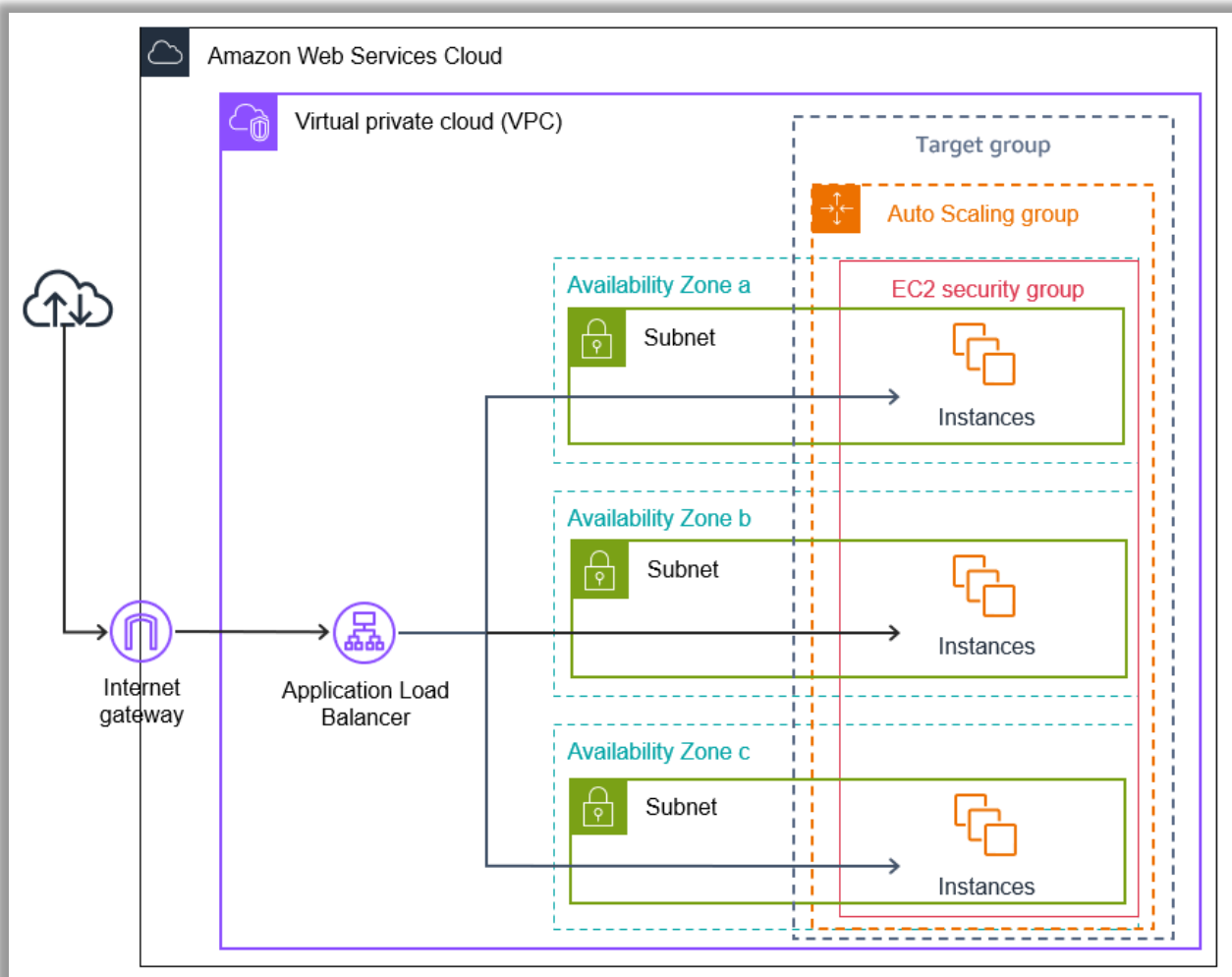


# Auto Scaling Group

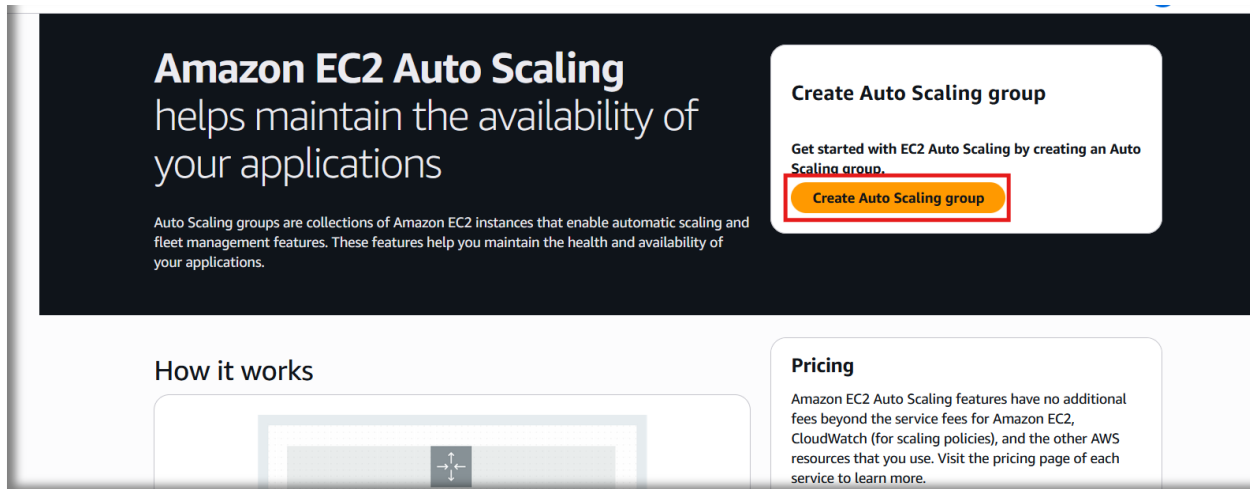
Your First Step Toward Cloud Automation



 What is an Auto Scaling Group (ASG)?

