St. JOSEPH'S
GROUP OF INSTITUTIONS
OMR, CHENNAI - 119

aws

Placement Empowerment Program

Cloud Computing and DevOps Centre

IMPLEMENT AUTO-SCALING IN THE CLOUD

(Set up an auto-scaling group for your cloud VMs to handle variable workloads)

NAME: Swathika K

DEPARTMENT: AML

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY

AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

INTRODUCTION:

Auto Scaling is an AWS services that ensure high availability, fault tolerance, and cost efficiency in cloud applications.

Auto Scaling (ASG - Auto Scaling Group):

Auto Scaling automatically adjusts the number of EC2 instances in response to demand.

What Auto Scaling Does:

- Increases Instances (Scale Out): When traffic or CPU usage increases, Auto Scaling launches new EC2 instances.
- Decreases Instances (Scale In): When traffic decreases, Auto Scaling terminates extra instances to reduce costs.
- Ensures Availability: If an instance fails (due to crash, hardware failure, etc.), Auto Scaling replaces it.
- Optimizes Costs: Ensures you only use the required number of instances at any given time.
- Elastic Load Balancer distributes incoming traffic across multiple EC2 instances to improve performance and availability.

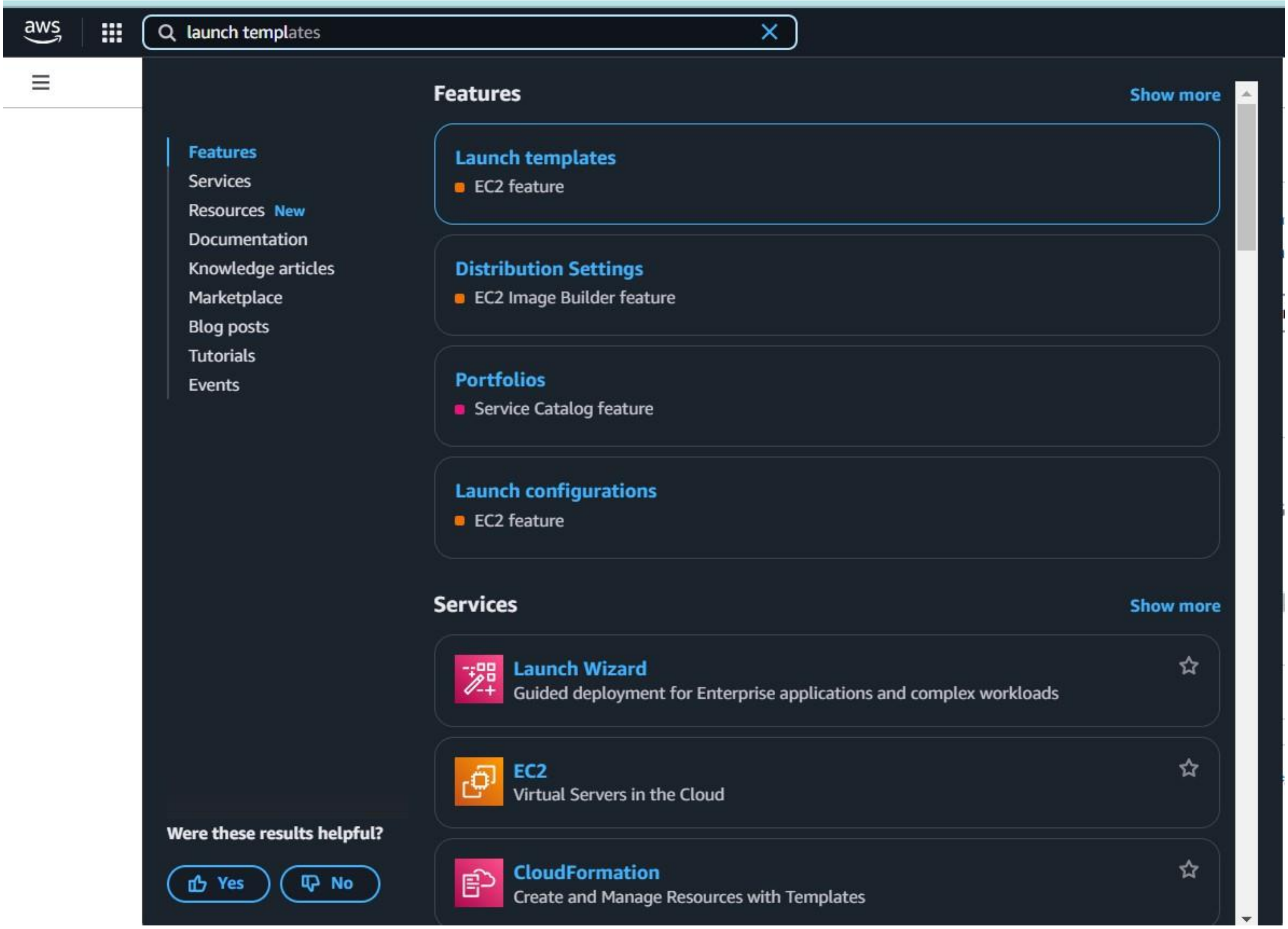
IMPORTANCE:

- Auto Scaling Group monitors load (CPU, traffic) and adds/removes instances as needed.
- If traffic increases, ASG launches more instances, and ELB distributes traffic evenly.
- If traffic decreases, ASG removes extra instances, reducing costs.
- Result: High availability, better performance, and optimized cloud costs!

STEP BY STEP OVERVIEW:

Step 1: LAUNCH TEMPLATES

- Login into your AWS console and search for launch templates.



Features

Launch templates
EC2 feature

Distribution Settings
EC2 Image Builder feature

Portfolios
Service Catalog feature

Launch configurations
EC2 feature

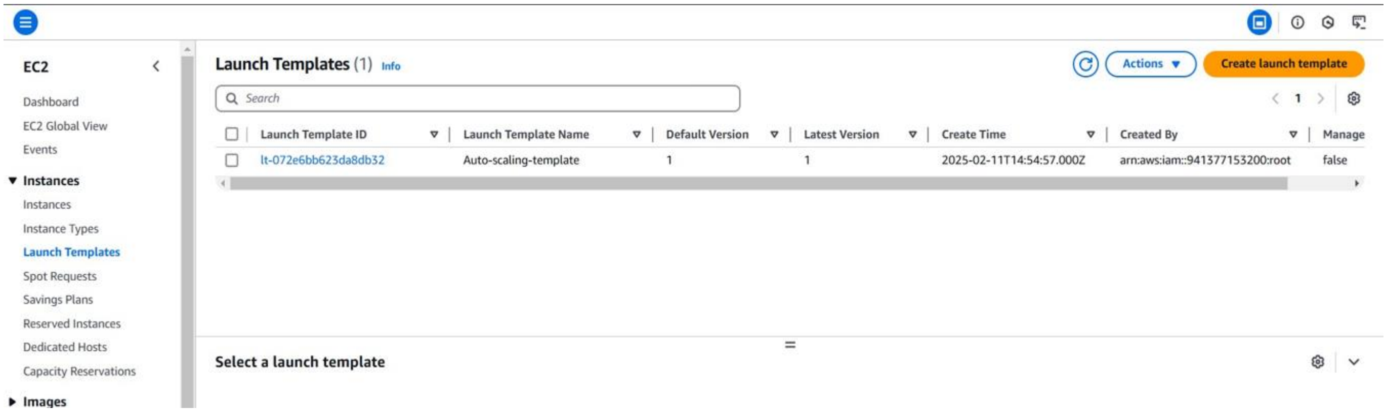
Services

Launch Wizard
Guided deployment for Enterprise applications and complex workloads

EC2
Virtual Servers in the Cloud

CloudFormation
Create and Manage Resources with Templates

- Navigate to it and click on launch template.
- Specify the template name, AMI image (Amazon Linux), Instance type (t2.micro), Key pair.
- Mainly, configure the security groups and add the security group role(such as: HTTP, ssh)
- Now launch the template.



