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: CHANDRU S

Department: INFORMATION TECHNOLOGY

St. JOSEPH'S COLLEGE OF ENGINEERING | St. JOSEPH'S INSTITUTE OF TECHNOLOGY
AUTONOMOUS INSTITUTIONS, AFFILIATED TO ANNA UNIVERSITY

Introduction

Modern applications often experience fluctuating workloads, making optimal performance and availability crucial. Auto Scaling, a feature provided by cloud platforms like AWS, dynamically adjusts computing resources based on demand changes. This Proof of Concept (PoC) demonstrates how to set up an Auto Scaling Group (ASG) for virtual machines (VMs) to efficiently handle varying workloads. The process includes defining launch configurations, setting scaling policies, and testing automatic scaling based on CPU usage.

Overview

This PoC focuses on building a scalable architecture using AWS Auto Scaling Groups. The key steps include:

1. **Defining a Launch Template:** Configuring VMs with specifications such as instance type, AMI, key pairs, and security groups.
2. **Creating an Auto Scaling Group:** Setting the initial group size and linking it to the launch template for dynamic instance management.
3. **Configuring Scaling Policies:** Establishing metrics like CPU utilization to trigger scaling actions, such as scaling up during high CPU usage.
4. **Testing Auto Scaling:** Simulating high CPU load to verify that the ASG launches additional instances when demand increases.

This PoC demonstrates the reliability, flexibility, and cost-efficiency of dynamic scaling in a cloud environment.

Objectives

The primary objectives of this PoC are to:

1. Implement an Auto Scaling Group (ASG) to manage workloads effectively.
2. Define and configure a Launch Template for virtual machines.
3. Set up and test scaling policies based on predefined metrics, such as CPU utilization.
4. Validate the scaling process by simulating real-world scenarios (e.g., high CPU usage).

By completing this PoC, users will gain hands-on experience with Auto Scaling and understand its importance in ensuring application availability and cost management.

Importance

Key benefits of implementing AWS Auto Scaling include:

- **Improved Application Availability:** Auto Scaling ensures applications remain available during traffic spikes by automatically adding more VMs.
- **Cost Optimization:** Resources are scaled down during low traffic periods, reducing unnecessary costs.

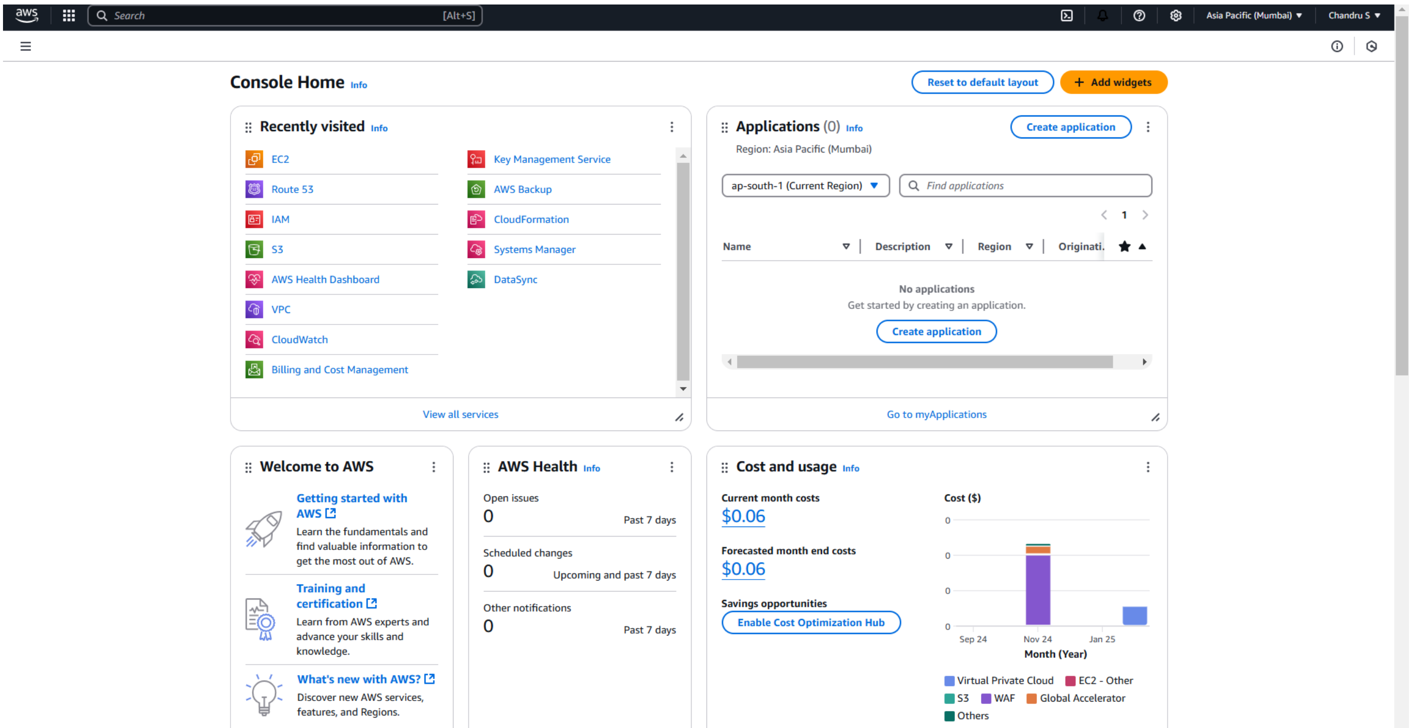
- **Efficient Resource Utilization:** Resources are provisioned based on actual demand, preventing over-provisioning and underutilization.
- **Resilience to Failures:** Unhealthy instances are automatically replaced, ensuring consistent performance.
- **Real-World Relevance:** Managing variable workloads is a critical cloud computing skill aligned with industry practices.