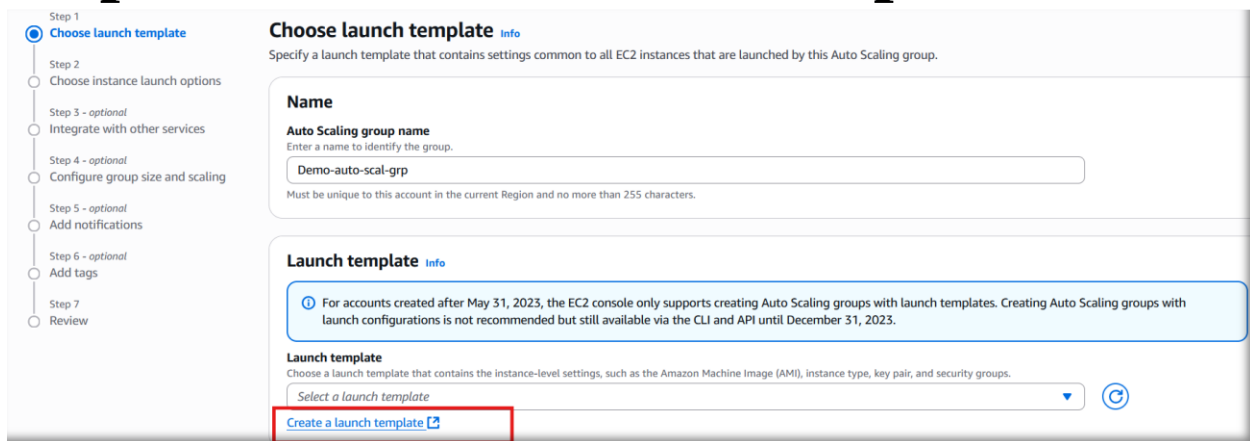
An **Auto Scaling Group** that automatically manages the number of **EC2 instances** in your application — based on real-time traffic and demand.