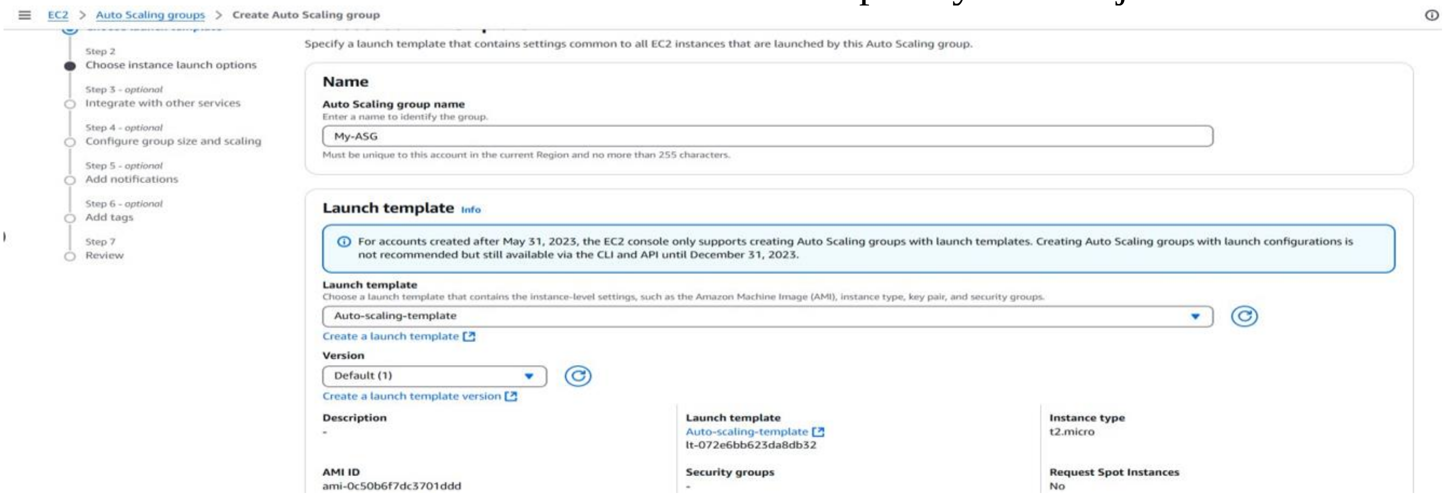
Launch Templates (1)

☐	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Manage
☐	lt-072e6bb623da8db32	Auto-scaling-template	1	1	2025-02-11T14:54:57.000Z	arn:aws:iam::941377153200:root	false

Select a launch template

Step 2: AUTO SCALING GROUPS

- Go to your EC2 dashboard, on the left sidebar you will find ‘Auto Scaling groups’. Click on create.
- Give the name to the ASG and select the template you have just created.



EC2 > Auto Scaling groups > Create Auto Scaling group
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

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

Launch template Info

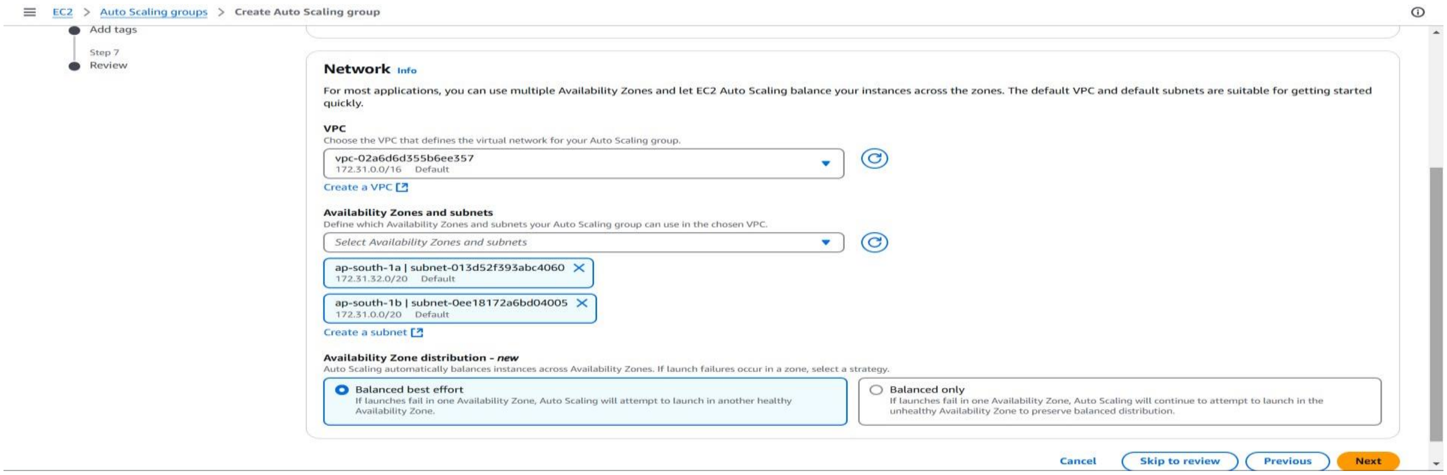
For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
Auto-scaling-template
Create a launch template

