Lab 3: Scale and Load Balance Your Architecture

This lab builds on the previous lab and walks you through using the Elastic Load Balancing (ELB) and Auto Scaling services to load balance and automatically scale your infrastructure.

Elastic Load Balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. It enables you to achieve fault tolerance in your applications by seamlessly providing the required amount of load balancing capacity needed to route application traffic. Elastic Load Balancing offers two types of load balancers that both feature high availability, automatic scaling, and robust security. These are the Classic Load Balancer which routes traffic based on either application- or network-level information, and the Application Load Balancer which routes traffic based on advanced application-level information that includes the content of the request. The Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, and the Application Load Balancer is ideal for applications that need advanced routing capabilities, microservices, and container-based architectures. The Application Load Balancer offers you the ability to route traffic to multiple services or load balance across multiple ports on the same EC2 instance.

Auto Scaling helps you maintain application availability and allows you to scale yourAmazon EC2 capacity out or in automatically according to conditions you define. You can use Auto Scaling to help ensure that you are running your desired number of Amazon EC2 instances. Auto Scaling can also automatically increase the number of Amazon EC2 instances during demand spikes to maintain performance and decrease capacity during lulls to reduce costs. Auto Scaling is well suited to applications that have stable demand patterns or that experience hourly, daily, or weekly variability in usage.

Objectives

After completing this lab, you can:

- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

This lab takes approximately **45 minutes**.

Access the AWS Management Console

Task 1: Create an AMI for Auto Scaling

In this task, you create an AMI as the starting point for launching new instances to use with Auto Scaling.

1. [1] In the **AWS Management Console**, on the **Services** menu, click **EC2**.
2. In the navigation pane, click **Instances**.

3. Verify that the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. If it doesn't, wait until it does before proceeding to the next step. Use the refresh icon in the upper right corner to check for updates.
4. Right-click on **Web Server 1**, and then click **Image** > **Create Image**.
5. Configure the following settings (and ignore any settings that aren't listed):
    - **Image name**: type `Web Server AMI`
    - **Image description**: type `Lab 6 AMI for Web Server`
6. Click **Create Image**.
   The confirmation screen displays the **AMI ID** for your new AMI. Click **Close**.

Task 2: Create a Load Balancer

In this task, you create a load balancer to balance traffic across several EC2 instances in two Availability Zones.

1. [7] In the navigation pane, click **Load Balancers**.
2. Click **Create Load Balancer**.
3. Select **Application Load Balancer**, and click **Continue**.
4. Configure the following settings (and ignore any settings that aren't listed):
    - **Name**: type `Lab6ELB`
    - **VPC**: Click **My Lab VPC**.
    - **Availability Zones**: Select both to see the available subnets.
      Then, select **Public Subnet 1** and **Public Subnet 2**
5. Click **Next: Configure Security Settings**.
6. Ignore the following warning: "*Improve your load balancer's security. Your load balancer is not using any secure listener*" and click **Next: Configure Security Groups**.
7. Select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access** and clear the **default** check box (indicating the default Security Group).
8. Click **Next: Configure Routing**.
9. Under **Target group**, for **Name**, type `Lab6Group`.
10. Expand **Advanced health check settings**, and configure the following settings (and ignore any settings that aren't listed):
    - **Healthy threshold**: type `2`
    - **Unhealthy threshold**: type `3`
    - **Timeout**: type `10`
11. Click **Next: Register Targets**.
    Auto Scaling will automatically add instances later. Click **Next: Review**.
12. Review the configuration of your load balancer and click **Create**.
13. On the "Successfully created load balancer" message, click **Close**.

Task 3: Create a Launch Configuration and an Auto Scaling Group

In this task, you create a launch configuration for your Auto Scaling group. A launch configuration is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch configuration, you specify information for the instances such as the AMI, the instance type, a key pair, one or more security groups and a block device mapping. An Auto Scaling group contains a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of instance scaling and management.

1. [20] In the navigation pane, click **Launch Configurations**.
2. Click **Create Auto Scaling group**.
3. Click **Create launch configuration**.
4. In the navigation pane, click **My AMIs**.
5. In the row for **Web Server AMI**, click **Select**.
6. Accept the **t2.micro** selection and click **Next: Configure details**.
7. Configure the following settings (and ignore any settings that aren't listed):
    - **Name**: type `Lab6Config`
    - **Monitoring**: Click **Enable CloudWatch detailed monitoring**.
8. Click **Next: Add Storage**.
9. Click **Next: Configure Security Group**.
10. Click **Select an existing security group**, and select the security group that contains **WebSecurityGroup** in the **Name** and a **Description** of **Enable HTTP access**.
11. Click **Review**.
12. Review the details of your launch configuration and click **Create launch configuration**.
    Ignore the "Improve security..." warning; this is expected.
13. Click **Choose an existing key pair**, select a key pair, select the acknowledgement check box, and click **Create launch configuration**.
14. Configure the following settings (and ignore any settings that aren't listed):
    - **Group name**: type `Lab6 AS Group`
    - **Group size Start with**: type `2` (instances)
    - **Network**: Click **My Lab VPC**.
      Ignore the message regarding "no public IP"; this is expected.
    - **Subnet**: Click **Private Subnet 1 (10.0.3.0/24)**, and
      Click **Private Subnet 2 (10.0.4.0/24)**.
15. Expand **Advanced Details**, configure the following settings (and ignore any settings that aren't listed):
    - **Load Balancing**: Click **Receive traffic from one or more load balancers**
    - **Target Groups**: Click **Lab6Group**.
    - **Health Check Type**: Click **ELB**.
    - **Monitoring**: Click **Enable CloudWatch detailed monitoring**.
16. Click **Next: Configure scaling policies**.
17. Select **Use scaling policies to adjust the capacity of this group**
18. Modify the **Scale between** text boxes to scale between **2** and **6** instances.
19. Click **Scale the Auto Scaling group using step or simple scaling policies**
20. In **Increase Group Size**, for **Execute policy when**, click **Add new alarm**.
21. Clear **Send a notification to:** .
22. Configure the following settings (and ignore any settings that aren't listed):