## Step 1. Create Auto Scaling Group



The screenshot shows the Amazon EC2 Auto Scaling console landing page. The main heading is "Amazon EC2 Auto Scaling helps maintain the availability of your applications". Below this, a subheading reads: "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." On the right side, there is a "Create Auto Scaling group" button, which is highlighted with a red rectangle. Below the button, there is a link to "Get started with EC2 Auto Scaling by creating an Auto Scaling group." and another "Create Auto Scaling group" button. At the bottom, there are sections for "How it works" and "Pricing".

## Step 2. Create a Launch Template



The screenshot shows the "Choose launch template" step in the Amazon EC2 console. On the left, there is a navigation pane with steps 1 through 7. Step 1, "Choose launch template", is selected. The main content area has a heading "Choose launch template" and a subheading "Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group." Below this, there is a "Name" section with a text input field labeled "Auto Scaling group name" and a value of "Demo-auto-scal-grp". A note below the input field states: "Must be unique to this account in the current Region and no more than 255 characters." Below the "Name" section, there is a "Launch template" section with a dropdown menu labeled "Select a launch template" and a "Create a launch template" button, which is highlighted with a red rectangle. A note above the dropdown menu states: "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."

Step 3. Give unique name to your template and choose quick start AMI

### Launch template name and description

Launch template name - *required*

my-temp-for-auto-scal-grp

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


### ▼ Application and OS Images (Amazon Machine Image) - required [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


Recents

Quick Start


Amazon Linux




macOS




Ubuntu




Windows




Red Hat




SUSE Linux



Debian



  
Browse more AMIs

Including AMIs from AWS, Marketplace and the Community

#### Amazon Machine Image (AMI)

Amazon Linux 2023 AMI

ami-006b4a3ad5f56fbd6 (64-bit (x86), uefi-preferred) / ami-0745b6edac3bff4ba (64-bit (Arm), uefi)

Virtualization: hvm    ENA enabled: true    Root device type: ebs

Free tier eligible ▼

**Step 4.** Also fill “**User Data**” field which is used to provide a **script that will run automatically** when the instance is launched for the first time.

**User data - optional** [Info](#)  
Upload a file with your user data or enter it in the field.

[Choose file](#)

```
#!/bin/bash
# Use this for your user data (script from top to bottom)
# install httpd (Linux 2 version)
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
echo "<ch1>Hello World from $(hostname -f)</h1>" > /var/www/html/index.html
```

☐ User data has already been base64 encoded

**Virtual server type (instance type)**  
t3.micro

**Firewall (security group)**  
default

**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](#)

## Template Launched Successfully.

✓ **Success**  
Successfully created [my-temp-for-auto-scal-grp\(lt-064a1ef869a963064\)](#).

## Step 5. For Auto scaling give your template name.

- It stores your instance configuration (AMI, instance type, key, etc)
- You can reuse it easily in Auto Scaling Groups or future launches.

**Launch template** [Info](#)

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.

**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.

[my-temp-for-auto-scal-grp](#) [Create a launch template](#)

**Version**  
Default (1) [Create a launch template version](#)

## Step 6. Select all available zones.

- It decides where your EC2 instance runs.
- Choosing the right zone improves performance, redundancy, and high availability

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

vpc-0d3e6b508f916d3d9  
172.31.0.0/16 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 and subnets

- eu-north-1c | subnet-0ae3d0fb8e13d5a67
- eu-north-1b | subnet-0dd8290178d80e454
- eu-north-1a | subnet-0564382b012af7878

[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 unhealthy Availability Zone to preserve balanced distribution.

Cancel Skip to review Previous Next

**Follow the given steps and do Next.**

**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 instance target groups that belong to the same VPC as your Auto Scaling group are available for selection.

Select target groups

demo-target-grp | HTTP  
Application Load Balancer: Inst-1-2-demo-load-balancer

demo-target-grp | HTTP

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.

