status: -

Date created: Thu Jan 16 2025 18:01:34 GMT+0530 (India Standard Time)

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.

Dynamic scaling policies (0) [Info](#) [Actions](#) [Create dynamic scaling policy](#)

No dynamic scaling policies have been created

Dynamic scaling policies use real-time data to scale your group based on configurable metrics.

[Create dynamic scaling policy](#)



Step18: creating dynamic scaling property for Home asg to check autoscaling at 50%

The screenshot shows the 'Create dynamic scaling policy' wizard. The 'Policy type' is set to 'Target tracking scaling'. The 'Scaling policy name' is 'Target Tracking Policy'. The 'Metric type' is 'Average CPU utilization'. The 'Target value' is '50'. The 'Instance warmup' is '10 seconds'. A checkbox for 'Disable scale in to create only a scale-out policy' is unchecked. At the bottom right are 'Cancel' and 'Create' buttons.



The screenshot shows the 'Auto Scaling group details' page for 'asg1home'. A green success message says 'Dynamic scaling policy created or edited successfully.' Below it, the 'Capacity overview' table shows 'Desired capacity' as 1, 'Scaling limits (Min - Max)' as 1 - 2, and 'Desired capacity type' as 'Units (number of instances)'. The 'Status' is '-'.

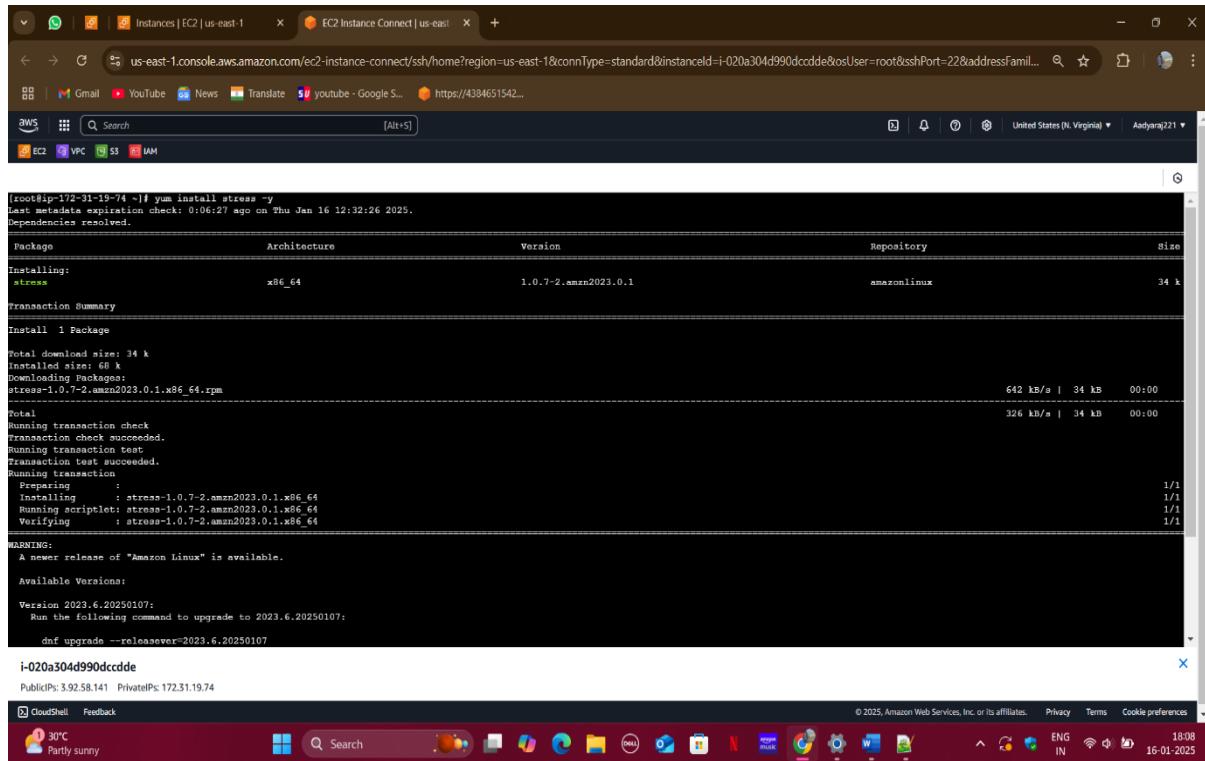
The navigation bar includes 'Details', 'Integrations - new', 'Automatic scaling' (which is selected), 'Instance management', 'Instance refresh', 'Activity', and 'Monitoring'.

