Version 4.6.6 (TESS3) + custom change

This lab 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.

Auto Scaling helps you maintain application availability and allows you to scale your Amazon 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:

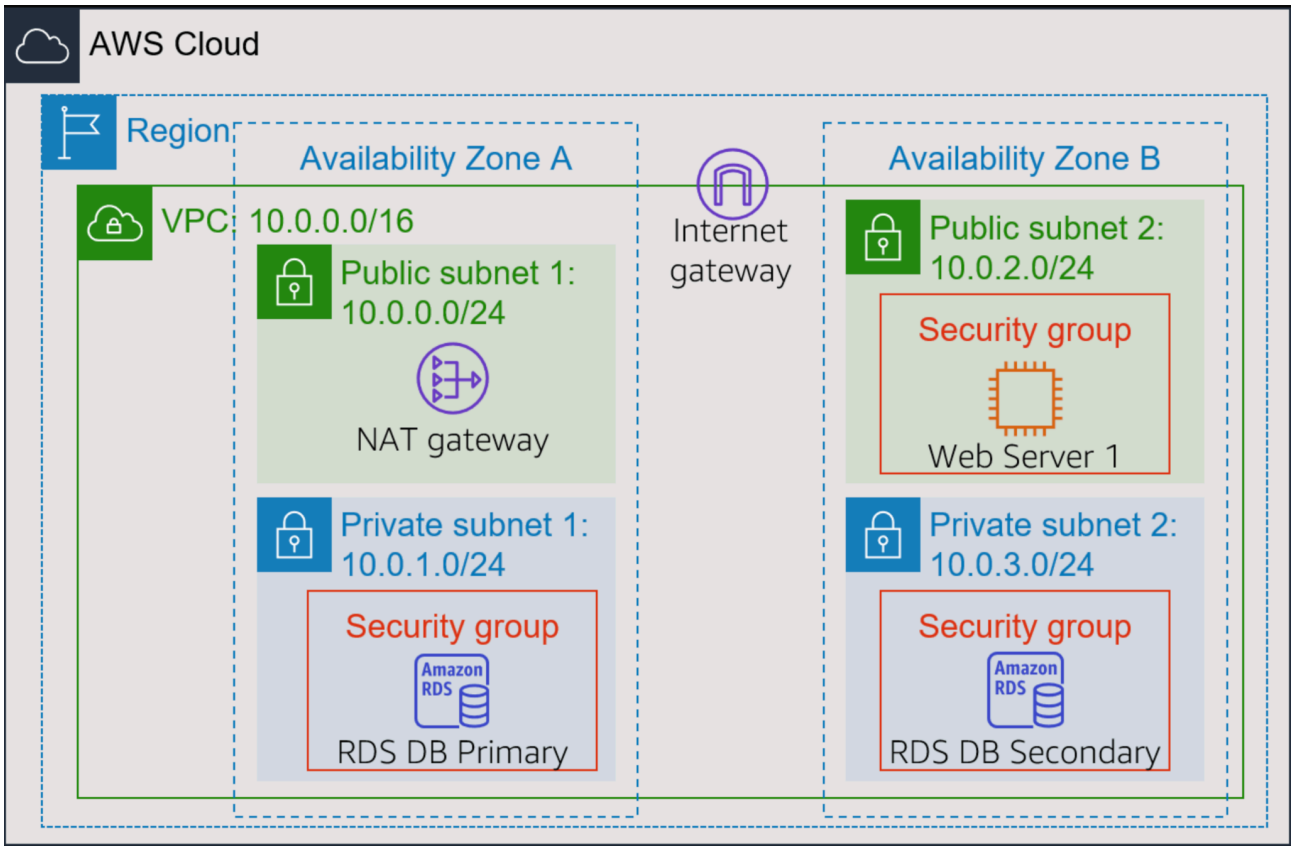
- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch configuration and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor the performance of your infrastructure.