**Here you can attach existing load balancer**

## Turn on Elastic Load Balancing Health Check

**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](#)

☒ Turn on Elastic Load Balancing health checks **Recommended**  
Elastic Load Balancing monitors whether instances are available to handle requests. When it reports an unhealthy instance, EC2 Auto Scaling can replace it on its next periodic check.

**EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the [Load Balancer console](#).** ✕

☐ Turn on VPC Lattice health checks  
VPC Lattice can monitor whether instances are available to handle requests. If it considers a target as failed a health check, EC2 Auto Scaling replaces it after its next periodic check.

☐ Turn on Amazon EBS health checks  
EBS monitors whether an instance's root volume or attached volume stalls. When it reports an unhealthy volume, EC2 Auto Scaling can replace the instance on its next periodic health check.

**Health check grace period** [Info](#)  
This time period delays the first health check until your instances finish initializing. It doesn't prevent an instance from terminating when placed into a non-running state.

300 seconds

[Cancel](#) [Skip to review](#) [Previous](#) [Next](#)

**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

**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.

2

**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**  
1  
Equal or less than desired capacity

**Max desired capacity**  
4  
Equal or greater than desired capacity

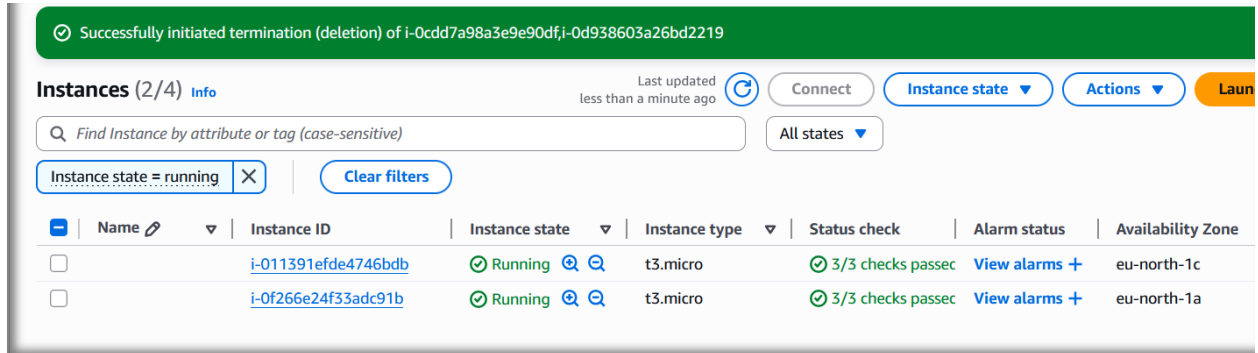
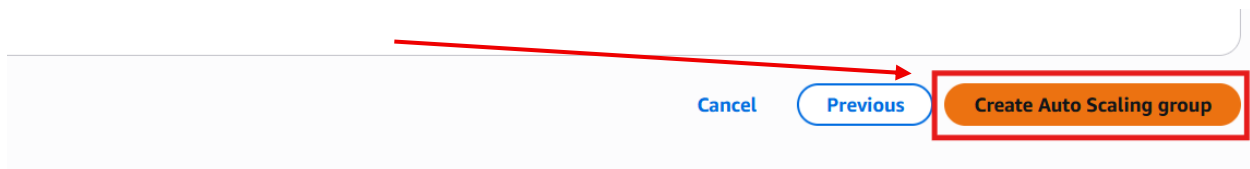
**Automatic scaling - optional**  
Choose whether to use a target tracking policy [Info](#)  
You can set up other metric-based scaling policies and scheduled scaling after creating your Auto Scaling group.