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.'

The 'Dynamic scaling policies' section shows one policy named 'Target Tracking Policy'. It has a 'Policy type' of 'Target tracking scaling'. There are 'Actions' and 'Create dynamic scaling policy' buttons, and a pagination indicator showing page 1 of 1.

The taskbar at the bottom shows the AWS CloudShell icon again.

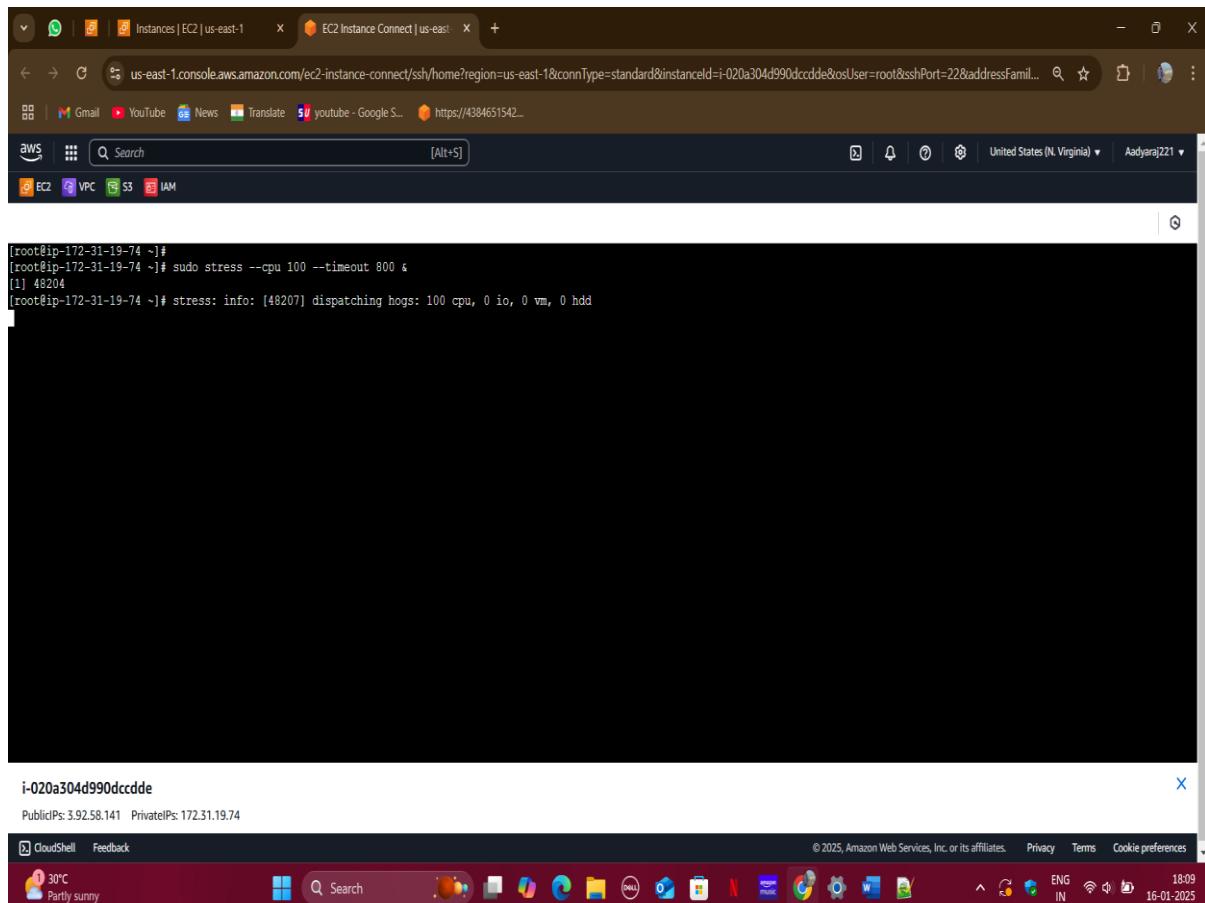
Step19: taking access of Home instance and installing stress-



```
[root@ip-172-31-19-74 ~]# yum install stress -y
Last metadata expiration check: 0:06:27 ago on Thu Jan 16 12:32:26 2025.
Dependencies resolved.
Package           Architecture      Version       Repository   Size
Installing:
stress            x86_64          1.0.7-2.amzn2023.0.1