Duration

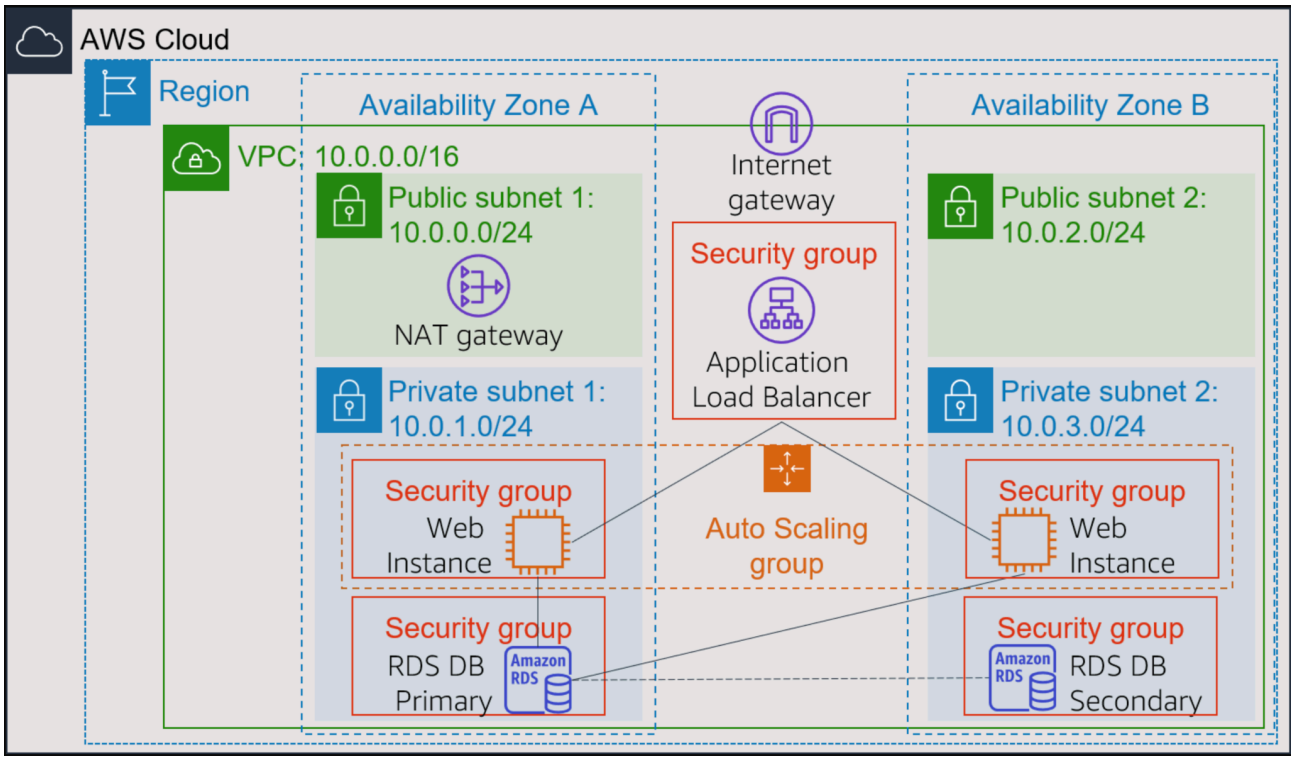
This lab takes approximately **30 minutes**.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:



Task 1: Create an AMI for Auto Scaling

In this task, you will create an AMI from the existing *Web Server 1*. This will save the contents of the boot disk so that new instances can be launched with identical content.

1. In the **AWS Management Console**, on the **Services** menu, click **EC2**.
2. In the left navigation pane, click **Instances**.
   Create a `WebServerAMI`
   You will now create an AMI based upon this instance.
3. Select **Web Server 1**.
4. In the **Actions** menu, click **Image and templates** > **Create image**, then configure:
   - **Image name:** `WebServerAMI`
   - **Image description:** `Lab AMI for Web Server`
5. Click **Create image**

A confirmation banner displays the **AMI ID** for your new AMI.

You will use this AMI when launching the Auto Scaling group later in the lab.

Task 2: Create a Load Balancer

In this task, you will create a load balancer that can balance traffic across multiple EC2 instances and Availability Zones.

6. In the left navigation pane, choose **Target Groups**.
   Analysis: *Target Groups* define where to *send* traffic that comes into the Load Balancer. The Application Load Balancer can send traffic to multiple Target Groups based upon the URL of the incoming request, such as having requests from mobile apps going to a different set of servers. Your web application will use only one Target Group.
   - Choose **Create target group**
   - Choose a target type: **Instances**
   - **Target group name**, enter: `LabGroup`
   - Select **Lab VPC** from the **VPC** drop-down menu.
7. Choose **Next**. The **Register targets** screen appears.
   Note: *Targets* are the individual instances that will respond to requests from the Load Balancer.
   You do not have any web application instances yet, so you can skip this step.
8. Review the settings and choose **Create target group**