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

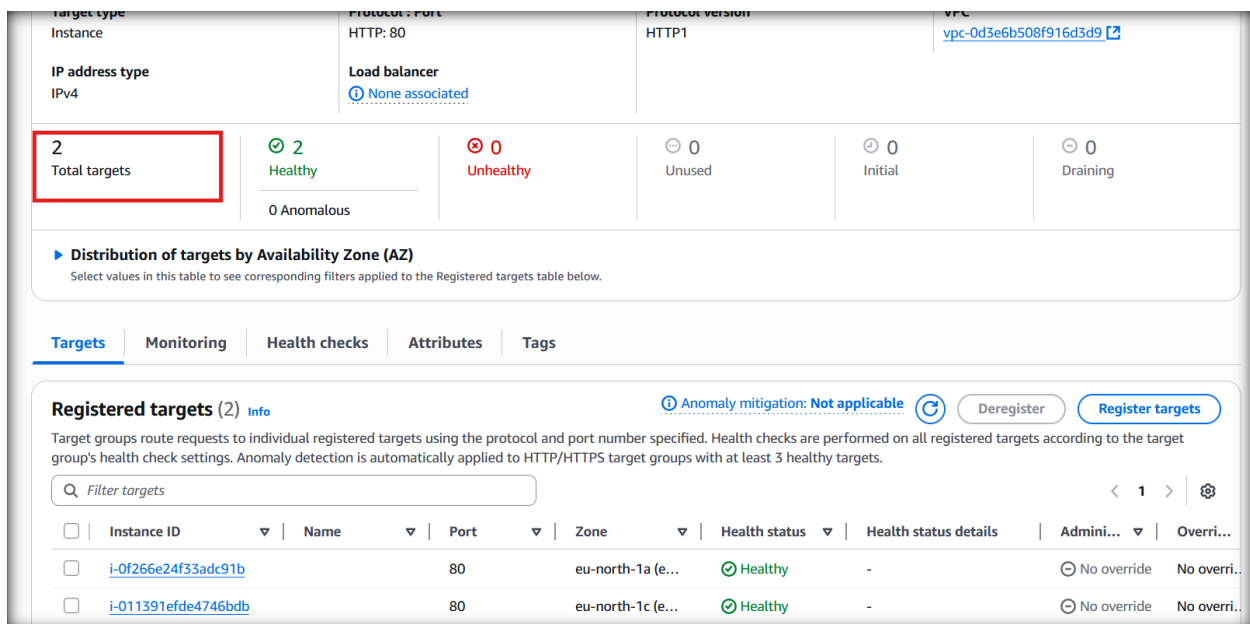
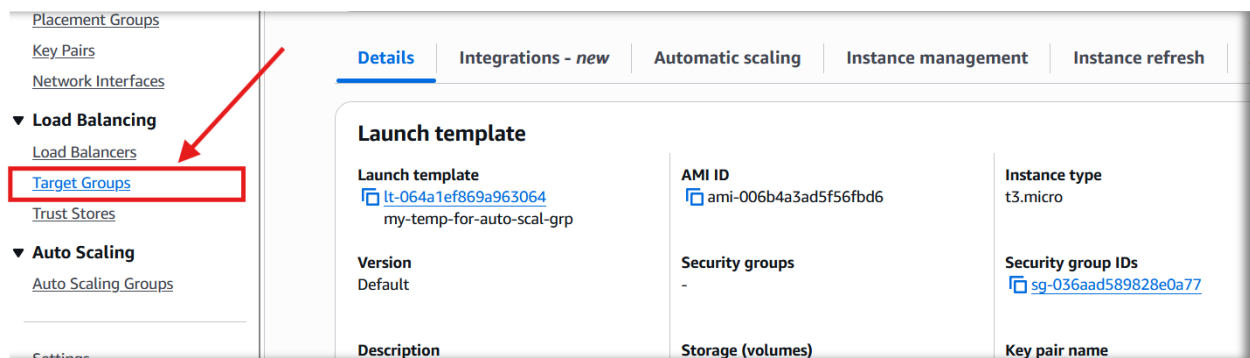
☐ Target tracking scaling policy  
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

Defines number of instances to run and

Click on Create Auto Scaling group



## Navigate to Target Groups...

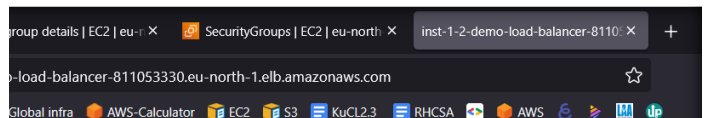


When I go to load balancer then copy the DNS name and paste on new tab

The screenshot shows the AWS Management Console interface. On the left sidebar, the 'Load Balancing' section is expanded, and 'Load Balancers' is highlighted with a red box and a red arrow. The main content area shows the 'Targets' tab for a selected load balancer, displaying 'Registered targets (2)'. Below this, there is a table of load balancers. The first row is highlighted with a red box and a red arrow, showing the 'Inst-1-2-demo-load-bal...' with a status of 'Active'.

