

AUTO SCALING

Auto Scaling in AWS is a service that automatically adjusts the number of EC2 instances or other AWS resources based on demand, ensuring high availability and cost optimization. It helps maintain application performance by dynamically adding or removing instances according to real-time traffic, workload, or other defined policies.

Auto Scaling is a cloud computing feature that allows an application to automatically adjust its resources (like compute instances) based on real-time demand. This ensures that the application maintains performance and availability while optimizing cost-efficiency.

Types of Auto Scaling:

1. Vertical Auto Scaling (Scaling Up/Down):

- This involves adding more resources (CPU, RAM, storage) to an existing instance to meet demand, or reducing resources when demand decreases.
- It's limited to scaling within the capabilities of a single instance (e.g., upgrading an instance type to a larger one).

2. Horizontal Auto Scaling (Scaling In/Out):

- This is the more common form of auto scaling and involves adding or removing instances from a pool based on demand.
- **Scaling Out:** Adding more instances when the demand increases (e.g., handling more traffic).
- **Scaling In:** Removing instances when the demand decreases (to save cost).

3. Scheduled Auto Scaling:

- Allows you to set up scaling actions based on a predefined schedule (e.g., scaling out at certain hours of the day when traffic is expected to be high).
- Useful for predictable workloads, like when traffic increases during specific hours or events.

4. **Dynamic Auto Scaling:**

- Adjusts resources based on real-time metrics such as CPU utilization, request latency, or memory usage.
- It reacts to changes in traffic or load automatically and continuously, without predefined schedules.

5. **Predictive Auto Scaling:**

- Uses machine learning algorithms to predict future traffic patterns and adjust resources ahead of time to ensure performance.
- Ideal for workloads with complex or cyclical traffic, as it anticipates the needs based on historical data.

Key Components of AWS Auto Scaling

1. **Launch Template/Configuration** – Defines the instance type, AMI, security group, and other settings.
2. **Auto Scaling Group (ASG)** – A collection of EC2 instances that are managed together.
3. **Scaling Policies** – Rules that determine when to add or remove instances:
 - **Target Tracking** – Keeps a metric (e.g., CPU utilization) at a target level.
 - **Step Scaling** – Adjusts capacity in steps (e.g., add 2 instances if CPU > 70%).
 - **Simple Scaling** – Adds/removes instances based on predefined conditions.
 - **Scheduled Scaling** – Increases/decreases instances at specific times.

Benefits of AWS Auto Scaling

- ✓ **High Availability** – Ensures the application remains available by maintaining the right number of instances.
- ✓ **Cost Optimization** – Reduces costs by removing unnecessary instances when demand is low.
- ✓ **Performance Efficiency** – Scales up resources to handle traffic spikes, preventing slowdowns.
- ✓ **Flexibility** – Works across multiple AWS services beyond EC2.

Configure Load Balancer with Auto Scaling and Creating Launch Template

Step 1: Sign in to AWS Console

Step 2: Navigate to **EC2 Dashboard** > scroll down and click "**Auto Scaling Groups**" under the "**Auto Scaling**" section.

Step 3: Create an Auto Scaling Group (ASG): Click the "Create Auto Scaling group" button > Enter a name in Auto Scaling Group

Step 4: Choose a Launch Template or Configuration

1. **Select a Launch Template** (recommended) or create a new one:
 - If you don't have a launch template, click "**Create a launch template**" and define:
 - Amazon Machine Image (AMI)
 - Instance type (t2.micro)
 - Key pair (for SSH access)
 - Security group (firewall rules) - HTTP & ssh > Click **Next**.

Step 5: Configure Auto Scaling Options

1. **Choose a VPC and Subnets** (for instance placement).
2. **Enable Load Balancer :**
 - **Application Load Balancer (ALB)**, can attach existing load balancer.