9. In the left navigation pane, click **Load Balancers**.
10. At the top of the screen, choose **Create Load Balancer**.
    Several different types of load balancer are displayed. You will be using an *Application Load Balancer* that operates at the request level (layer 7), routing traffic to targets — EC2 instances, containers, IP addresses and Lambda functions — based on the content of the request. For more information, see: Comparison of Load Balancers
11. Under **Application Load Balancer**, choose **Create**
12. Under **Load balancer name**, enter: `LabELB`
13. Scroll down to the **Network mapping** section, then:
    - For **VPC**, select: **Lab VPC**
      You will now specify which *subnets* the Load Balancer should use. The load balancer will be internet facing, so you will select both Public Subnets.
    - Choose the **first** displayed Availability Zone, then select **Public Subnet 1** from the Subnet drop down menu that displays beneath it.
    - Choose the **second** displayed Availability Zone, then select **Public Subnet 2** from the Subnet drop down menu that displays beneath it.
      You should now have two subnets selected: **Public Subnet 1** and **Public Subnet 2**. (If not, go back and try the configuration again.)
14. In the **Security groups** section:
    - Choose the Security groups drop down menu and select **Web Security Group**
    - Below the drop down menu, choose the **X** next to the default security group to remove it.
      The **Web Security Group** security group should now be the only one that appears.
15. For the Listener HTTP:80 row, set the Default action to forward to **LabGroup**.
16. Scroll to the bottom and choose **Create load balancer**
    The load balancer is successfully created.
    - Choose **View load balancer**
17. The load balancer will show a state of *provisioning*. There is no need to wait until it is ready. Please continue with the next task.

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

In this task, you will 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, security group and disks.

18. In the left navigation pane, click **Launch Configurations**.
19. Click **Create launch configuration**
20. Configure these settings:
    - **Launch configuration name:** `LabConfig`
    - **Amazon Machine Image (AMI)** Choose *Web Server AMI*
    - **Instance type:**
        - Choose **Choose instance type**
        - Select *t3.micro*
        - Choose **Choose**
    - **Note:** If you have launched the lab in the us-east-1 Region, select the **t2.micro** instance type. To find the Region, look in the upper right-hand corner of the Amazon EC2 console.
      Note: If you receive the error message "Something went wrong. Please refresh and try again.", you may ignore it and continue with the exercise.
    - **Additional configuration**
        - **Monitoring:** Select *Enable EC2 instance detailed monitoring within CloudWatch*
    - This allows Auto Scaling to react quickly to changing utilization.
21. Under **Security groups**, you will configure the launch configuration to use the *Web Security Group* that has already been created for you.
    - Choose **Select an existing security group**
    - Select **Web Security Group**
22. Under **Key pair** configure:
    - **Key pair options:** *Choose an existing key pair*
    - **Existing key pair:** vockey
    - Select **I acknowledge...**
    - Click **Create launch configuration**
23. You will now create an Auto Scaling group that uses this Launch Configuration.
24. Select the checkbox for the *LabConfig* Launch Configuration.
25. From the **Actions** menu, choose *Create Auto Scaling group*
26. Enter Auto Scaling group name:
    - **Name:** `Lab Auto Scaling Group`
27. Choose **Next**
28. On the **Network** page configure
    - **Network:** *Lab VPC*
      You can ignore the message regarding "No public IP address"
    - **Subnet:** Select *Private Subnet 1 (10.0.1.0/24)* **and** *Private Subnet 2 (10.0.3.0/24)*
29. This will launch EC2 instances in private subnets across both Availability Zones.
30. Choose **Next**

31. In the **Load balancing - *optional*** pane, choose **Attach to an existing load balancer**
32. In the **Attach to an existing load balancer** pane, use the dropdown list to select *LabGroup*.
33. In the **Additional settings - *optional*** pane, select **Enable group metrics collection within CloudWatch**
    This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.
34. Choose **Next**
35. Under **Group size**, configure:
    - **Desired capacity:** 2
    - **Minimum capacity:** 2
    - **Maximum capacity:** 6
36. This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 6 instances running.
37. Under **Scaling policies**, choose *Target tracking scaling policy* and configure:
    - **Lab policy name:** `LabScalingPolicy`
    - **Metric type:** *Average CPU Utilization*
    - **Target value:** `60`
38. This tells Auto Scaling to maintain an *average* CPU utilization *across all instances* at 60%. Auto Scaling will automatically add or remove capacity as required to keep the metric at, or close to, the specified target value. It adjusts to fluctuations in the metric due to a fluctuating load pattern.
39. Choose **Next**
    Auto Scaling can send a notification when a scaling event takes place. You will use the default settings.
40. Choose **Next**
    Tags applied to the Auto Scaling group will be automatically propagated to the instances that are launched.
41. Choose **Add tag** and Configure the following:
    - **Key:** `Name`
    - **Value:** `Lab Instance`
42. Click **Next**
43. Review the details of your Auto Scaling group, then click **Create Auto Scaling group**.
    If you encounter an error **Failed to create Auto Scaling group**, then click **Retry Failed Tasks**.
    Your Auto Scaling group will initially show an instance count of zero, but new instances will be launched to reach the **Desired** count of 2 instances.