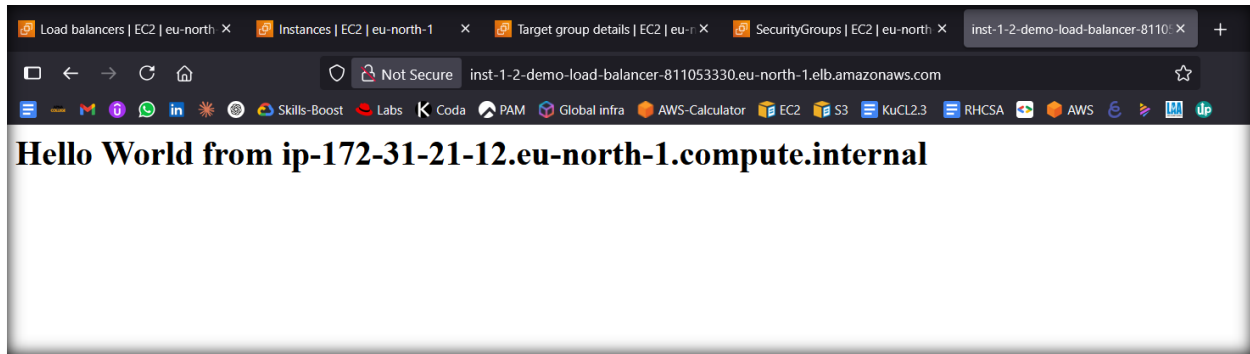
Name	DNS name	State	VPC ID	Availability Zones
Inst-1-2-demo-load-bal...	Inst-1-2-demo-load-balanc...	Active	vpc-0d3e6b508f916d3d9	3 Availability Zones