Step-by-Step Overview

Step 1:

Access AWS Management Console

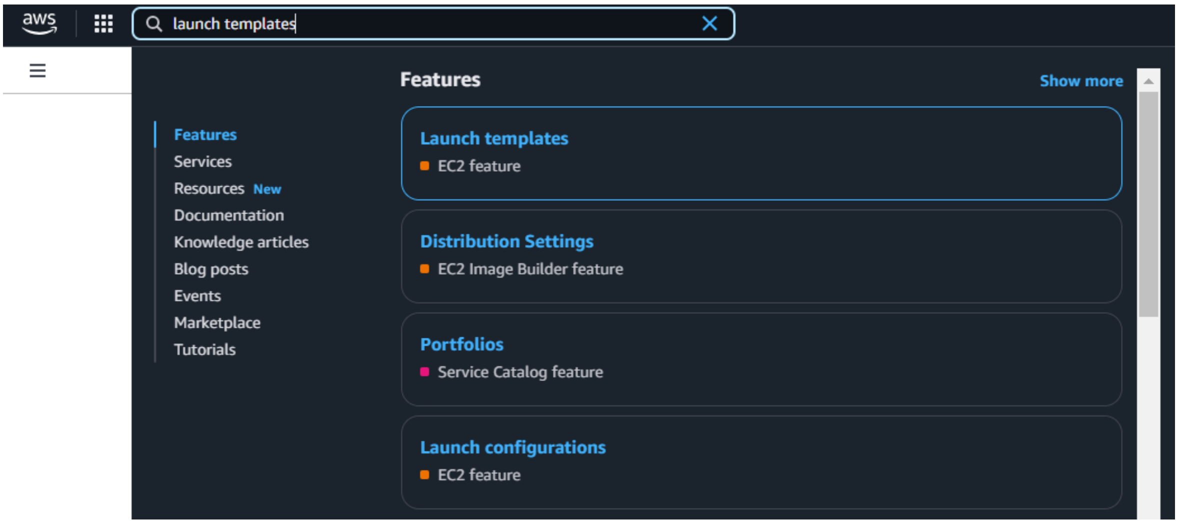
- Log in to the AWS Management Console with your credentials.



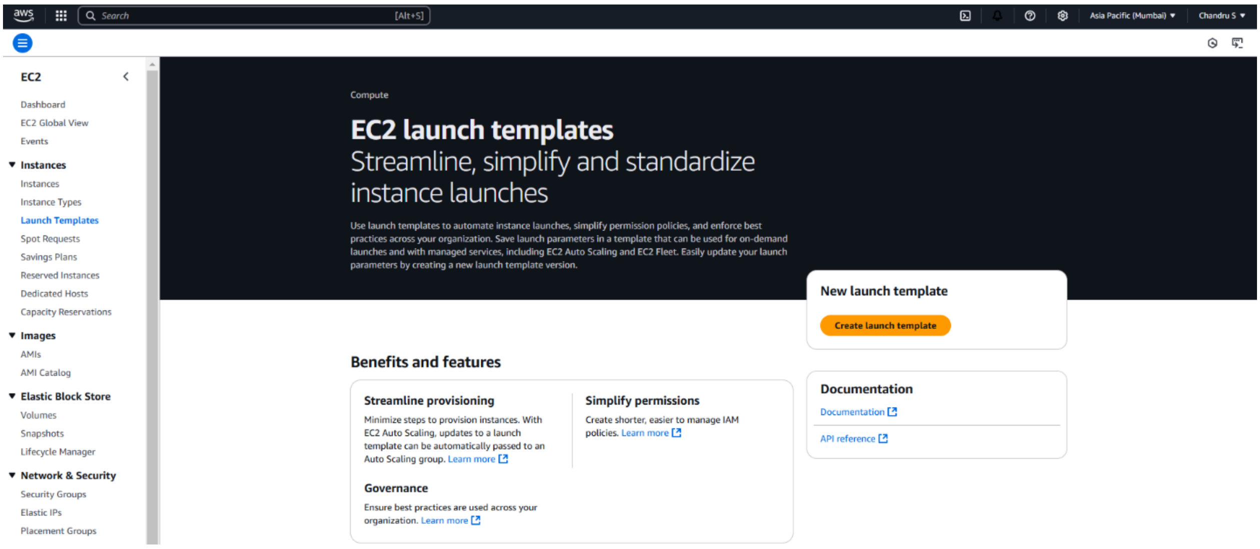
Step 2:

Create Launch Template

1. Search for **Launch Templates** in the AWS Console.



2. Click Create Launch Template.



Compute

EC2 launch templates

Streamline, simplify and standardize instance launches

Use launch templates to automate instance launches, simplify permission policies, and enforce best practices across your organization. Save launch parameters in a template that can be used for on-demand launches and with managed services, including EC2 Auto Scaling and EC2 Fleet. Easily update your launch parameters by creating a new launch template version.

New launch template

Benefits and features

Streamline provisioning

Minimize steps to provision instances. With EC2 Auto Scaling, updates to a launch template can be automatically passed to an Auto Scaling group. Learn more

Simplify permissions

Create shorter, easier to manage IAM policies. Learn more

Governance

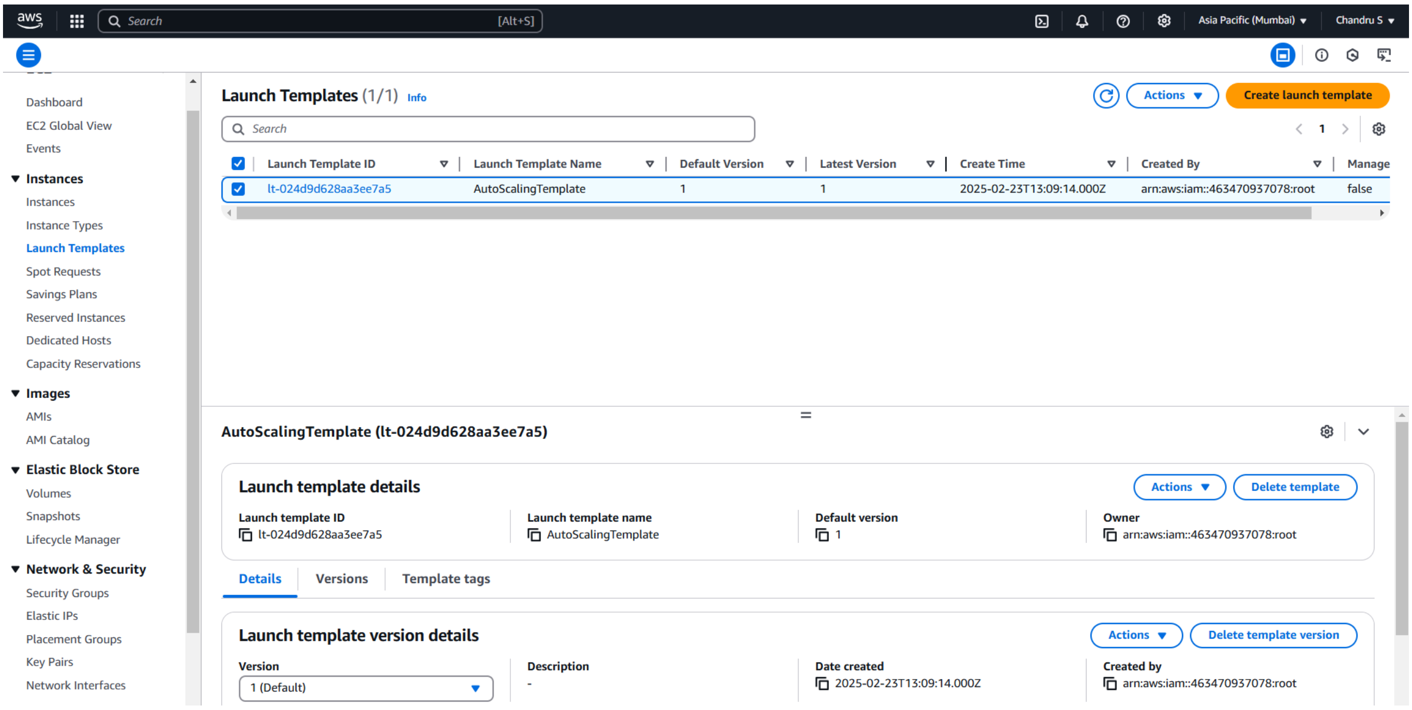
Ensure best practices are used across your organization. Learn more