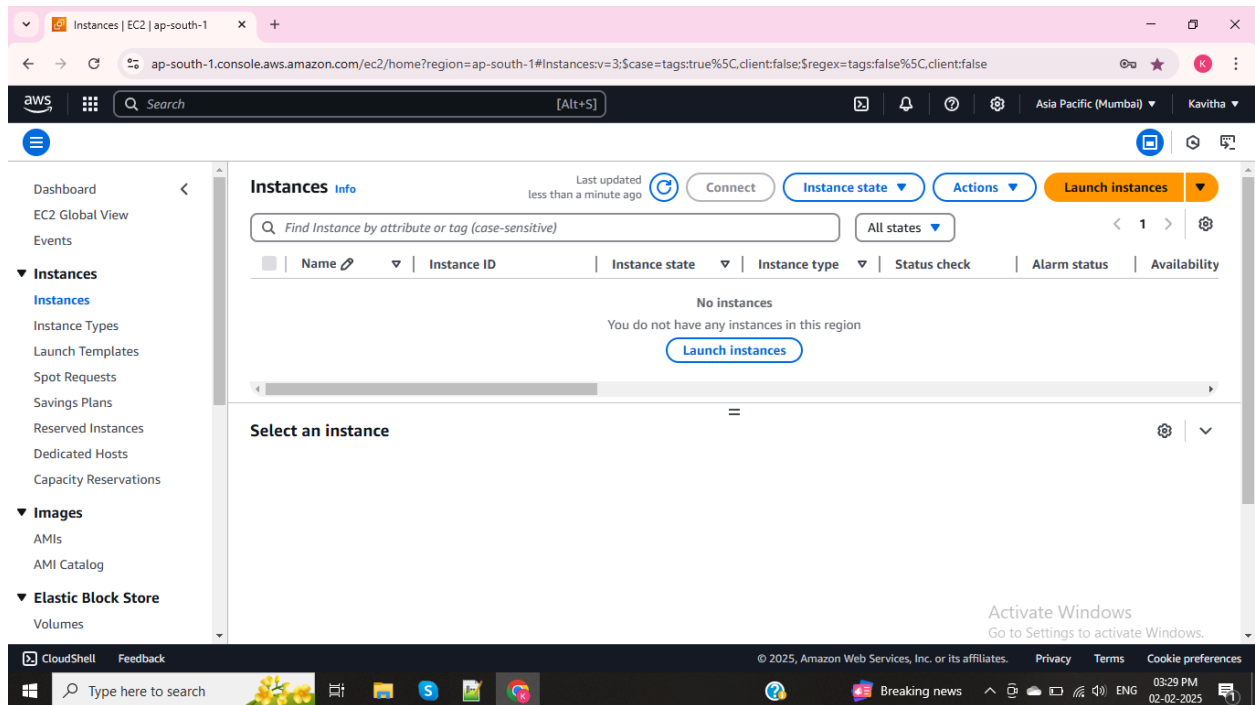
Step 6: Define Scaling Policies

1. Select **Scaling policy type**:
2. Define the minimum and maximum number of instances.
3. Click **Next**.

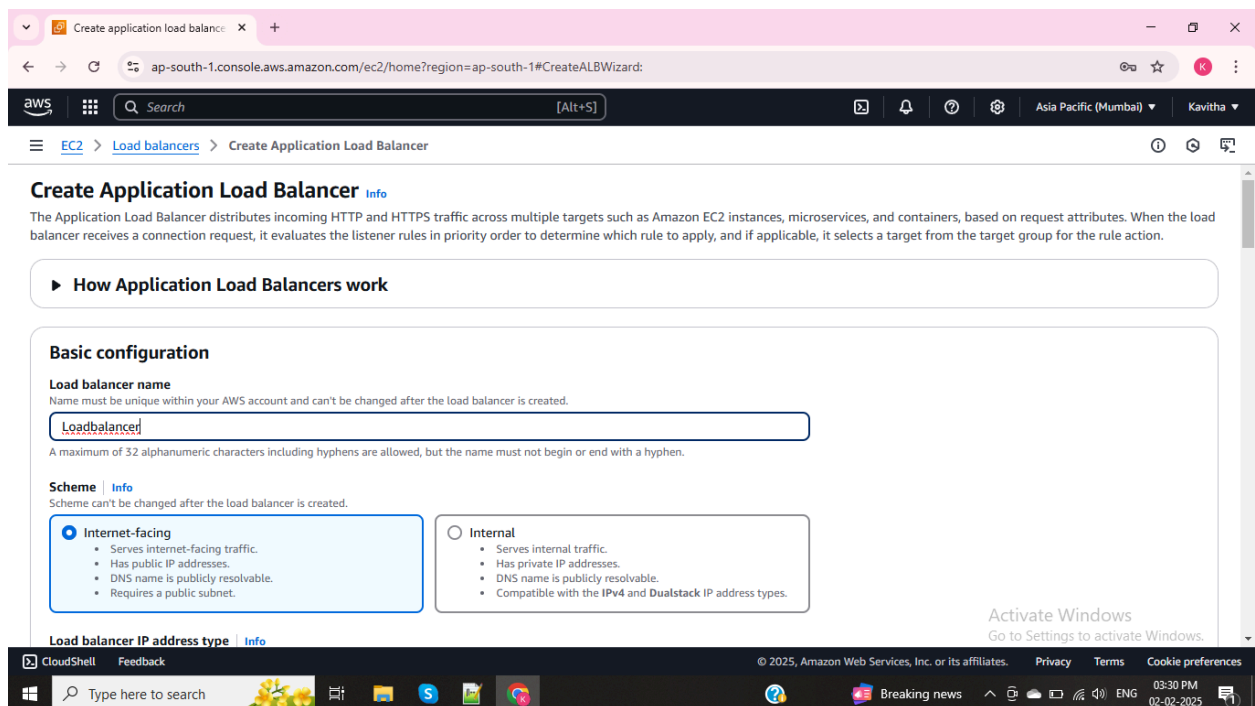