Copy the DNS name and paste in new tab

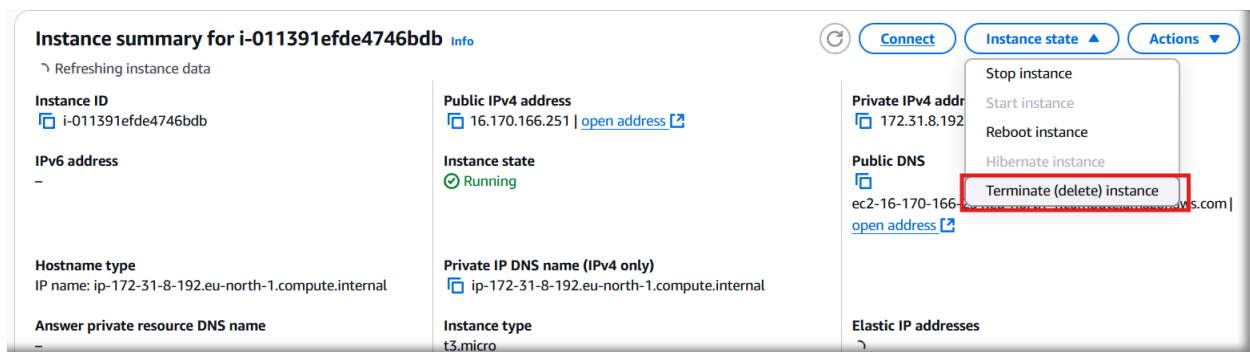


Hello world from ip-172-31-8-192.eu-north-1.compute.internal

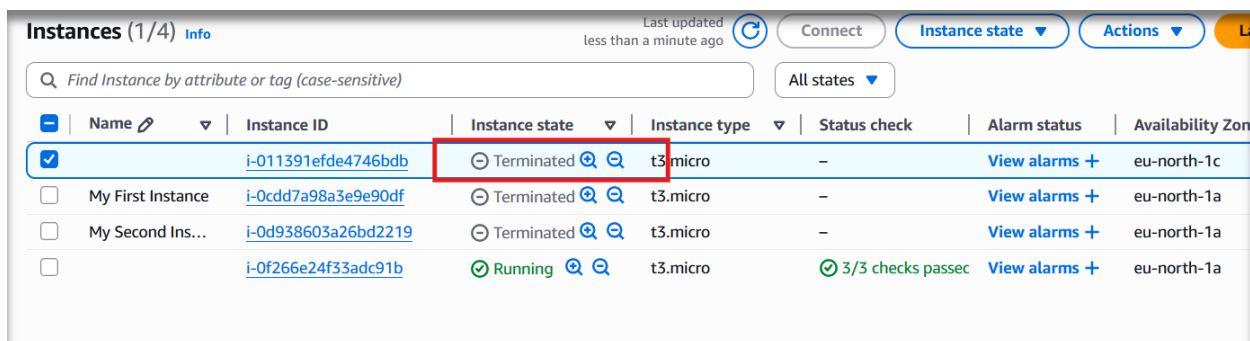




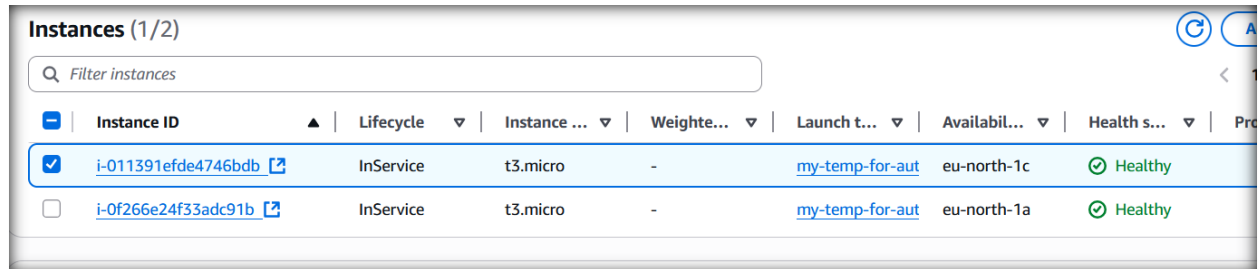
I get hello world from my both instances; this is cool because these instances are created by Auto Scaling group.



Let's terminate one instance and see what happens to it.

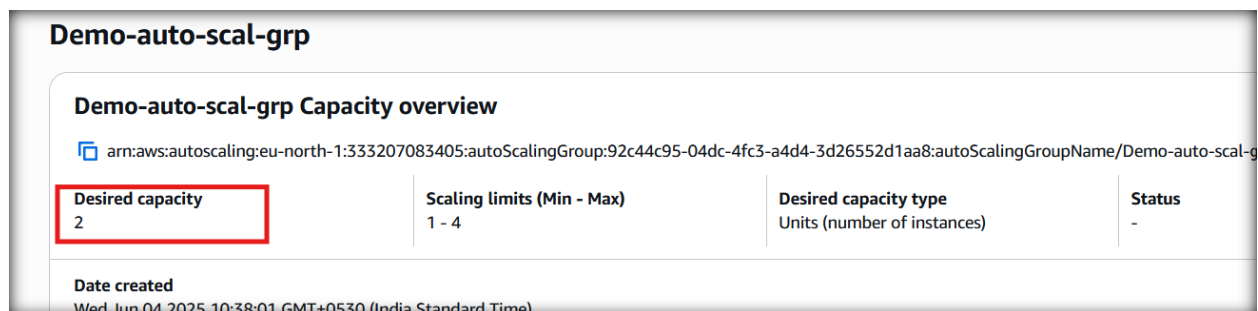


After terminating one instance, still both are healthy.