Version
Default (1)
Create a launch template version

Description	**Launch template**	**Instance type**
-	Auto-scaling-template lt-072e6bb623da8db32	t2.micro
AMI ID	**Security groups**	**Request Spot Instances**
ami-0c50b6f7dc3701ddd	-	No

Choose your VPC or the default VPC and select two subnets of two different availability zone.



Network Info

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

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-02a6d6d355b6ee357
172.31.0.0/16 Default
Create a VPC

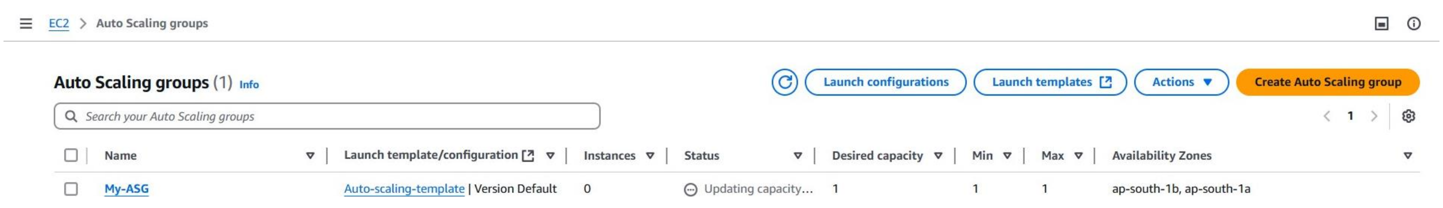
Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets
ap-south-1a | subnet-013d52f393abc4060
172.31.32.0/20 Default
ap-south-1b | subnet-0ee18172a6bd04005
172.31.0.0/20 Default
Create a subnet

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.

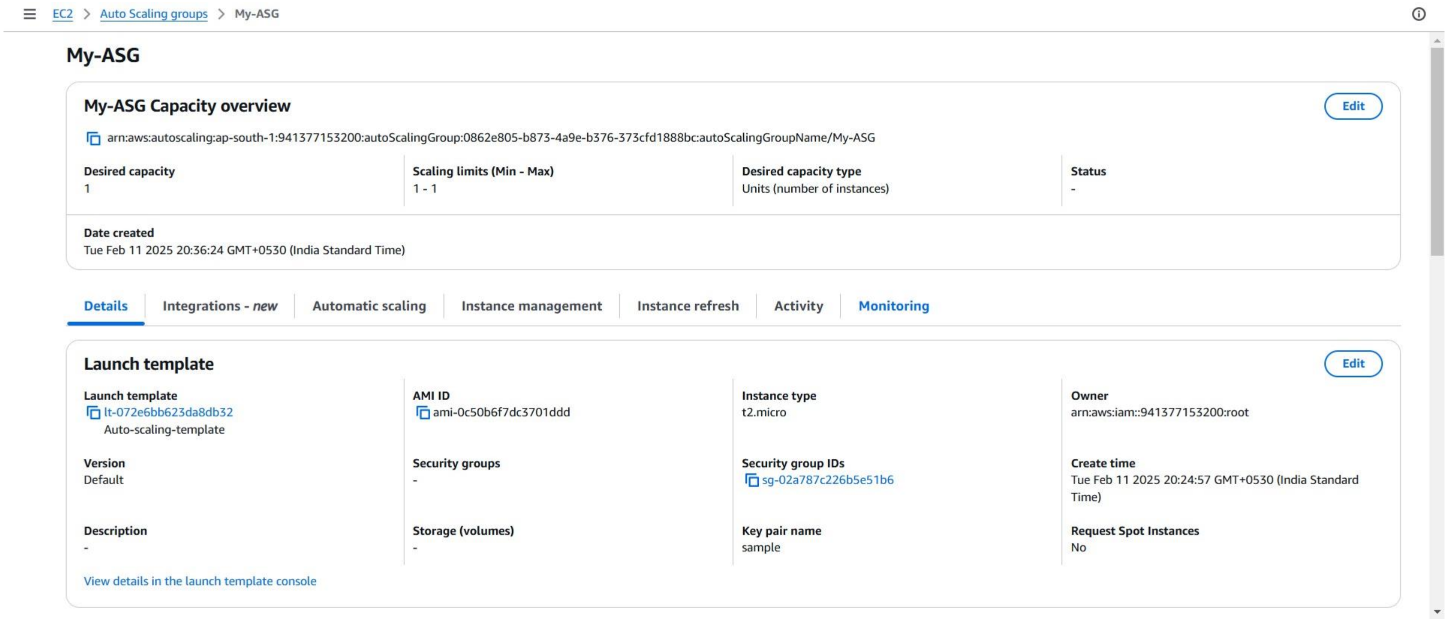
○ Balanced only
If launches fail in one Availability Zone, Auto Scaling will continue to attempt to launch in the unhealthy Availability Zone to preserve balanced distribution.

