Creating an Auto Scaling Group and Application Load Balancer

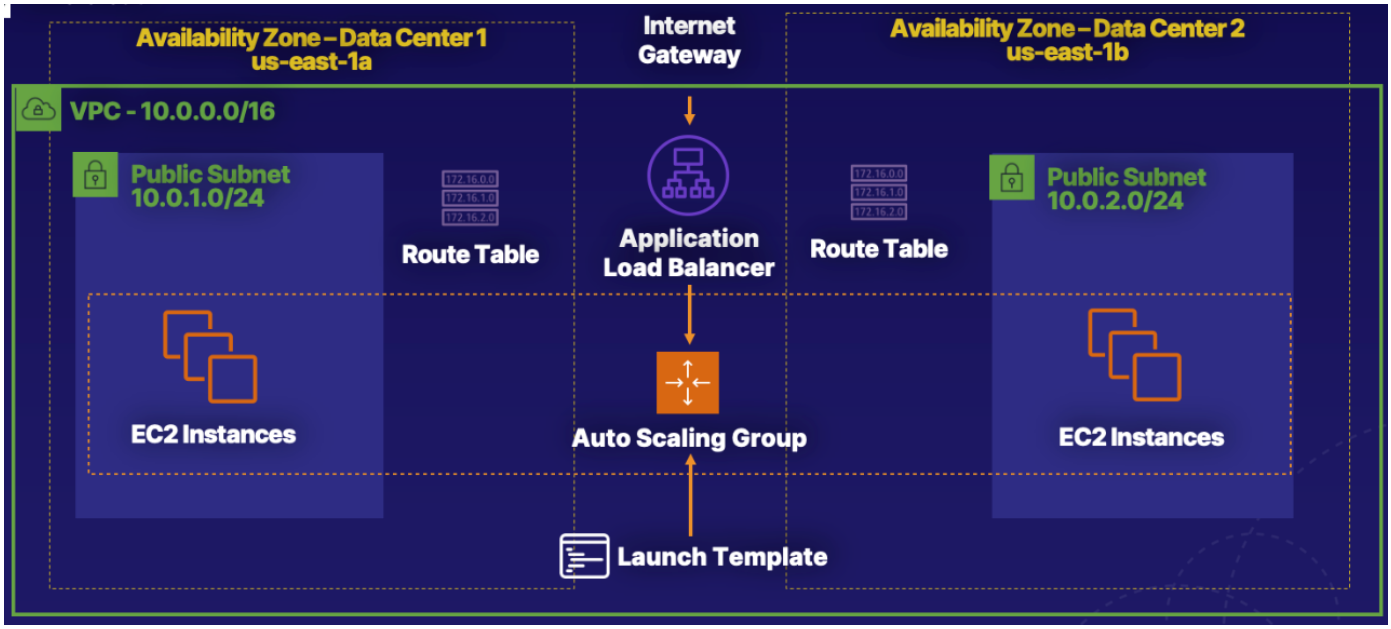
Introduction

In this hands-on lab scenario, you're a cloud network engineer working for an organization that sells products online. You're gearing up for the annual sale that provides a 50% discount on all items. This sale drives a ton of traffic and revenue. Your job is to ensure the website doesn't go down and is able to handle every request efficiently.

In this lab, you will integrate two powerful AWS services: Elastic Load Balancers and Auto Scaling groups.

Specifically, you will create an Auto Scaling group of EC2 instances operating as web servers and then configure an Application Load Balancer to load balance between the instances inside that Auto Scaling group.

After everything is set up, you'll simulate stress tests on the EC2 instances to confirm the Auto Scaling group works as expected.



1- Create an Application Load Balancer

1.Navigate to EC2 > Load Balancers.

2. Click Create Load Balancer.

3. In the Application Load Balancer card, click Create.

4. In the Basic Configuration section, set the following values:

    - Name: HOLALB

    - Scheme: internet-facing

    - IP address type: ipv4

5. Leave the settings in the Listeners section as-is.

6. In the Availability Zones section, select the listed VPC.

7. Select us-east-1a and us-east-1b.

8. Click Next: Configure Security Settings.

9. Click Next: Configure Security Groups.

10. For Assign a security group, select Create a new security group.

11. Give it a Security group name and Description of "ALBSG".

12. Make sure the following inbound rule is added:

    - Type: HTTP

    - Source: Anywhere-IPv4 0.0.0.0/0

13. Select the ALBSG security group and remove the default security group .

14. Click Next: Configure Routing.

15. In the Target group section, set the following values:

- Target group: New target group

- Name: ALBTG

- Target type: Instance

- Protocol: HTTP

- Port: 80

16. Leave the Health checks section settings as-is.

17. In the Advanced health check settings section, change the Healthy threshold to "2".

18. Click Next: Register Targets.

19. Leave the settings as-is.

20. Click Next: Review.

21. Click Create

It will take about 5–10 minutes to be created.

22. On the load balancer dashboard, with the ALB selected, copy the DNS name listed in the Description section.

23. Paste it into a new browser tab, which will result in an error.

2- Create a Launch Template

2.1- Create SSH Key Pair

1. In the AWS console, navigate to EC2 > Key Pairs.

2. Click Create key pair.

3. Give it a name of "ALBKP".

4. Click Create key pair.

5. Once it's created, it will automatically download. Make sure you know where it's saved on your computer, as we'll need it later.