The screenshot shows the 'Instances (1/2)' page in the AWS Management Console. It displays a table with two instances, both in the 'InService' state and marked as 'Healthy'. The first instance has ID 'i-011391efde4746bdb' and the second has ID 'i-0f266e24f33adc91b'. Both are using 't3.micro' instances and are part of the 'my-temp-for-aut' launch template in the 'eu-north-1' region.

Instance ID	Lifecycle	Instance ...	Weighte...	Launch t...	Availabil...	Health s...	Pro
<input checked="" type="checkbox"/> <a href="#">i-011391efde4746bdb</a>	InService	t3.micro	-	<a href="#">my-temp-for-aut</a>	eu-north-1c	Healthy	
<input type="checkbox"/> <a href="#">i-0f266e24f33adc91b</a>	InService	t3.micro	-	<a href="#">my-temp-for-aut</a>	eu-north-1a	Healthy	



The screenshot shows the 'Demo-auto-scal-grp' page in the AWS Management Console. It displays the 'Capacity overview' for the 'Demo-auto-scal-grp'. The 'Desired capacity' is highlighted with a red box and is set to 2. The 'Scaling limits (Min - Max)' are 1 - 4. The 'Desired capacity type' is 'Units (number of instances)' and the 'Status' is '-'. The 'Date created' is 'Wed Jun 04 2025 10:38:01 GMT+0530 (India Standard Time)'.

Desired capacity	Scaling limits (Min - Max)	Desired capacity type	Status
2	1 - 4	Units (number of instances)	-

Date created  
Wed Jun 04 2025 10:38:01 GMT+0530 (India Standard Time)

If you terminate 1 instance out of 2 in an Auto Scaling Group, it will automatically launch a new one **to maintain the desired capacity** and ensure high availability.

Here is output:

Details	Integrations - new	Automatic scaling	Instance management	Instance refresh	Activity	Monitoring
Instances (3)						
<input type="text" value="Filter instances"/>						
<input type="checkbox"/>	Instance ID	Lifecycle	Instance ...	Weighte...	Launch t...	Availabil...
<input type="checkbox"/>	<a href="#">i-011391efde4746bdb</a>	Terminating	t3.micro	-	<a href="#">my-temp-for-aut</a>	eu-north-1c
<input type="checkbox"/>	<a href="#">i-0bd5eaff76bf3956f</a>	InService	t3.micro	-	<a href="#">my-temp-for-aut</a>	eu-north-1c
<input type="checkbox"/>	<a href="#">i-0f266e24f33adc91b</a>	InService	t3.micro	-	<a href="#">my-temp-for-aut</a>	eu-north-1a
						Unhealthy
						Healthy
						Healthy

That instance gone Unhealthy and new instance created automatically.

Instances (1/5)	Info	Connect	Instance state	Actions	Launch instances
<input type="text" value="Find Instance by attribute or tag (case-sensitive)"/>					
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check
<input type="checkbox"/>		<a href="#">i-0bd5eaff76bf3956f</a>	Running	t3.micro	3/3 checks passec
<input checked="" type="checkbox"/>		<a href="#">i-011391efde4746bdb</a>	Terminated	t3.micro	-
<input type="checkbox"/>	My First Instance	<a href="#">i-0cdd7a98a3e9e90df</a>	Terminated	t3.micro	-
<input type="checkbox"/>	My Second Ins...	<a href="#">i-0d938603a26bd2219</a>	Terminated	t3.micro	-
<input type="checkbox"/>		<a href="#">i-0f266e24f33adc91b</a>	Running	t3.micro	3/3 checks passec
					View alarms +
					View alarms +
					View alarms +
					View alarms +
					View alarms +

## ✓ Conclusion

- Auto Scaling Groups help build scalable and reliable cloud apps.
- They adjust EC2 instances automatically based on traffic.
- Great for learning cloud automation and DevOps basics.
- A strong step toward becoming a Cloud/DevOps Engineer.

Keep experimenting. The cloud is your playground. ☁️🚀