Assn#2: Horizontal Scaling Scenario#1

For the topic of "AWS: Horizontal Scaling," we'll focus on a classic and highly relevant service: EC2 Auto Scaling combined with Application Load Balancer (ALB). This setup beautifully demonstrates how to automatically scale out your application in response to demand.

Configure group size:

* Desired capacity: 1

* Minimum capacity: 2

* Maximum capacity: 4 (This allows for scaling up)

AWS: HorizontalScaling- Hands-onSteps

Phase 1: Launching a Basic EC2 Instance (Our Baseline)

1. Navigate to EC2: Open the AWS Management Console and go to the EC2 ser vice.
2. Launch Instance: Click on "Launch instances."
3. Choose an AMI: Under "Amazon Machine Image (AMI)," select a Free Tier eligible Amazon Linux 2 AMI (or any other Free Tier Linux AMI).

4. Choose Instance Type: Select the "t2.micro" instance type (Free Tier eligible).

5. Configure Instance Details:

    - For "Number of instances," leave it as 1.
    - For "Network," select your default VPC.
    - For "Subnet," select any of your public subnets.
    - For "Auto-assign Public IP," ensure it is set to "Enable."
    - Leave other settings as default.

6. Add Storage: Accept the default 8 GiB General Purpose SSD (gp2) root volume (Free Tier eligible).

7. Add Tags (Red9SysTech Logging): Click "Add Tag."

    - Key: `Name`
    - Value: `cst-R9ST-Baseline-WebServer`
    - Click "Add Tag" again.
    - Key: `Environment`
    - Value: `Red9SysTech-Demo`

8. Configure Security Group:

    - Select "Create a new security group."
    - Security group name: `cst-R9ST-Baseline-SG`
    - Description: `Allow HTTP access for baseline web server`
    - Add a rule:
        - Type: `HTTP`
        - Protocol: `TCP`
        - Port range: `80`
        - Source: `Anywhere` (For demo purposes; in a real-world scenario, restrict this)
    - Click "Add Rule."

    - Type: SSH
    - Protocol: TCP
    - Port range: 22
    - Source: My IP (or your organization's allowed IP range for secure access)

9. Review and Launch: Review your instance configuration and click "Launch."
10. Select or Create Key Pair: Choose "Create a new key pair."
    - Key pair name: cst-R9ST-Baseline-KeyPair
    - Click "Download Key Pair" and store the .pem file in a secure location. You'll need this to SSH into the instance if needed.
    - Check the box acknowledging you have access to the private key file.
    - Click "Launch Instances."

Phase 2: Installing a Simple Web Server on the Baseline Instance

1. Connect to the Instance via SSH:
    - Open your terminal or SSH client.
    - Use the following command (replace placeholders with your actual values):

```
ssh -i "path/to/your/cst-R9ST-Baseline-KeyPair.pem" ec2-user@<Public IPv4 address of your instance>
```

2. Install HTTPD (Apache Web Server):

```
#!/bin/bash
sudo su
sudo yum update -y
sudo install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
```

3. Create a Simple Web Page:

   ```
   echo '<h1>Hello from Red9SysTech Baseline Server!</h1>' | sudo tee /var/www/html/index.html
   ```

4. Verify Web Server: Open a web browser and navigate to the Public IPv4 address of your EC2 instance. You should see "Hello from Red9SysTech Baseline Server!"

Phase 3: Creating an Application Load Balancer (ALB)

1. Navigate to Load Balancers: In the EC2 service console, under "Load Balancing," click on "Load Balancers."
2. Create Load Balancer: Click "Create Load Balancer."
3. Select Application Load Balancer: Choose "Application Load Balancer" and click "Create."
4. Configure Load Balancer:
   - Basic Configuration:
     - Load balancer name: `cst-R9ST-WebApp-ALB`
     - Scheme: `Internet-facing`
     - IP address type: `IPv4`
   - Network Mapping:
     - Select your VPC.
     - Select at least two public subnets in different Availability Zones.
5. Configure Listener:
   - Listener: `HTTP`, Port `80`
   - Default action: Create a new target group.
6. Configure Target Group:
   - New target group:
     - Target group name: `cst-R9ST-WebApp-TG`

        - Protocol: HTTP
        - Port: 80
        - Health checks:
            - Protocol: HTTP
            - Path: /
            - Keep other settings as default.
    - Click "Next."
    - Register Targets: Select your running cst-R9ST-Baseline-WebServer instance and click "Include as pending below." Then click "Create target group."

7. Review and Create: Review your ALB configuration and click "Create."

Phase 4: Creating an Auto Scaling Group (ASG)

1. Navigate to Auto Scaling Groups: In the EC2 service console, under "Auto Scaling," click on "Auto Scaling Groups."
2. Create Auto Scaling Group: Click "Create Auto Scaling group." - cst-R9ST-Baseline-ASG
3. Choose Launch Template or Launch Configuration: Click "Create a launch template."
4. Create Launch Template:
    - Launch template name: cst-R9ST-WebApp-LT
    - Provide a description (e.g., Launch template for web servers in the ASG - Red9SysTech).
    - Under "Amazon Machine Image (AMI)," select the same Free Tier eligible Amazon Linux 2 AMI you used before.
    - Under "Instance type," select "t2.micro."
    - Under "Key pair (login)," select the cst-R9ST-Baseline-KeyPair you created.
    - Under "Network settings," select the security group cst-R9ST-Baseline-SG you created.
    - Under "Advanced network configuration," ensure "Auto-assign public IP" is set to "Do not assign a public IP" (as the ALB will handle public access).