Documentation

3. Configure the template:

   - **Template Name:** AutoScalingTemplate
   - **AMI:** Amazon Linux 2 (or any default image)
   - **Instance Type:** t2.micro (Free-tier eligible)
   - **Key Pair:** Choose an existing key pair or create a new one for SSH access.
   - **Security Group:** Allow HTTP (port 80) and SSH (port 22).

4. Review the configuration and click Create Launch Template.



Launch Templates (1/1) Info

☑	Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By	Manage
☑	lt-024d9d628aa3ee7a5	AutoScalingTemplate	1	1	2025-02-23T13:09:14.000Z	arn:aws:iam::463470937078:root	false

AutoScalingTemplate (lt-024d9d628aa3ee7a5)

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-024d9d628aa3ee7a5	AutoScalingTemplate	1	arn:aws:iam::463470937078:root

Launch template version details

Version	Description	Date created	Created by
1 (Default)	-	2025-02-23T13:09:14.000Z	arn:aws:iam::463470937078:root

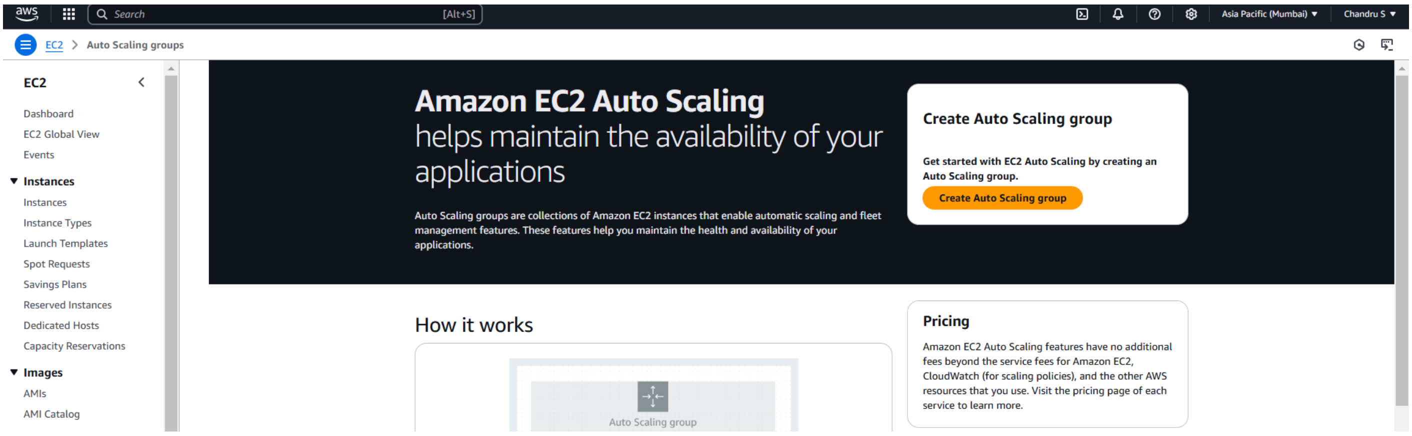
Step 3:

Create Auto Scaling Group

1. Go to the **EC2 Dashboard**.

2. In the left sidebar, click **Auto Scaling Groups**.

3. Click Create Auto Scaling Group.



Amazon EC2 Auto Scaling
helps maintain the availability of your applications

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.

