Auto Scaling

What is Auto Scaling?

Auto Scaling in AWS is a feature that automatically adjusts the number of Amazon EC2 instances based on demand. It ensures your application maintains performance while optimizing costs.

The purpose of AWS Auto Scaling is to automatically adjust the capacity of AWS resources to meet application demands. This helps ensure that applications perform at the desired level and maintain a high quality of service.

Benefits of AWS Auto Scaling

- **Cost optimization**

  Reduces cloud computing costs by adding capacity when demand is high and removing capacity when demand is low
- **Scalability**

  Increases scalability by automatically scaling resources in response to demand
- **Performance**

  Provides consistent performance by automatically adjusting capacity to meet demand
- **User experience**
  Improves user experience by automatically adjusting capacity to meet demand

How AWS Auto Scaling works:

AWS Auto Scaling monitors applications and automatically adjusts capacity based on defined situations, such as traffic or utilization levels. You can also manually adjust the size of your Auto Scaling group as needed.

☑ Key Components of Auto Scaling

1. **Launch Template or Launch Configuration**:
   - Defines instance details like AMI, instance type, key pair, security groups, and more.
2. **Auto Scaling Group (ASG)**:
   - A group of EC2 instances managed by Auto Scaling.
3. **Scaling Policies**:
   - Rules that determine when and how instances should scale (scale in or scale out).
4. **CloudWatch Alarms**:
   - Monitors metrics (CPU utilization, memory, network traffic) and triggers scaling events.

First create the VPC:

☑ Step 1: Login to AWS Management Console

1. Go to AWS Management Console
2. Navigate to **Services → VPC**
3. Select **Your VPCs → Create VPC**

☑ Step 2: Configure the VPC

- **Name**: Provide a name for your VPC (e.g., `MyCustomVPC`)
- **IPv4 CIDR Block**: Enter a valid CIDR block (e.g., `10.0.0.0/16`)
  - This gives you **65,536 IP addresses** (usable for subnets).
- **IPv6 CIDR Block (Optional)**: Choose if you need IPv6 support.
- **Tenancy**:
  - **Default**: Instances share hardware with other AWS accounts.
  - **Dedicated**: Instances run on dedicated hardware (more expensive).

Click **Create VPC**.

☑ Step 3: Create a Subnet

1. Go to **Subnets → Create Subnet**.
2. Select the VPC you created.
3. Provide:
   - **Subnet Name**: Example - `MyCustomSubnet`
   - **Availability Zone**: Select an AZ (e.g., `us-east-1a`).
   - **IPv4 CIDR Block**: Example - `10.0.1.0/24` (This gives 256 IP addresses).
4. **Create Subnet**.

☑ Step 4: Create an Internet Gateway

1. Go to **Internet Gateways → Create Internet Gateway**.
2. Provide a **Name**: Example - `MyCustomIGW`.
3. Click **Create Internet Gateway**.
4. Select the IGW → **Actions → Attach to VPC** → Select your VPC.

Create a Route Table

1. Go to **Route Tables** → **Create Route Table**.
2. Provide:
    - **Name**: Example - `MyCustomRouteTable`
    - **VPC**: Select your VPC.
3. Click **Create Route Table**.
4. Select the route table → **Routes** → **Edit Routes** → **Add Route**.
    - **Destination**: `0.0.0.0/0` (This means all traffic)
    - **Target**: Select the Internet Gateway.
5. Save the route.

Associate Route Table with Subnet

1. Go to **Route Tables** → Select your route table.
2. Go to **Subnet Associations** → **Edit Subnet Associations**.
3. Select your custom subnet and **Associate**.

Create the Target Group:

- Navigate to **EC2** → **Target Groups** under the **Load Balancing** section.
- Click on **Create Target Group**.

Select the instance first option→give the target name→change the VPC as custom VPC→then all are take default→click Next→just create the blank target group→click on the crate target group.

Create the Load Balancer:

Open load balancer→give load balancer name→select VPC→select the both subnets→here security group by using Http(80) and default one also→click create security group→select the security group which we crate→select the custom target group→click crate load balancer.

Create the auto scaling Group:

Click the auto scaling group→give the name for auto scaling group→crate the lunch template→give a name for lunch template→select the ubuntu→t2.micro→keypair→no need to select the subnet here→create the security group→name→select our custom VPC→allow the ssh and Http→and crate the security group→select the security which we created→enable the public IP→add in advanced setting install the Apache→click the lunch template→select the lunch temple which we have crate→ version is Default one→select the VPC→availability zone both 1a and 1b→click next→click the attach to load balancer→ select the target group our's→enable elastic health checks→health check grace period just give low like(20 to

50)→group size desired cap-2→mini cap-1→max cap-3→scaling just keep on none→click next→next→ next→ click crate auto scaling group.

Check the instance has been started or not.

Check with the DNS in browser it change the IP address of ur private IP addresses.

Then try to terminate the instance it recreate.