Step 7: Review and Create the Auto Scaling Group

1. Review all configurations.
2. Click "**Create Auto Scaling group**".

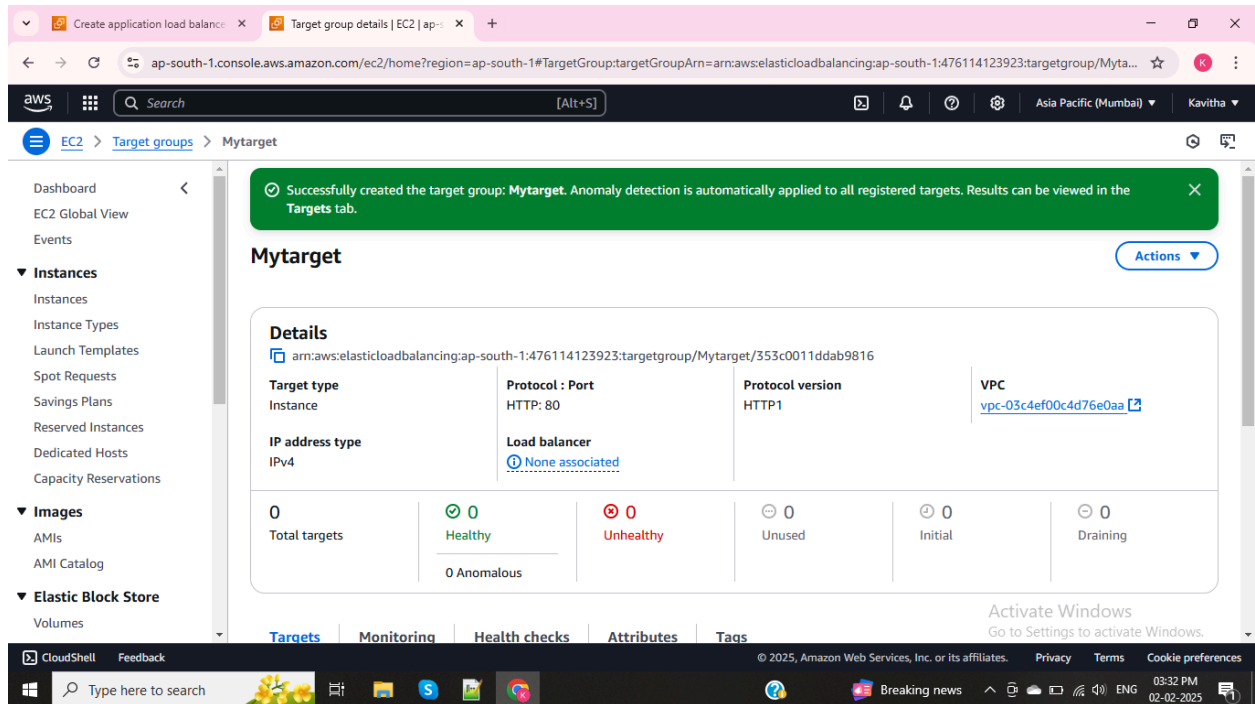
No Instance in EC2 Instance:



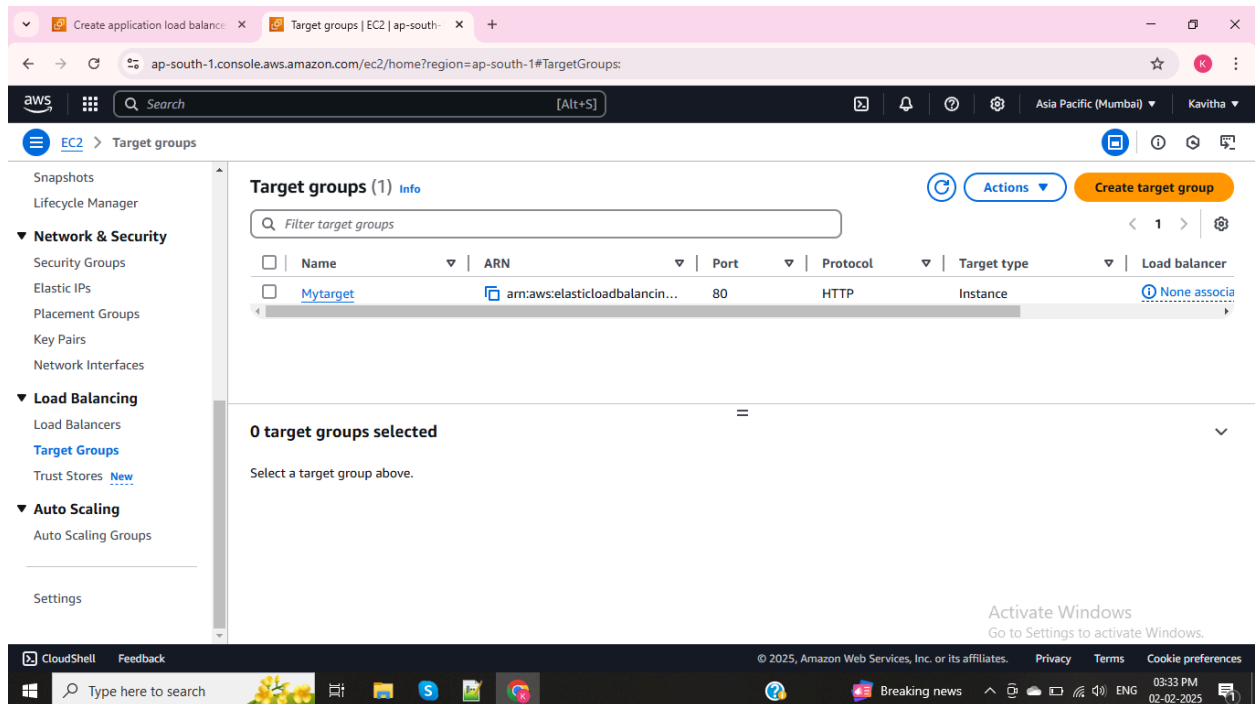
Creating Application LoadBalancer to attach with Auto scaling group



