

**Placement Empowerment Program**

***Cloud Computing and DevOps Centre***

# Implement Auto-scaling in the CloudSet up an auto scaling group for your cloud VMs to handle variable workloads.

Name: MADHUMETHRA A P Department: IT



**Introduction**

As modern applications face varying workloads, ensuring optimal performance and availability is critical. Auto Scaling, a feature provided by cloud platforms like AWS, dynamically adjusts computing resources in response to demand changes. This Proof of Concept (PoC) demonstrates how to set up an Auto Scaling Group (ASG) for virtual machines (VMs) to handle fluctuating workloads effectively. It explores defining launch configurations, setting scaling policies, and testing automatic scaling based on CPU usage.

**Overview**

This PoC focuses on implementing a scalable architecture using AWS Auto Scaling Groups. The workflow includes: 1. Defining a Launch Template: Configuring virtual machines (VMs) with required specifications like instance type, AMI, key pairs, and security groups. 2. Creating an Auto Scaling Group: Setting initial group size and linking it to the launch template to manage instances dynamically. 3. Configuring Scaling Policies: Setting up metrics like CPU utilization to trigger scaling actions (e.g., scaling up during high CPU usage). 4. Testing Auto Scaling: Simulating high CPU load to verify that the ASG launches additional instances to handle demand. This PoC will demonstrate the reliability, flexibility, and cost efficiency of dynamic scaling in a cloud environment.

**Objective**

The primary objective of this PoC is 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, the goal is to gain hands-on experience with Auto Scaling and to understand its importance in ensuring application availability and cost management.

**Importance**

1. Improved Application Availability: Auto Scaling ensures that applications remain available even during traffic spikes by automatically adding more VMs to meet demand.

2. Cost Optimization: It dynamically reduces the number of VMs during low traffic periods, minimizing unnecessary costs.

3. Efficient Resource Utilization: By scaling resources based on actual demand, Auto Scaling prevents over-provisioning and underutilization.

4. Resilience to Failures: Auto Scaling can replace unhealthy instances automatically, ensuring consistent application performance.

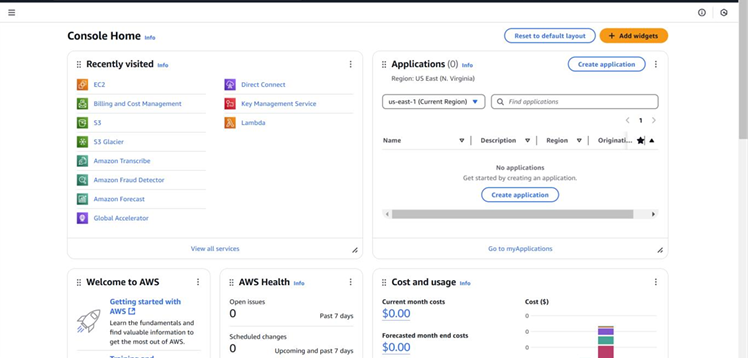
5. Real-World Relevance: The ability to manage variable workloads is a critical skill in cloud computing and aligns with industry practices.

**Step-by-Step Overview**

Step1:

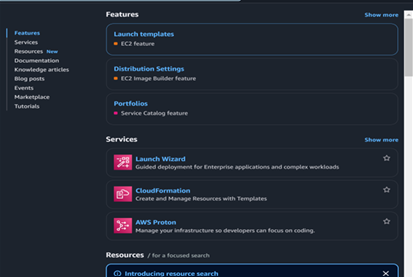
Go to AWS Management Console.

Enter your username and password to log in.



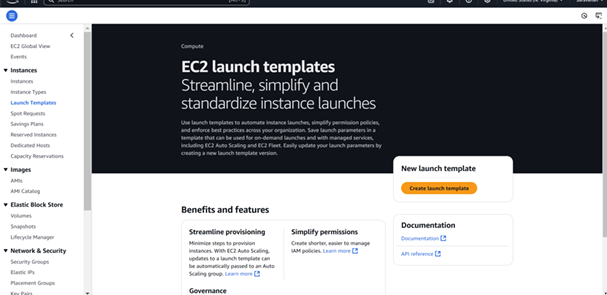
Step 2:

Search for Launch Templates.



