

Hands-on with AWS Auto Scaling

What is Scaling?

Scaling is the process of **increasing or decreasing the number of servers** based on requirements.

Types of Scaling

1. **Vertical Scaling** – Add or reduce resources like CPU/RAM in a **single machine**
 - *Scale Up* – Increase resources
 - *Scale Down* – Decrease resources
2. **Horizontal Scaling** – Add or remove **multiple machines**
 - *Scale Out* – Add more machines
 - *Scale In* – Remove machines

Auto Scaling means adjusting the number of servers **automatically** based on system load.

Auto Scaling Practical Steps (AWS)

Step 1: Initial Setup

- Create a **Security Group** with port **22 (SSH)** and **80 (HTTP)**
- Create an **empty Target Group (TG)**
- Launch a **Load Balancer (LB)** and attach the TG

The screenshot shows the AWS Security Groups console. A success message at the top states: "Security group (sg-0ac40be40db7ad191 | SG-targets) was created successfully". Below this, the "sg-0ac40be40db7ad191 - SG-targets" page is displayed. The "Details" section shows the security group name is "SG-targets", owner is "396608790002", and it has 2 permission entries. The "Inbound rules" tab is selected, showing two rules: one for port 80 (HTTP) and another for port 22 (SSH). The "Outbound rules" tab is also present.

Screenshot 1: Security Group

The screenshot shows the AWS EC2 Target groups page. On the left, there's a navigation sidebar with options like AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main content area is titled "Target groups (1/1)" and shows a single target group named "TG1". The details for "TG1" are: ARN: arn:aws:elasticloadbalancing:...; Port: 80; Protocol: HTTP; Target type: Instance; Load balancer: None associated; VPC ID: vpc-0d0f0e0. There are "Actions" and "Create target group" buttons at the top right.

Screenshot 2: Empty Target Group

The screenshot shows the AWS EC2 Load balancers page. The left sidebar includes options for AMI Catalog, Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main section is titled "Load balancers (1/1)" and displays one load balancer named "LB1". The details for "LB1" are: DNS name: LB1-39541708.ap-south-1.elb.amazonaws.com; State: Provisioning.; VPC ID: vpc-0d08096a0a77e4f38; Availability Zones: 3 Availability Zones; Type: application. There are "Actions" and "Create load balancer" buttons at the top right.

Screenshot 3: Load Balancer creation

Step 2: Launch Template

- Define OS, CPU, RAM, storage, key pair, SG, and **bootstrap script**
- This template helps to **automatically create EC2 instances** during high load

User data - optional | Info

Upload a file with your user data or enter it in the field.

```
#!/bin/bash
yum install -y httpd
systemctl start httpd
chkconfig httpd on
echo "HELLO ALL" > /var/www/html/index.html
```

The screenshot shows the AWS EC2 Launch Templates page. The left sidebar has sections for Dashboard, EC2 Global View, Events, Instances (Instances, Instance Types), and Launch Templates. The main part is titled "Launch Templates (1/1)" and shows one launch template named "LT1". The details for "LT1" are: Launch Template ID: lt-0704a60cd58d9474b; Launch Template Name: LT1; Default Version: 1; Latest Version: 1; Create Time: 2025-06-04T10:22:48.000Z; Created By: arn:aws:iam:3966. There are "Actions" and "Create launch template" buttons at the top right.

Screenshot 4: Launch Template with AMI, CPU, RAM, Key, SG, and script

Step 3: Auto Scaling Group (ASG)

- Configure **minimum**, **maximum**, and **desired capacity**
 - ◆ *Minimum* → Instances during low load
 - ◆ *Maximum* → Instances during high load
 - ◆ *Desired* → Normal state between min & max

The screenshot shows the AWS EC2 Auto Scaling groups page. It displays a table with one row for an Auto Scaling group named 'ASG'. The table columns include Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. The 'Desired capacity' is set to 2, 'Min' to 2, and 'Max' to 5. The 'Availability Zones' listed are 'ap-south-1c, ap-south...'.