Creating Target group



OUTPUT



Load Balancer is created successfully

The screenshot shows the AWS Management Console for the 'ap-south-1' region. The 'Load balancers' page displays a table with one entry: 'Loadbalancer-1006499783....' in an 'Active' state, associated with VPC 'vpc-03c4ef00c4d76e0aa' and '3 Availability Zones'. The left sidebar shows the 'Network & Security' and 'Load Balancing' sections. The bottom of the screen shows a Windows taskbar with the date '02-02-2025' and time '03:37 PM'.

Name	DNS name	State	VPC ID	Availability Zones	Type
Loadbalancer	Loadbalancer-1006499783....	Active	vpc-03c4ef00c4d76e0aa	3 Availability Zones	application

Creating Launch template

The screenshot shows the 'Create launch template' page in the AWS Management Console. The 'Launch template name and description' section has 'Autoscaling' as the name and 'A prod webserver for MyApp' as the description. The 'Auto Scaling guidance' section is checked. The 'Summary' section on the right shows the configuration details. A 'Free tier' notification is displayed at the bottom right. The bottom of the screen shows a Windows taskbar with the date '02-02-2025' and time '03:35 PM'.

Launch template name and description

Launch template name - required:

Template version description:

Auto Scaling guidance

Select this if you intend to use this template with EC2 Auto Scaling

☒ Provide guidance to help me set up a template that I can use with EC2 Auto Scaling

Summary

- Software Image (AMI): -
- Virtual server type (instance type): -
- Firewall (security group): -
- Storage (volumes): -

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 750 hours of public IPv4 address

Configure group size and scaling policies

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 **Max desired capacity**

2 4

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** [Info](#)
Choose a CloudWatch metric and target value and let the scaling policy adjust the desired capacity in proportion to the metric's value.

CloudShell Feedback

Type here to search

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

31°C Mostly sunny 03:51 PM 02-02-2025

Reviewing all groups before created Auto Scaling

Step 1 Choose launch template

Step 2 Choose instance launch options

Step 3 - optional Integrate with other services

Step 4 - optional Configure group size and scaling

Step 5 - optional Add notifications

Step 6 - optional Add tags

Step 7 **Review**

Review [Info](#)

Step 1: Choose launch template [Edit](#)

Group details

Auto Scaling group name
Myscalinggroup

Launch template

Launch template	Version	Description
Autoscaling lt-0e2d0c8dd9b09682d	Default	

Step 2: Choose instance launch options [Edit](#)

Network

VPC
[vpc-03c4ef00c4d76e0aa](#)

CloudShell Feedback

Type here to search

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

31°C Mostly sunny 03:54 PM 02-02-2025

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

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 7 Review

Step 2: Choose instance launch options

[Edit](#)

Network

VPC
[vpc-03c4ef00c4d76e0aa](#)

Availability Zones and subnets

Availability Zone	Subnet	Subnet CIDR range
ap-south-1a	subnet-069adf10dc45d03f2	172.31.32.0/20
ap-south-1b	subnet-013b752352710a797	172.31.0.0/20
ap-south-1c	subnet-006c569ae60da613c	172.31.16.0/20

Availability Zone distribution
Balanced best effort

Instance type requirements

Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

Type here to search

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

31°C Mostly sunny 03:54 PM 02-02-2025

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

EC2 > Auto Scaling groups > Create Auto Scaling group

Step 3: Integrate with other services

[Edit](#)