Create Auto Scaling group

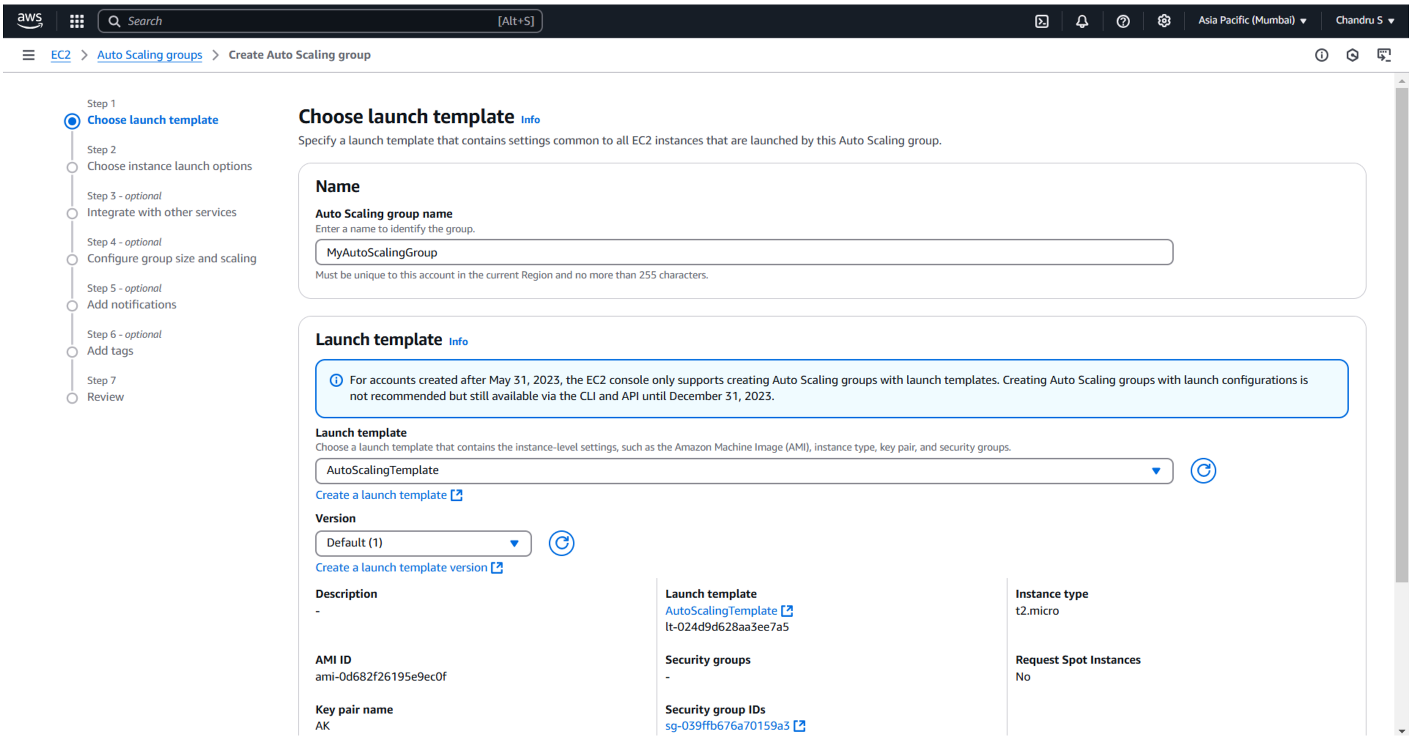
Get started with EC2 Auto Scaling by creating an Auto Scaling group.

How it works

Pricing

Amazon EC2 Auto Scaling features have no additional fees beyond the service fees for Amazon EC2, CloudWatch (for scaling policies), and the other AWS resources that you use. Visit the pricing page of each service to learn more.