2.2- Create Security Group for EC2 Instances

1. In the left-hand menu, select Security Groups.
2. Click Create Security Group, and set the following values:
    - Security group name: EC2WEBSG
    - Description: EC2WEBSG
    - VPC: The listed lab VPC

3. Under Inbound rules, click Add rule, and set the following values:
    - Type: SSH
    - Source: Anywhere

4. Still under Inbound rules, click Add rule again, and set the following values:
    - Type: HTTP
    - Source: Custom, and select ALBSG from the search dropdown
5. Click Create security group.

2.3- Create EC2 Instance Launch Template

1. Select Launch Templates in the left-hand menu.

2. Click Create launch template.

3. Set the following values:

    - Launch template name: HOLLT

    - Template version description: HOLLT

4. Under Auto Scaling guidance, check the box that say Provide guidance to help me set up a template that I can use with EC2 Auto Scaling.

5. In the Launch template contents section, click into the dropdown.

6. Search for "Amazon Linux 2".

7. Select Amazon Linux 2 AMI 64-bit (x86).

8. Set the following values:

    - Instance type: t1.micro

    - Key pair name: ALBKP

    - Network type: VPC

    - Subnet: Don't include in launch template

    - Security groups: EC2WEBSG

9. Expand the Advanced details section, and paste the following into the User data box:

```
#!/bin/bash
yum update -y
yum install -y httpd
yum install -y wget
cd /var/www/html
wget
https://raw.githubusercontent.com/ACloudGuru-Resources/Course-Certified-S
olutions-Architect-Associate/master/labs/creating-an-auto-scaling-group-a
nd-app-load-balancer-aws/index.html
wget
https://raw.githubusercontent.com/ACloudGuru-Resources/Course-Certified-S
olutions-Architect-Associate/master/labs/creating-an-auto-scaling-group-a
nd-app-load-balancer-aws/acg.jpg
service httpd start
```

10. Click Create launch template
11. Click View launch templates.

3- Create an Auto Scaling Group

1. In the left-hand menu, click Load Balancers.

2. Ensure the HOLALB load balancer displays an active state.

3. Select Auto Scaling Groups in the left-hand menu.

4. Click Create an Auto Scaling group.

5. On the Choose launch template or configuration screen, set the following values:

    - Auto Scaling group name: HOLASG

    - Launch template: HOLLT

6. Click Next.

7. On the Configure settings screen, set the following values:

    - Purchase options and instance types: Adhere to launch template

    - VPC: Select the listed VPC

    - Subnets: us-east-1a and us-east-1b

8. Click Next.

9. On the Configure advanced options screen, set the following values:

    - Load balancing: Enable load balancing, Application Load Balancer or Network Load Balancer

    - Choose a target group for your load balancer: ALBTG

    - Leave the Health checks settings as-is.

    - Under Additional settings, check the box to Enable group metrics collection within CloudWatch.

10. Click Next.

11. On the Configure group size and scaling policies screen, set the following values:

    - Desired capacity: 2
    - Minimum capacity: 2
    - Maximum capacity: 6
    - Select Target tracking scaling policy.
    - Scaling policy name: Target Tracking Policy
    - Metric type: Average CPU utilization
    - Target value: 30
    - Instances need: 300

12. Click Next.

13. Leave the Add notifications screen as-is, and click Next.

14. Leave the Add tags screen as-is, and click Next.

15. Click Create Auto Scaling group.

16. Refresh the browser tab where we tried visiting the ALB's DNS name. Now, it should properly display a website.

17. In the AWS console, click Instances in the left-hand menu. We should see two instances there.

4- Test Horizontal Scaling

1. Still on the Instances page, right-click one of the instances, and click Connect.

2. Click SSH client.

3. Copy the chmod command listed.

4. Open a terminal session, and change to your downloads directory or whatever directory the key pair file is saved (e.g., cd Downloads).

5. Run the chmod command to change the permissions on the key pair.

6. Copy the ssh connection string listed in the connection dialog in the AWS console.

7. Run the connection string in the terminal session to connect to the instance.

8. Install the stress test application:

    sudo amazon-linux-extras install epel -y

    sudo yum install -y stress

9. Run the stress test on the EC2 instance:

    stress --cpu 2 --timeout 300

    It could take up to 10 minutes to increase the CPU utilization.

10. On the instances page in the AWS console, click the Monitoring tab to see the increased CPU utilization.

11. Refresh the instances table to see if any were added. (We should see several more are being added.)

12. Click Auto Scaling Groups in the left-hand menu.

13. Click our HOLASG Auto Scaling group.

14. Click the Activity tab, where we should see new instances being launched.

15. In the terminal, stop the stress test by pressing Ctrl+C.

16. In the AWS console, navigate back to Instances.

17. Click the Monitoring tab, where we should now see the CPU utilization is starting to go back down.

18. After a few minutes, refresh the page to see CPU utilization much lower.

19. Click Auto Scaling Groups in the left-hand menu.

20. Click our HOLASG Auto Scaling group.

21. Click the Activity tab, where we should see instances are being terminated.

22. Click Instances in the left-hand menu.

23. Refresh the instances table, where we should now see some instances are being terminated.