Load balancing

Load balancer 1

Name	Type	Target group
Loadbalancer	Application/HTTP	Mytarget

VPC Lattice integration options

VPC Lattice target groups
-

Application Recovery Controller (ARC) zonal shift

ARC zonal shift
Disabled

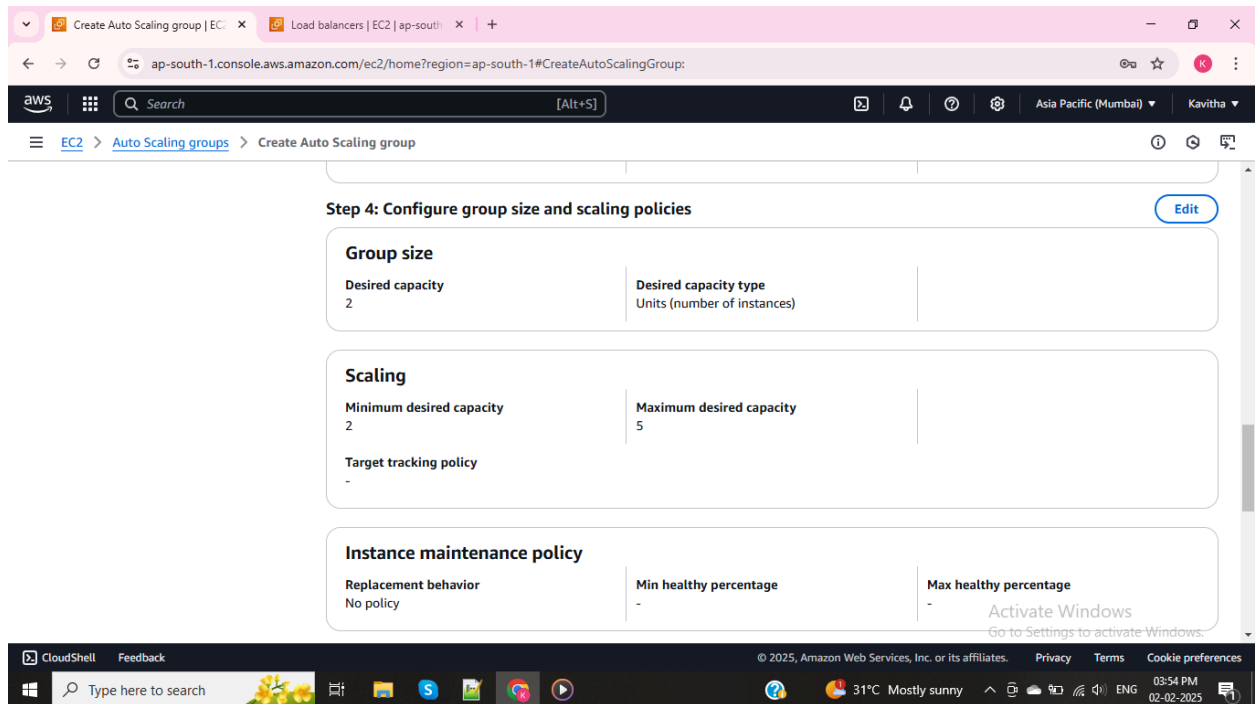
Activate Windows
Go to Settings to activate Windows.

