AWS Auto Scaling

AWS Auto Scaling helps you automatically adjust the number of Amazon EC2 instances or other resources to meet changing demand. It ensures your application has the right resources at the right time by scaling up (increasing resources) when demand is high and scaling down (reducing resources) when demand is low.

Advantages of AWS Auto Scaling:

1. **Improved Fault Tolerance**: Auto Scaling detects unhealthy instances and replaces them, ensuring your application is always running smoothly.
2. **Cost Optimization**: It reduces costs by dynamically adjusting the number of resources to the actual demand, preventing over-provisioning.
3. **Better Performance**: Ensures that you have enough capacity to handle incoming traffic without degrading performance.
4. **Increased Availability**: Auto Scaling ensures that the desired number of instances is always running, contributing to high availability.

Real-Time Use Cases:

1. **E-Commerce Websites**: During peak shopping seasons (e.g., Black Friday), traffic may spike, and Auto Scaling can increase EC2 instances to handle the load. After the traffic decrease, it scales down to save costs.
2. **Media Streaming Services**: Auto Scaling can adjust the infrastructure during major events or live streams when user activity is unpredictable.
3. **Gaming Applications**: During launch or new game content releases, game servers might need scaling to meet increased user demand.

Lab: AWS Auto Scaling

Step 1: Create a Classic Load Balancer

1. **Navigate to the EC2 Dashboard**:
   - In the AWS Management Console, go to **EC2**.
2. **Create a Classic Load Balancer**:
   - On the left sidebar, click **Load Balancers**, then select **Create Load Balancer**.
   - Choose **Classic Load Balancer**.
3. **Configure Load Balancer**:
   - Name the Load Balancer (e.g., MyLB).
   - Select all Availability Zones
4. **Configure Security Group**:
   - Create SG with SSH & HTTP opened
5. **Register Targets**:
   - Leave this section blank for now since the Auto Scaling group will add instances.
6. **Review and Create**:
   - Review the details and click **Create**.

Step 2: Create a Launch Template

1. **Navigate to Launch Templates**:
   - In the EC2 Dashboard, click **Launch Templates** on the left sidebar.
   - Select **Create Launch Template**.
2. **Configure Launch Template**:

    - **Name** your launch template (e.g., MyLTMP).
    - **AMI**: Select an Amazon Machine Image (AMI). You can use Amazon Linux for a simple web server setup.
    - **Instance Type**: Choose the instance type, such as **t2.micro** (free-tier eligible).
    - **Key Pair**: Choose an existing key pair or create a new one.

3. **Network Settings**:
    - Select existing **Security Group** that allows HTTP (port 80) access for the web server.
4. **Storage**:
    - Leave the default root volume size, or modify it based on your requirements.
5. **User Data** (optional):
    - Add a simple user data script to install a web server:

```
#!/bin/bash

sudo su -

yum install httpd -y

cd /var/www/html

echo “MyGoogle” > index.html

service httpd start
```

6. **Review and Create**:
    - Click **Create Launch Template**.

Step 3: Create an Auto Scaling Group

1. **Navigate to Auto Scaling Groups**:
    - In the EC2 Dashboard, go to **Auto Scaling Groups** and click **Create Auto Scaling Group**.
2. **Select the Launch Template**:
    - Choose the **Launch Template** you created in Step 2.
3. **Configure the Auto Scaling Group**:
    - **Name** the Auto Scaling group (e.g., MyASG).
    - Choose all Availability Zones
4. **Attach to Classic Load Balancer**:
    - Under the **Load Balancing** section, select **Attach to a Classic Load Balancer**.
    - Choose the Load Balancer you created in Step 1 (MyLB).
5. **Scaling Policies**:
    - Set the **Desired Capacity** (e.g., 4 instances), **Minimum Capacity** (3 instance), and **Maximum Capacity** (e.g., 4 instances).
    - Add **Scaling Policies** to scale based on load. For example:
        - Increase instances by 1 when CPU utilization is above 80%.
6. **Health Checks**:
    - Enable **ELB Health Checks**
7. **Tags**:
    - Give any name to instances (Eg: WebServer)
8. **Review and Create**:
    - Review the configuration and click **Create Auto Scaling Group**.

Verification:

1. **Monitor the Auto Scaling Group**:
    - Go to **Auto Scaling Groups** in the EC2 Dashboard.
    - Monitor the number of running instances. They should automatically register with the Classic Load Balancer.
2. **Test the Setup**:
    - Use the DNS name of your Classic Load Balancer to access your application in the browser.
    - You should see the web page served by the EC2 instances created by the Auto Scaling group.

Clean Up:

- Delete the Auto Scaling group, Launch Template and Classic Load Balancer to avoid unnecessary charges.