amazonlinux        34 kB
Transaction Summary
Install 1 Package
Total download size: 34 k
Installed size: 60 k
Downloading Packages:
stress-1.0.7-2.amzn2023.0.1.x86_64.rpm
Total
Running transaction check
transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing :
Installing : stress-1.0.7-2.amzn2023.0.1.x86_64
Running scriptlet: stress-1.0.7-2.amzn2023.0.1.x86_64
Verifying  : stress-1.0.7-2.amzn2023.0.1.x86_64
WARNING:
A newer release of "Amazon Linux" is available.

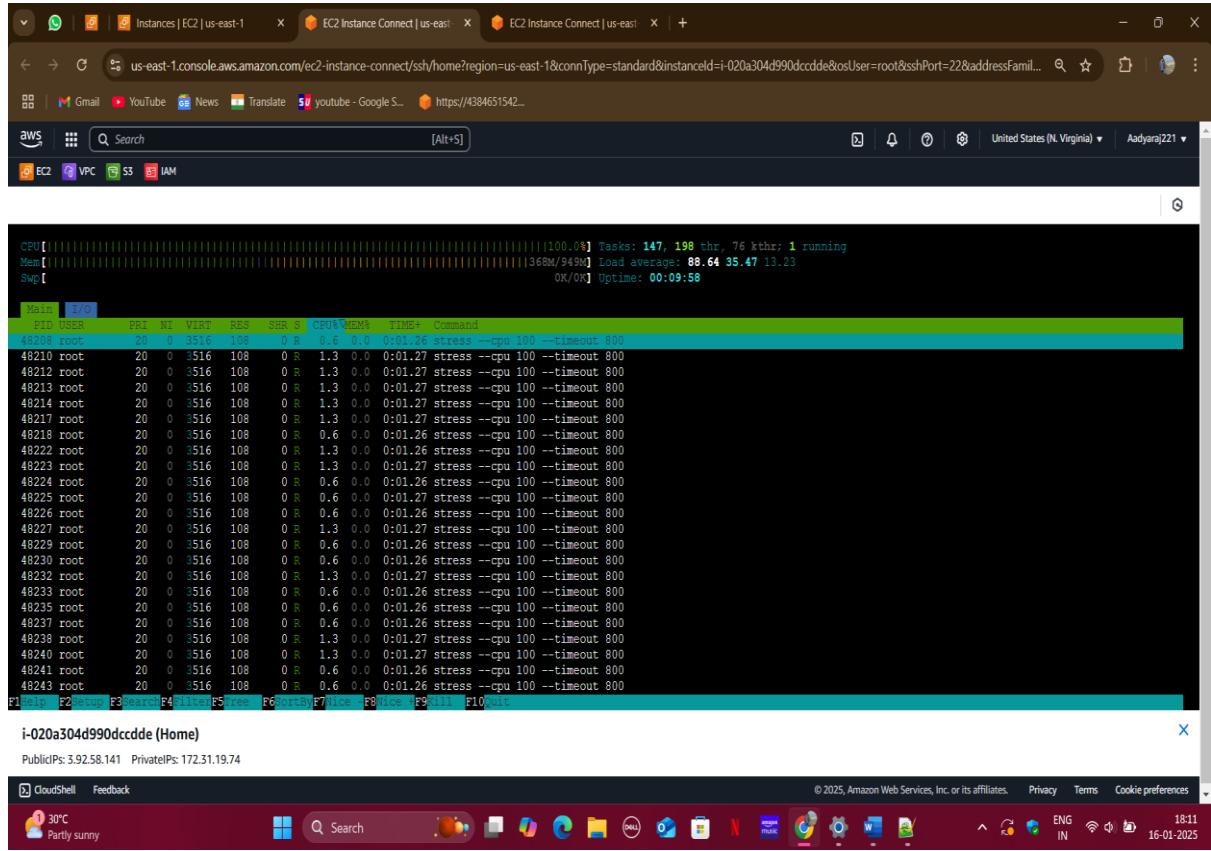
Available Versions:
Version 2023.6.20250107:
Run the following command to upgrade to 2023.6.20250107:
dnf upgrade --releasever=2023.6.20250107
i-020a304d990dccde
Public IPs: 3.92.58.141 Private IPs: 172.31.19.74
```