Screenshot 5: Auto Scaling Group setup with min, max, and desired values

Step 4: SNS Setup (Simple Notification Service)

- SNS is a region-specific service used to send notifications
- Create an **SNS topic**, add a **subscription** (email of the recipient)

The screenshot shows the AWS Amazon SNS Topics page. It displays a table with one row for a subscription. The subscription details shown are:

- Subscription:** 0dc98a32-a2de-4072-b704-2e61a81f315c
- ARN:** arn:aws:sns:ap-south-1:396608790002:Topic1:0dc98a32-a2de-4072-b704-2e61a81f315c
- Endpoint:** jayshrilandge3011@gmail.com
- Protocol:** EMAIL

Screenshot 6: SNS Topic and email subscription

Step 5: CloudWatch Alarms

- Create 2 alarms:

- CPU > 80% → Scale Out
- CPU < 20% → Scale In

The screenshots illustrate the creation and status of two CloudWatch alarms over four stages:

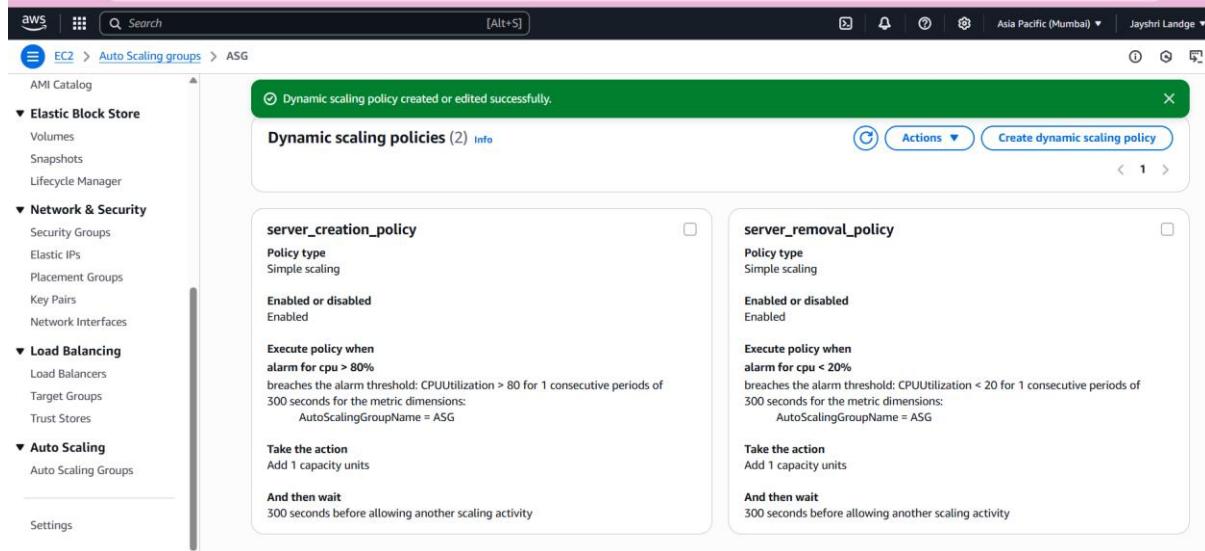
- Stage 1:** The first screenshot shows the creation of the 'alarm for cpu < 20%' alarm. A green success message at the top indicates it was successfully created. The table lists the alarm with a green 'OK' state.
- Stage 2:** The second screenshot shows both alarms created. The 'alarm for cpu < 20%' is now in an 'In alarm' state (red triangle). The 'alarm for cpu > 80%' remains 'OK'.
- Stage 3:** The third screenshot shows both alarms in an 'OK' state again. The 'Actions enabled' column indicates actions have been configured for both.
- Stage 4:** The fourth screenshot shows the 'alarm for cpu < 20%' back in an 'In alarm' state, while the 'alarm for cpu > 80%' remains 'OK'.

Alarm Name	State	Last state update (UTC)	Conditions	Actions
alarm for cpu < 20%	In alarm	2025-06-04 11:45:44	CPUUtilization < 20 for 1 datapoints within 5 minutes	Actions enabled
alarm for cpu > 80%	OK	2025-06-04 11:27:10	CPUUtilization > 80 for 1 datapoints within 5 minutes	Actions enabled

Screenshot 7: CloudWatch alarms - CPU > 80% and CPU < 20%

Step 6: Scaling Policies

- **Policy 1:** Create instances (attach CPU > 80% alarm)
- **Policy 2:** Remove instances (attach CPU < 20% alarm)



Screenshot 8: Scaling policies attached to alarms

Components of Auto Scaling (AWS)

1. EC2 Target
2. Target Group
3. Load Balancer (ALB)
4. Launch Template
5. Auto Scaling Group (ASG)
6. SNS Topic
7. CloudWatch Alarms
8. Scaling Policies

Conclusion

This hands-on exercise helped me understand how AWS Auto Scaling automatically handles instance provisioning and removal based on real-time metrics. It improves **availability, cost-efficiency, and performance** of applications hosted on EC2.