4. Configure the group:
   - **Group Name:** MyAutoScalingGroup
   - **Launch Template:** Select AutoScalingTemplate.



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

Choose launch template Info

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

Launch template Info

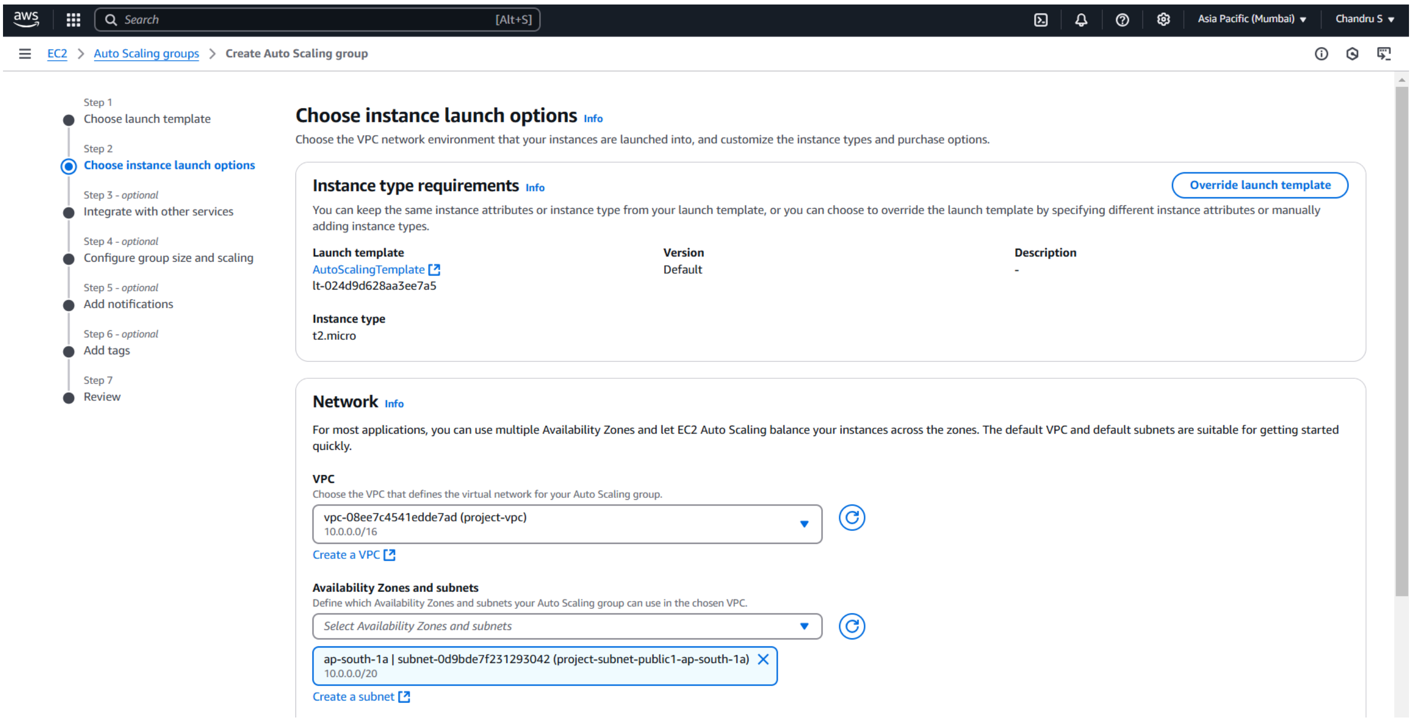
ⓘ For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

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.
AutoScalingTemplate ▾
Create a launch template

Version
Default (1) ▾
Create a launch template version

Description -	**Launch template** AutoScalingTemplate lt-024d9d628aa3ee7a5	**Instance type** t2.micro
AMI ID ami-0d682f26195e9ec0f	**Security groups** -	**Request Spot Instances** No
Key pair name AK	**Security group IDs** sg-039ffb676a70159a3	

- **VPC and Subnets:** Use the default VPC and select at least two subnets in different Availability Zones for high availability.



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

Choose instance launch options Info

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

Instance type requirements Info

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 AutoScalingTemplate lt-024d9d628aa3ee7a5	**Version** Default	**Description** -
Instance type t2.micro		

Network Info

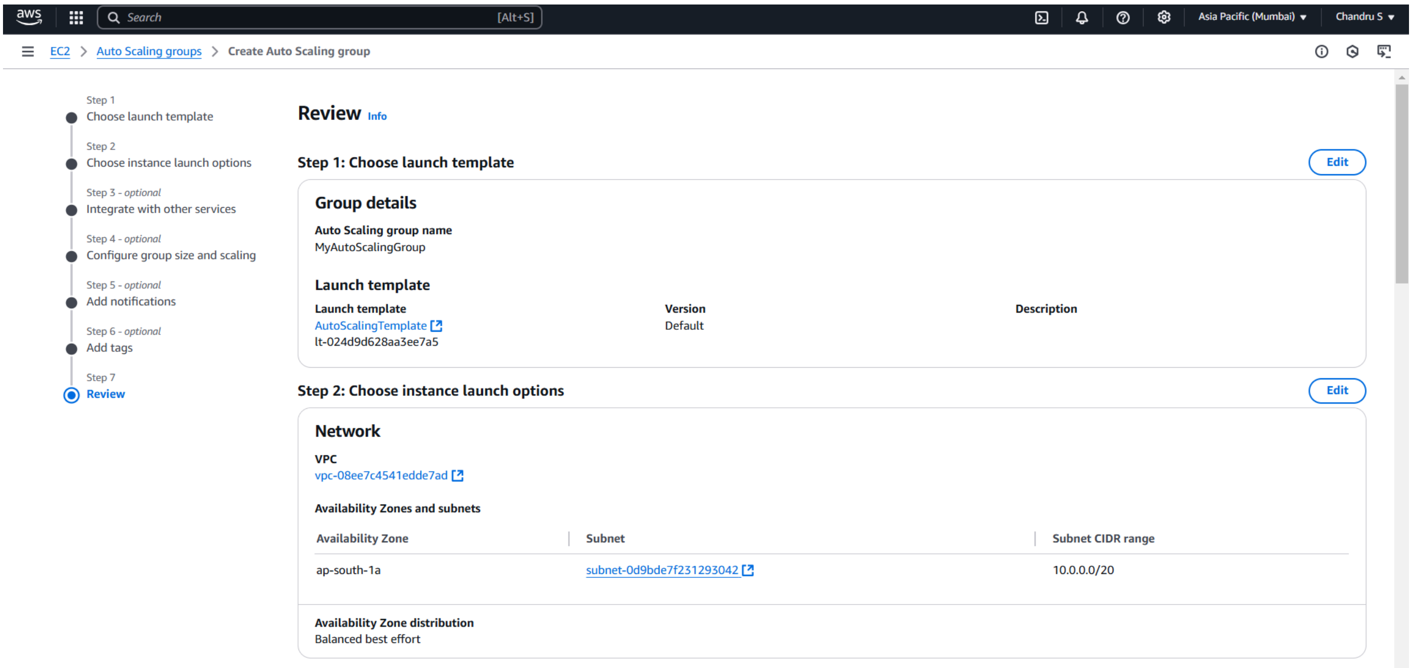
For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-08ee7c4541edde7ad (project-vpc)
10.0.0.0/16
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-0d9bde7f231293042 (project-subnet-public1-ap-south-1a) ✕
10.0.0.0/20
Create a subnet