Step20: Entering stress command for Home instance to check Autoscaling-



```
[root@ip-172-31-19-74 ~]#
[root@ip-172-31-19-74 ~]# sudo stress --cpu 100 --timeout 800 &
[1] 48204
[root@ip-172-31-19-74 ~]# stress: info: [48207] dispatching hogs: 100 cpu, 0 io, 0 vm, 0 hdd
i-020a304d990dccde
Public IPs: 3.92.58.141 Private IPs: 172.31.19.74
```

Step21: Entering htop command for Home instance to check Autoscaling-



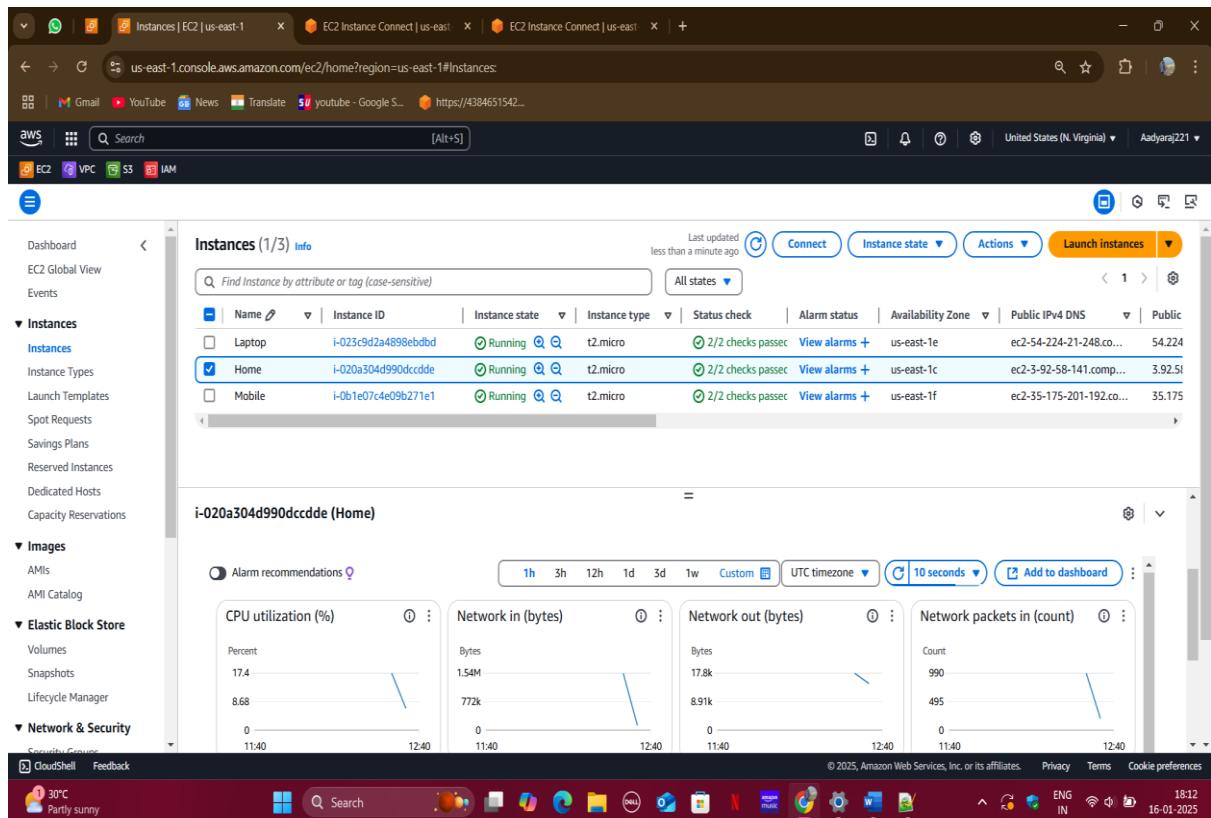
```
[100.0%] Tasks: 147, 198 thr, 76 kthr; 1 running
Mem: [ 368M/949M] Load average: 88.64 35.47 13.23
Swap: [OK/OK] Uptime: 00:09:58

CPU [ 0.0% idle ] I/O [ 0.0% idle ]
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
48208 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48210 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48212 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48213 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48214 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48217 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48218 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48222 root 20 0 3516 108 0 R 1.3 0.0 0:01.26 stress --cpu 100 --timeout 800
48223 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48224 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48225 root 20 0 3516 108 0 R 0.6 0.0 0:01.27 stress --cpu 100 --timeout 800
48226 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48227 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48229 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48230 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48232 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48233 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48235 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48237 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48238 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48240 root 20 0 3516 108 0 R 1.3 0.0 0:01.27 stress --cpu 100 --timeout 800