- **Whenever**: Click **Average**, and then click **CPU Utilization**.
- **Is**: Click **>=**, and then type `65` (indicating percent).
- **For at least**: type `1`, and then click **1 Minute**.
- **Name of alarm**: Replace exiting entry with **High CPU Utilization**

23. Click **Create Alarm**.
24. In **Increase Group Size**, configure the following settings (and ignore any settings that aren't listed):
    - **Take the action**: type `1`, click **instances**, and then type `65`
    - **Instances need**: type `60`
      (seconds to warm up after each step)
25. In **Decrease Group Size**, for **Execute policy when**, click **Add new alarm**.
26. Clear **Send a notification to:** .
27. Configure the following settings (and ignore any settings that aren't listed):
    - **Whenever**: Click **Average**, and then click **CPU Utilization**.
    - **Is**: Click **<=**, and then type `20`
    - **For at least**: type `1`, and then click **1 Minute**.
    - **Name of alarm**: Replace exiting entry with **Low CPU Utilization**
28. Click **Create Alarm**.
29. In **Decrease Group Size**, for **Take the action**: click **Remove**, type `1`, click **instances**, and then type `20`
30. Click **Next: Configure Notifications**.
31. Click **Next: Configure Tags**.
32. Configure the following settings (and ignore any settings that aren't listed):
    - **Key**: type `Name`
    - **Value**: type `Lab 6 Web Instance`
33. Click **Review**.
34. Review the details of your Auto Scaling group, and then click **Create Auto Scaling group**.
35. Click **Close** when your Auto Scaling group has been created.

Task 4: Verify Auto Scaling is Working

In this task, you verify that Auto Scaling is working correctly.

1. [54] In the navigation pane, click **Instances**.
   Four instances are displayed: **Web Server 1**, **NAT Server**, and two new instances labeled as **Lab 6 Web Instance.**
   Note: The new instances should appear as running after a few minutes.
2. In the navigation pane, click **Target Groups**.
3. Select **Lab6Group**, and click the **Targets** tab.
   Two **Lab 6 Web Instance** instances should be listed for this target group.
4. Wait until the **Status** of both instances transitions to *healthy*. Use the refresh icon in the upper right corner to check for updates.
5. In the navigation pane, click **Load Balancers**.
6. Select **Lab6ELB** and on the **Description** tab in the lower pane, copy the **DNS name** of your load balancer, making sure to omit "(A Record)".

Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. You created Auto Scaling policies to increase and decrease the group by one instance. You created Amazon CloudWatch alarms to trigger these policies when the aggregate average CPU of the group is ≥ 65% and ≤ 20% respectively. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now monitor this infrastructure using the CloudWatch alarms that you created.

In this task you test the Auto Scaling configuration you implemented.

1. [60] On the **Services** menu, click **CloudWatch**.
2. In the navigation pane, click **Alarms** (*not* **ALARM**).
   The two alarms **High CPU Utilization** and **Low CPU Utilization** are displayed. **Low CPU Utilization** has a **State** of *ALARM* and **High CPU Utilization** has a **State** of *OK*. This is because the current group CPU Utilization is < 20%. Auto Scaling is not removing any instances because the group size is currently at its minimum (2).
3. Paste the load balancer's DNS name that you copied in Task 4 in a new browser window or tab and press *ENTER*.
4. Click **Load Test** under the AWS logo. The application load tests your instances and auto-refreshes in 5 seconds. The Current CPU Load jump to 100%. The **Load Test** link triggers a simple background process. Do not close this tab.
5. Return to the window or tab with the **AWS CloudWatch console**.
   In less than 5 minutes, the **Low CPU** alarm status changes to *OK* and the **High CPU** alarm status changes to *ALARM*. Click the refresh icon to see the changes.
6. On the **Services** menu, click **EC2**.
7. In the navigation pane, click **Instances**.
   More than two instances labeled **Lab 6 Web Instance** are now running. They may be in creation, and the tags may not appear immediately. The new instance was created by Auto Scaling based on the CloudWatch Alarm you created in an earlier step.

Task 6 (Optional): Terminate Web Server 1

In this task, you terminate Web Server 1 in Public Subnet 2. Your Auto Scaling group launched instances into private subnets, and the original publically accessible web server is no longer needed.

1. [67] On the **Services** menu, click **EC2**.
2. In the navigation pane, click **Instances**.
3. Right-click **Web Server 1**, and click **Instance State** > **Terminate**.
4. Click **Yes, Terminate**.

Lab Complete

Congratulations! You have successfully managed your architecture using Auto Scaling and Elastic Load Balancing. Cleanup your environment now.