AutoScaling-LoadBalancer-website

**1) Configure EC2 Security Groups**

* **Application Load Balancer Security Group:**
  + Create a security group specifically for the Application Load Balancer.
  + Set inbound rules to allow HTTP traffic with **source: Anywhere (IPv4)**.
  + Save by clicking on **Create Security Group**.
* **AutoScaling Security Group:**
  + Create a separate security group for AutoScaling.
  + Set inbound rules to allow **All TCP** with source set to the Application Load Balancer’s security group.
  + Add an additional inbound rule to allow SSH with **source: Anywhere (IPv4)**.

**2) Create an EC2 AutoScaling Group**

* **Set Up AutoScaling Group**:
  + Navigate to **AutoScaling Group** and provide a name, such as DemoAutoScaling.
  + For the **launch template**, create a template named TestAutoScaling:
    - **Application and OS Images (AMI)**: Choose **Ubuntu** from Quickstart.
    - **Instance Type**: Select **t2.micro** (free tier).
    - **Key Pair**: Select your SSH key pair.
    - **Network Settings**: Attach the AutoScaling security group.

**User Data**: Enter the following script to install and start Apache:  
bash  
Copy code  
#!/bin/bash

apt-get update -y

apt-get install -y apache2

systemctl start apache2

systemctl enable apache2

echo "<html><body><h1>Apache is running on EC2 instance: $(hostname -i)</h1></body></html>" > /var/www/html/index.html

* Click on **Launch**.
* **Configure Subnets and Load Balancer**:
  + In the AutoScaling Group settings, select subnets to include (e.g., Subnets 3 and 4).
  + Under **Load Balancing**, select **Attach a New Load Balancer** and name it, e.g., DemoAppLoadBalancer.
  + Set the load balancer to **Internet-facing**.
  + Create a target group, configure health checks, and proceed with the default settings.
* **Set Group Size and Scaling**:
  + Configure desired capacity: **Min = 1**, **Max = 2**.
  + Review and complete the setup.

**3) Update Load Balancer Security Group**

* Go to your Load Balancer settings, select **Security**, and change to the **Application Load Balancer** security group if necessary.

**4) Test Load Balancer**

* Copy the DNS name of the Load Balancer and paste it in your browser.
* You should see the message: *"Apache is running on an EC2 instance."*

**5) Configure Automatic Scaling Policy**

* In the AutoScaling group, select **Automatic Scaling** and create a **Dynamic Scaling Policy**.
* Set the **Target Tracking Policy** to monitor **Average CPU Utilization** with a **target value of 30**.

**6) Verify CloudWatch Metrics**

* CloudWatch will automatically monitor the scaling conditions. Ensure CPU metrics are displayed.

**7) Connect to EC2 Instance and Simulate Load**

Connect to an instance in the AutoScaling group and install the stress library:  
bash  
Copy code  
sudo apt install stress -y

Run a stress test to increase CPU load:  
bash  
Copy code  
sudo stress --cpu 12 --timeout 240s

* **Monitor the Stress Process**:

Check for active stress processes:  
bash  
Copy code  
ps aux | grep stress

Use the kill command to stop a specific process:  
bash  
Copy code  
kill -9 <process\_id>

**8) Verify Scaling**

* Monitor CloudWatch to ensure the condition (CPU > 30%) triggers scaling.
* Confirm new instances are created automatically in response to CPU load, and check if they appear in the AutoScaling group.
* Verify that the public IP changes for the instances behind the Load Balancer when scaling occurs.