48241 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
48243 root 20 0 3516 108 0 R 0.6 0.0 0:01.26 stress --cpu 100 --timeout 800
F1 3.92.58.141 F2 172.31.19.4 F3 172.31.19.4 F4 172.31.19.4 F5 172.31.19.4 F6 172.31.19.4 F7 172.31.19.4 F8 172.31.19.4 F9 172.31.19.4 F10 172.31.19.4
```

i-020a304d990dccdde (Home)
PublicIPs: 3.92.58.141 PrivateIPs: 172.31.19.4

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 30°C Partly sunny 18:11 16-01-2025

Step22: monitoring Home instance for Autoscaling-



Step23: Autoscaling of Home instance done successfully-

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

Instances (1/4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS	Public
Laptop	i-023c9d2a4898ebdbd	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1e	ec2-54-224-21-248.co...	54.224
i-0cad7ec75468e603	Running	t2.micro	Initializing	2/2 checks passed	View alarms +	us-east-1f	ec2-18-204-220-201.co...	18.204
Home	i-020a304d990dcddde	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1c	ec2-3-92-58-141.comp...	3.925
Mobile	i-0b1e07c4e09b271e1	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1f	ec2-35-175-201-192.co...	35.175

i-020a304d990dcddde (Home)

Alarm recommendations

1h 3h 12h 1d 3d 1w Custom UTC timezone 10 seconds Add to dashboard

CPU utilization (%) Network in (bytes) Network out (bytes) Network packets in (count)

Percent	Bytes	Bytes	Count
17.4	1.54M	17.8k	990
8.68	772k	8.91k	495

CloudShell Feedback 30°C Partly sunny Search ENG IN 18:14 16-01-2025

*Do same for other two asg and instances for checking Autoscaling.