5. Leave other settings as **default** and click **Next**.

6. Review the configuration and click **Create Auto Scaling Group**.



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

Group details

Auto Scaling group name
MyAutoScalingGroup

Launch template

Launch template	Version	Description
AutoScalingTemplate lt-024d9d628aa3ee7a5	Default	

Step 2: Choose instance launch options

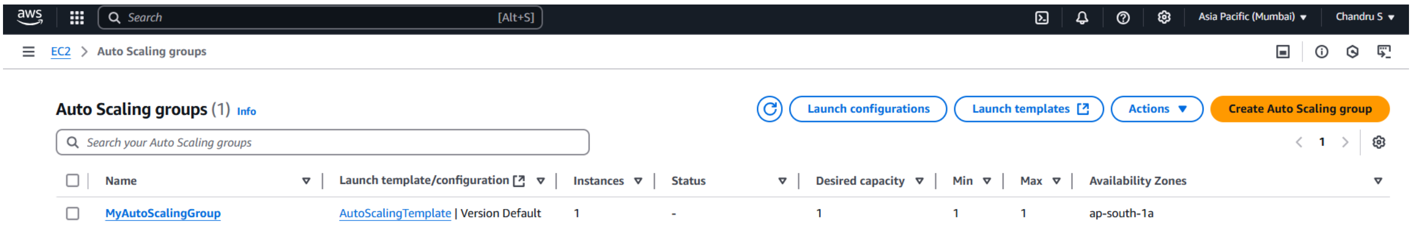
Network

VPC
vpc-08ee7c4541edde7ad

Availability Zones and subnets

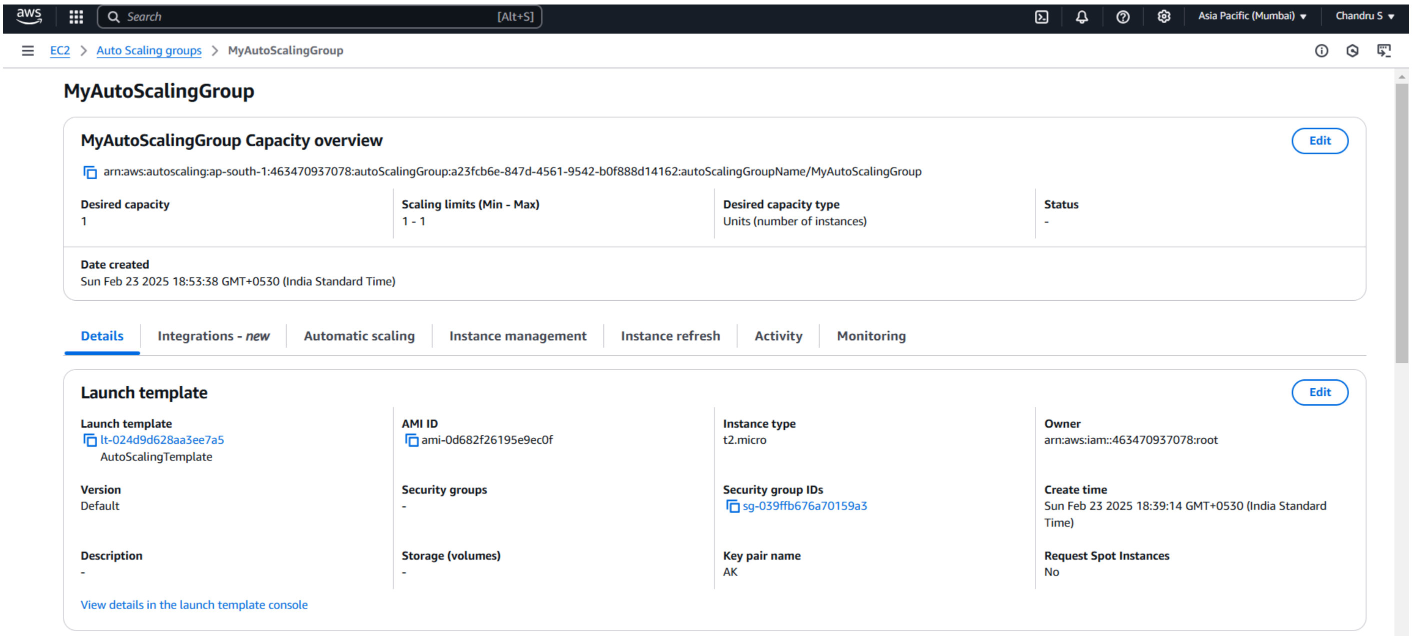
Availability Zone	Subnet	Subnet CIDR range
ap-south-1a	subnet-0d9bde7f231293042	10.0.0.0/20

Availability Zone distribution
Balanced best effort



Auto Scaling groups (1) Info

□	Name	Launch template/configuration	Instances	Status	Desired capacity	Min	Max	Availability Zones
□	MyAutoScalingGroup	AutoScalingTemplate | Version Default	1	-	1	1	1	ap-south-1a



MyAutoScalingGroup

MyAutoScalingGroup Capacity overview

arn:aws:autoscaling:ap-south-1:463470937078:autoScalingGroup:a23fcb6e-847d-4561-9542-b0f888d14162:autoScalingGroupName/MyAutoScalingGroup

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

Date created
Sun Feb 23 2025 18:53:38 GMT+0530 (India Standard Time)

Launch template

Launch template	AMI ID	Instance type	Owner
lt-024d9d628aa3ee7a5 AutoScalingTemplate	ami-0d682f26195e9ec0f	t2.micro	arn:aws:iam::463470937078:root
Version	**Security groups**	**Security group IDs**	**Create time**
Default	-	sg-039ffb676a70159a3	Sun Feb 23 2025 18:39:14 GMT+0530 (India Standard Time)
Description	**Storage (volumes)**	**Key pair name**	**Request Spot Instances**
-	-	AK	No

View details in the launch template console

Step 4:

Testing Auto Scaling