CloudShell Feedback

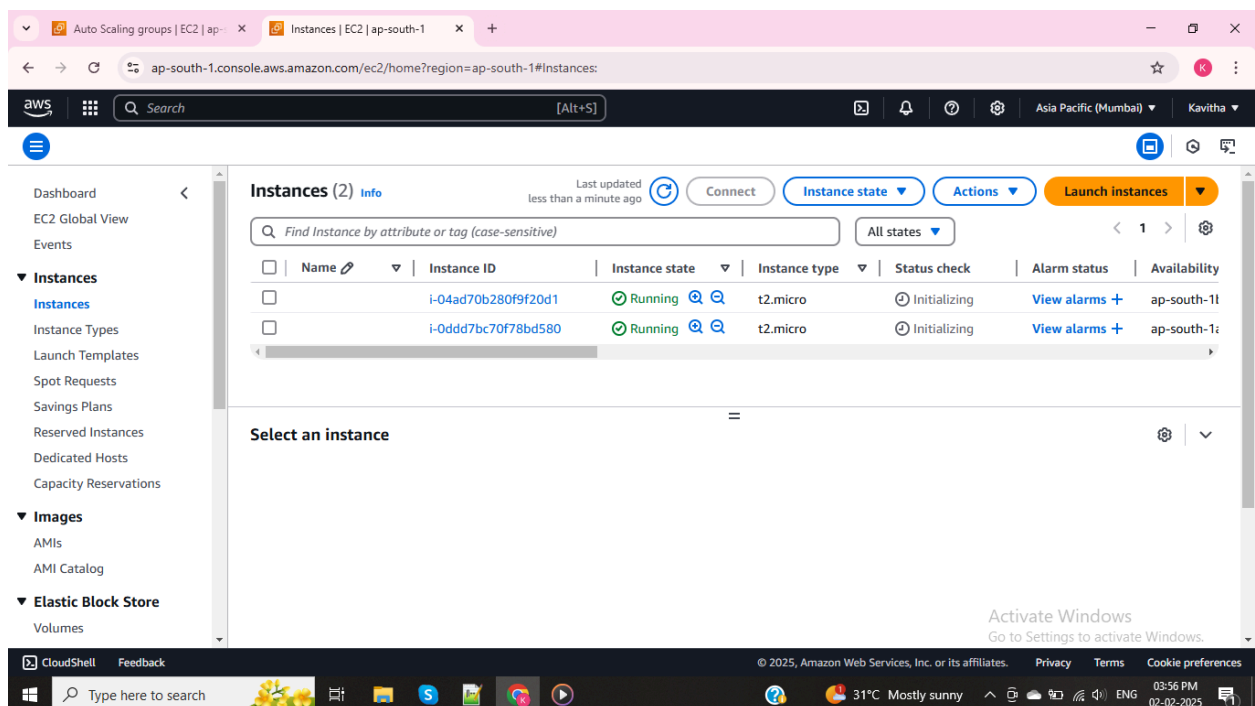
Type here to search

© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

31°C Mostly sunny 03:54 PM 02-02-2025



EC2 Instance launch successfully with the help of Launch template



Now, EC2 Instance terminating whether instance launch again with minimum desired count or not

The screenshot shows the AWS Management Console for the 'ap-south-1' region. A green notification banner at the top states: 'Successfully initiated termination (deletion) of i-04ad70b280f9f20d1, i-0ddd7bc70f78bd580'. Below this, the 'Instances' section shows 2 instances selected. The table lists the following instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
	i-04ad70b280f9f20d1	Shutting-d...	t2.micro	-	View alarms +	ap-south-1l
	i-0ddd7bc70f78bd580	Shutting-d...	t2.micro	-	View alarms +	ap-south-1i

The 'Monitoring' section below the table shows various metrics like CPU utilization, Network in/out, and Network packets, all with a value of 'No unit'. The bottom of the console shows the Windows taskbar with the time 03:56 PM on 02-02-2025.

After Instance terminated again automatically its created new two EC2 Instance (In scaling I give minimum desired capacity is 2)

The screenshot shows the AWS Management Console for the 'ap-south-1' region. A green notification banner at the top states: 'Successfully initiated termination (deletion) of i-04ad70b280f9f20d1, i-0ddd7bc70f78bd580'. Below this, the 'Instances' section shows 4 instances. The table lists the following instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
	i-01b658ff4e7bce430	Running	t2.micro	Initializing	View alarms +	ap-south-1l
	i-04ad70b280f9f20d1	Terminated	t2.micro	-	View alarms +	ap-south-1l
	i-0ddd7bc70f78bd580	Terminated	t2.micro	-	View alarms +	ap-south-1i
	i-09bd601e3d69cd3a0	Running	t2.micro	2/2 checks pass	View alarms +	ap-south-1i

The 'Monitoring' section below the table shows various metrics like CPU utilization, Network in/out, and Network packets, all with a value of 'No unit'. The bottom of the console shows the Windows taskbar with the time 04:00 PM on 02-02-2025.