- Under "User data (optional)," you can paste the same script you used to install the web server on the baseline instance:

```
#!/bin/bash
sudo su
sudo yum update -y
sudo install httpd -y
sudo systemctl start httpd
sudo systemctl enable httpd
echo '<h1>Hello from Red9SysTech Auto Scaled Server!</h1>' | sudo tee /var/www/html/index.html
```

- Click "Create launch template."

5. Create Auto Scaling Group (using the Launch Template):

- Go back to the "Auto Scaling Groups" page and click "Create Auto Scaling group."
- Auto Scaling group name: `cst-R9ST-WebApp-ASG`
- Under "Launch Template," select the `cst-R9ST-WebApp-LT` you just created. Choose the latest version. Click "Next."
- Configure settings:
    - Network: Select your VPC.
    - Availability Zones and subnets: Select at least two public subnets in different Availability Zones. (Avoid ap-south-1c zone)
- Configure Load Balancing:
    - Select "Attach to an existing load balancer."
    - Choose the `cst-R9ST-WebApp-ALB` you created.
    - Under "Target Groups," select the `cst-R9ST-WebApp-TG`.
- Configure health checks: Ensure "ELB" is selected.
- Configure group size:
    - Desired capacity: `1`

    - Minimum capacity: 2
    - Maximum capacity: 4 (This allows for scaling up)
- Configure scaling policies:
    - Select "Target tracking scaling policy."
    - Policy name: cst-R9ST-CPU-Tracking
    - Metric type: EC2 instance CPU utilization
    - Target value: 50 (Scale out when average CPU utilization across the group exceeds 50%)
    - Keep other settings as default.
- Configure notifications (Optional): You can set up notifications for scaling events if you wish.
- Configure tags (Red9SysTech Logging): Click "Add Tag."
    - Key: Name
    - Value: cst-R9ST-WebApp-Instance
    - Click "Add Tag" again.
    - Key: Environment
    - Value: Red9SysTech-Demo
- Click "Create Auto Scaling group."

Phase 5: Testing Horizontal Scaling

1. Access the Application: Open a web browser and navigate to the DNS name of your Application Load Balancer. You should see "Hello from Red9SysTech Auto Scaled Server!" (or the baseline server initially until the ASG launches new instances).

2. Simulate Load (Optional but Recommended): To trigger the scaling policy, you can simulate load on the instances. Connect to one of the instances via SSH and run a CPU-intensive command (e.g., `yes > /dev/null &`). Monitor the CPU utilization in the AWS Management Console (EC2 → Instances → Monitoring tab).

3. Observe Scaling: After a few minutes, if the average CPU utilization of your Auto Scaling group exceeds 50%, you should see the Auto Scaling group launch a new EC2 instance. You can monitor this in the "Auto Scaling Groups" tab under "Activity history."

4. Verify New Instance: Once the new instance is healthy and registered with the ALB, refresh your web browser. You might see the "Hello from Red9SysTech Auto Scaled Server!" message again, potentially served by a different instance.

Phase 6: Cleanup (Important for Free Tier)

1. Delete the Auto Scaling Group:
    - Navigate to "Auto Scaling Groups" in the EC2 console.
    - Select the `cst-R9ST-WebApp-ASG`.
    - Click "Actions" and then "Delete."
    - Confirm the deletion. Choose the option to "Delete associated EC2 instances" if you want the ASG to terminate the instances it launched.
2. Delete the Launch Template:
    - Navigate to "Launch Templates" in the EC2 console.
    - Select the `cst-R9ST-WebApp-LT`.
    - Click "Actions" and then "Delete template."
    - Confirm the deletion.
3. Delete the Application Load Balancer:
    - Navigate to "Load Balancers" in the EC2 console.
    - Select the `cst-R9ST-WebApp-ALB`.
    - Click "Actions" and then "Delete."
    - Confirm the deletion.
4. Delete the Target Group:
    - Navigate to "Target Groups" in the EC2 console.
    - Select the `cst-R9ST-WebApp-TG`.

- Click "Actions" and then "Delete."
- Confirm the deletion.

5. Terminate the Baseline EC2 Instance:
    - Navigate to "Instances" in the EC2 console.
    - Select the `cst-R9ST-Baseline-WebServer` instance.
    - Click "Instance state" and then "Terminate instance."
    - Confirm the termination.
6. Delete the Security Groups:
    - Navigate to "Security Groups" in the EC2 console.
    - Select the `cst-R9ST-Baseline-SG`.
    - Click "Actions" and then "Delete security group."
    - Confirm the deletion.
7. Delete the Key Pair (Optional): If you no longer need the key pair, you can delete it from the "Key Pairs" section in the EC2 console.
8. Verify No Running Resources: Ensure that all the resources you created (EC2 instances, Load Balancer, Auto Scaling Group, etc.) have been terminated or deleted to avoid incurring any charges.
    ============================
    Final Notes to remember:
    Desired capacity: 1
    Minimum capacity: 2
    Maximum capacity: 4

Initial State: When the ASG is created, it will launch 2 instances immediately. The ASG always ensures that the number of running instances is at least the minimum capacity, even if the desired capacity is lower.

Scaling Up: If a scaling policy is triggered, the ASG will launch additional instances, up to a maximum of 4.

Scaling Down: If a scale-down policy triggers, the ASG will terminate instances, but it will never go below the minimum capacity of 2. The desired capacity will remain 1 unless a scaling action changes it.

Instance Failure: If one of the 2 running instances fails, the ASG will launch a replacement to maintain at least the minimum of 2.

Testing:

Access the ALB – you should see traffic distributed across 2 instances.

Simulate load to trigger scaling up; observe the launch of 1 or 2 more instances.

Let the load subside; observe scaling down, but it will stop at 2 instances (the minimum).

Manually terminate one of the ASG-managed instances; observe the ASG launching a replacement to maintain 2.