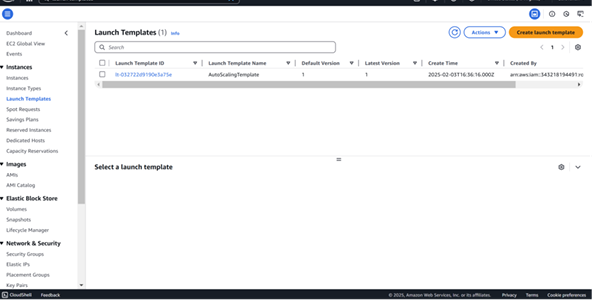
Step 3:

Click on the Create launch template.



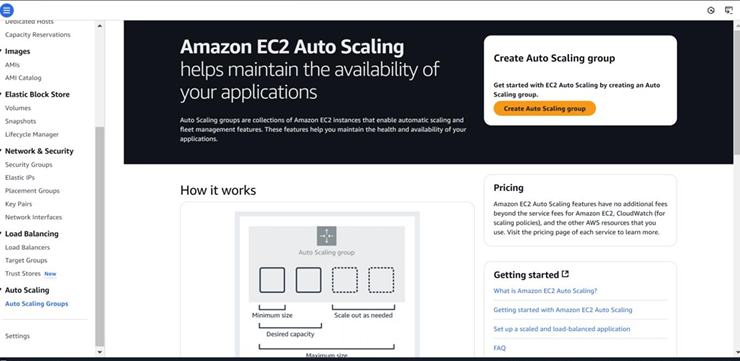
Step 4:

Create a Launch Template named AutoScalingTemplate using an Amazon Machine Image (AMI) like Amazon Linux 2 or any default image, and choose an instance type such as t2.micro for free-tier eligibility. Select an existing key pair (or create a new one) to enable SSH access, and configure a security group that allows HTTP (port 80) and SSH (port 22). Once all details are filled out, click Create launch template to complete the setup.



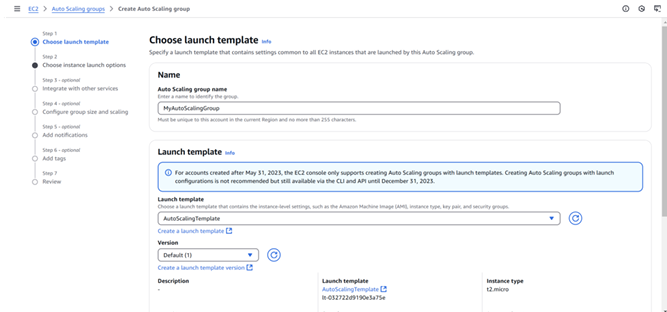
Step 5:

Go to the EC2 Dashboard . On the left sidebar, click on Auto Scaling Groups. Click on Create an Auto Scaling group.



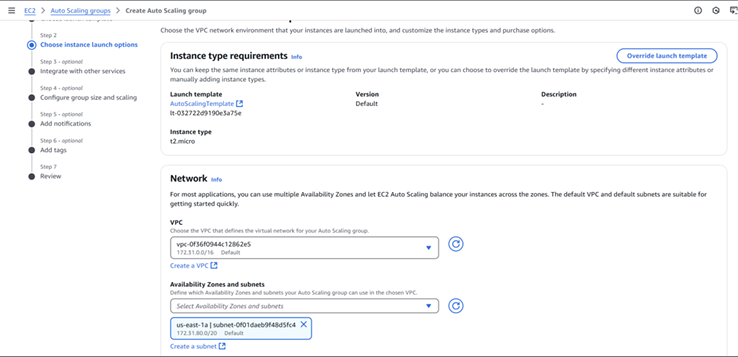
Step 6:

Auto Scaling group name: Give it a name (e.g., MyAutoScalingGroup). Launch Template: Select the launch template you created earlier (AutoScalingTemplate).



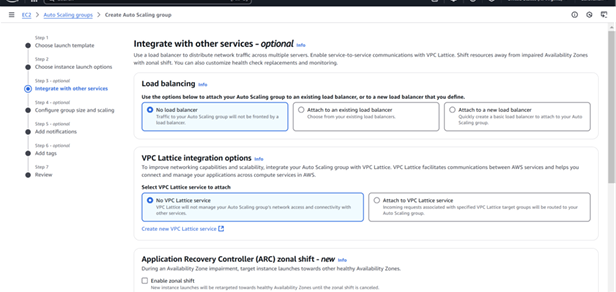
Step 7:

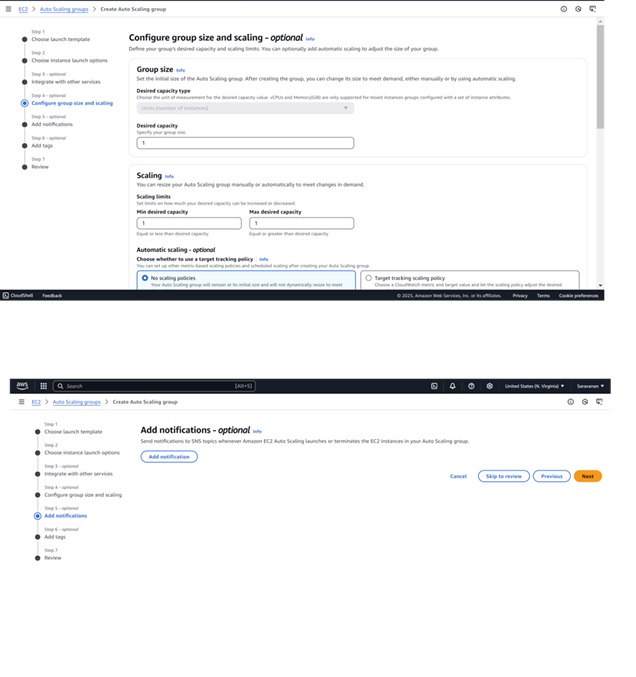
VPC and Subnets: Choose your VPC (it’s fine to use the default one). Select at least two subnets in different Availability Zones (this ensures high availability).



Step 8:

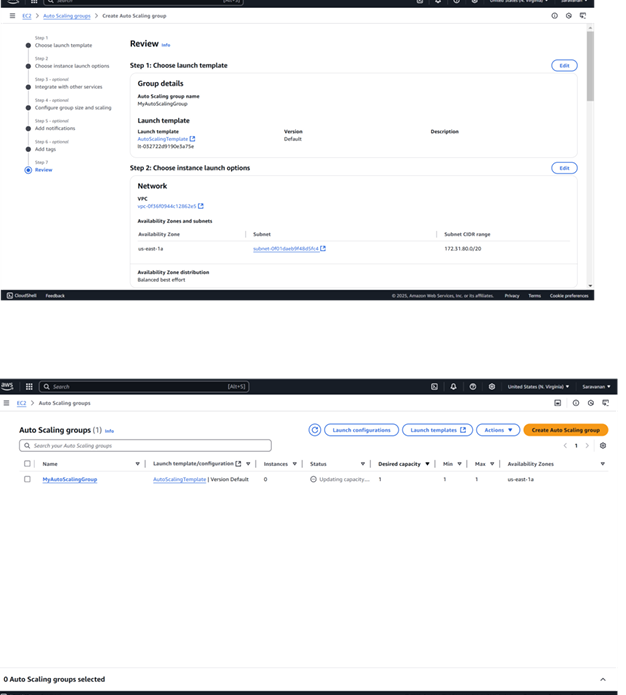
For this PoC leave the next settings as default and click next .

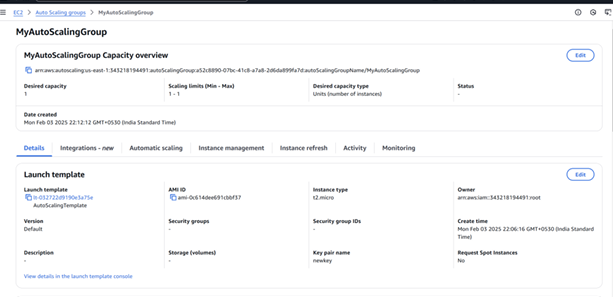




Step 9:

Review all the settings you've configured. Once satisfied, click Create Auto Scaling Group.





**Output**

This Proof of Concept (PoC) aimed to implement Auto Scaling in AWS to dynamically manage EC2 instances based on workload demand, ensuring efficient resource utilization and cost-effectiveness. Here's the outcome of the PoC:

1. Launch Template and Auto Scaling Group Setup: Successfully created a launch template and configured an Auto Scaling Group with scaling policies to dynamically manage EC2 instances based on workload.

2. Dynamic Scaling and Monitoring: Implemented scaling policies triggered by CPU utilization and verified automatic scaling actions using the Auto Scaling Group's Activity History.

3. Cost Awareness: Highlighted potential costs of running additional instances beyond the AWS Free Tier during testing and ensured resource usage was optimized.