Important Note: Avoid this test if you wish to prevent additional AWS costs.

1. **Simulate High CPU Usage**
   - Connect to an EC2 instance in the Auto Scaling Group using SSH.
   - Install the stress package and simulate CPU load:

```
sudo yum install -y stress
stress --cpu 2 --timeout 300
```

   - This command utilizes 2 CPU cores for 5 minutes to simulate high usage.

2. **Monitor Scaling Activities**
   - Go to **AWS Management Console > EC2 Dashboard > Auto Scaling Groups**.
   - Select your Auto Scaling Group and navigate to the Activity History tab.
   - Check if new instances are launched based on the scaling policy (e.g., CPU utilization exceeding 50%).

3. **Terminate the Stress Test**
   - Stop the CPU load by pressing Ctrl+C in the terminal or terminating the stress process.

4. **Verify Scaling Down**
   - After CPU usage drops, check the Auto Scaling Group to confirm that unnecessary instances are terminated, returning the group to its desired capacity.

Outcome

This Proof of Concept successfully demonstrated how AWS Auto Scaling dynamically manages EC2 instances based on workload demand, ensuring efficient resource utilization and cost-effectiveness. Key outcomes include:

1. **Launch Template and ASG Setup:** Successfully created a launch template and Auto Scaling Group with scaling policies.
2. **Dynamic Scaling and Monitoring:** Implemented scaling policies triggered by CPU utilization and verified scaling actions using the Activity History.
3. **Cost Awareness:** Highlighted potential costs of running additional instances beyond the AWS Free Tier and ensured optimal resource usage.

By completing this PoC, users gain hands-on experience with AWS Auto Scaling, enhancing their cloud computing skills for real-world applications.