- Leave the next setting as default. Now, review and click on create Auto Scaling group.



EC2 > Auto Scaling groups
Auto Scaling groups (1) Info
Launch configurations
Launch templates
Actions ▼
Create Auto Scaling group
Search your Auto Scaling groups

□	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
□	My-ASG	Auto-scaling-template | Version Default	0	Updating capacity...	1	1	1	ap-south-1b, ap-south-1a



My-ASG

My-ASG Capacity overview

arn:aws:autoscaling:ap-south-1:941377153200:autoScalingGroup:0862e805-b873-4a9e-b376-373cfd1888bc:autoScalingGroupName/My-ASG

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

Date created
Tue Feb 11 2025 20:36:24 GMT+0530 (India Standard Time)

Launch template

Launch template lt-072e6bb623da8db32 Auto-scaling-template	**AMI ID** ami-0c50b6f7dc3701ddd	**Instance type** t2.micro	**Owner** arn:aws:iam::941377153200:root
Version Default	**Security groups** -	**Security group IDs** sg-02a787c226b5e51b6	**Create time** Tue Feb 11 2025 20:24:57 GMT+0530 (India Standard Time)
Description -	**Storage (volumes)** -	**Key pair name** sample	**Request Spot Instances** No

Step 3: TESTING AUTO SCALING

Important Note: Do Not Perform This Test If You Want to Avoid Costs:

1. Launching and running additional EC2 instances will incur charges beyond the AWS Free Tier.

2. Simulating high CPU usage and triggering scaling may increase costs temporarily due to additional resource allocation.

TESTING:

1. Simulate High CPU Usage on an EC2 Instance: Connect to one of your EC2 instances in the Auto Scaling Group using SSH.
2. Run a command to create artificial CPU load. For example: sudo yum install -y stress, stress--cpu 2--timeout 300.

This command will utilize 2 CPU cores for 5 minutes, simulating high CPU usage.

MONITOR SCALING ACTIVITIES:

Navigate to the AWS Management Console > EC2 Dashboard > Auto Scaling Groups.

Select your Auto Scaling Group and go to the Activity History tab.

Check if a new instance is being launched based on your scaling policy (e.g., CPU utilization exceeding 50%).

TERMINATE THE STRESS TEST:

Once testing is done, stop the CPU load by pressing Ctrl+C in the terminal or by terminating the stress process.

VERIFY SCALING DOWN:

After the CPU usage drops, monitor the Auto Scaling Group again to confirm that unnecessary instances are terminated, returning to the desired capacity.

CONCLUSION:

By completing this PoC, we will be able to:

- Implement Auto Scaling group and launch the Template.
- Attain dynamic scaling and monitoring.
- Most importantly, we should be aware of Costs.