Task 4: Verify that Load Balancing is Working

In this task, you will verify that Load Balancing is working correctly.

44. In the left navigation pane, click **Instances**.
    You should see two new instances named **Lab Instance**. These were launched by Auto Scaling.
    If the instances or names are not displayed, wait 30 seconds and click refresh in the top-right.
    First, you will confirm that the new instances have passed their Health Check.
45. In the left navigation pane, click **Target Groups** (in the *Load Balancing* section).
46. Choose *LabGroup*
47. Click the **Targets** tab.
    Two **Lab Instance** targets should be listed for this target group.
48. Wait until the **Status** of both instances transitions to *healthy*. Click Refresh in the upper-right to check for updates.
    Healthy indicates that an instance has passed the Load Balancer's health check. This means that the Load Balancer will send traffic to the instance.
    You can now access the Auto Scaling group via the Load Balancer.
49. In the left navigation pane, click **Load Balancers**.
50. In the lower pane, copy the **DNS name** of the load balancer, making sure to omit "(A Record)".
    It should look similar to: *LabELB-1998580470.us-west-2.elb.amazonaws.com*
51. Open a new web browser tab, paste the DNS Name you just copied, and press Enter.
    The application should appear in your browser. This indicates that the Load Balancer received the request, sent it to one of the EC2 instances, then passed back the result.

Task 5: Test Auto Scaling

You created an Auto Scaling group with a minimum of two instances and a maximum of six instances. Currently two instances are running because the minimum size is two and the group is currently not under any load. You will now increase the load to cause Auto Scaling to add additional instances.

52. Return to the AWS management console, but do not close the application tab — you will return to it soon.

53. On the **Services** menu, click **CloudWatch**.
54. In the left navigation pane, choose **All alarms**.
    Two alarms will be displayed. These were created automatically by the Auto Scaling group. They will automatically keep the average CPU load close to 60% while also staying within the limitation of having two to six instances.
    Note: Please follow these steps only if you do not see the alarms in 60 seconds.
    - On the **Services** menu, click **EC2**.
    - In the left navigation pane, choose **Auto Scaling Groups**.
    - Select **Lab Auto Scaling Group**.
    - In the bottom half of the page, choose the **Automatic Scaling** tab.
    - Select **LabScalingPolicy**.
    - Click **Actions** and **Edit**.
    - Change the **Target Value** to `50`.
    - Click **Update**
    - On the **Services** menu, click **CloudWatch**.
    - In the left navigation pane, click **All alarms** and verify you see two alarms.

55. Click the **OK** alarm, which has *AlarmHigh* in its name.
    If no alarm is showing **OK**, wait a minute then click refresh in the top-right until the alarm status changes.
    The **OK** indicates that the alarm has *not* been triggered. It is the alarm for **CPU Utilization > 60**, which will add instances when average CPU is high. The chart should show very low levels of CPU at the moment.
    You will now tell the application to perform calculations that should raise the CPU level.
56. Return to the browser tab with the web application.
57. Click **Load Test** beside the AWS logo.
    This will cause the application to generate high loads. The browser page will automatically refresh so that all instances in the Auto Scaling group will generate load. Do not close this tab.
58. Return to browser tab with the **CloudWatch** console.
    In less than 5 minutes, the **AlarmLow** alarm should change to **OK** and the **AlarmHigh** alarm status should change to *In alarm*.
    You can click Refresh in the top-right every 60 seconds to update the display.
    You should see the **AlarmHigh** chart indicating an increasing CPU percentage.
    Once it crosses the 60% line for more than 3 minutes, it will trigger Auto Scaling to add additional instances.
59. Wait until the **AlarmHigh** alarm enters the *In alarm* state.
    You can now view the additional instance(s) that were launched.
60. On the **Services** menu, click **EC2**.
61. In the left navigation pane, click **Instances**.
    More than two instances labeled **Lab Instance** should now be running. The new instance(s) were created by Auto Scaling in response to the Alarm.

Task 6: Terminate Web Server 1

In this task, you will terminate *Web Server 1*. This instance was used to create the AMI used by your Auto Scaling group, but it is no longer needed.

62. Select **Web Server 1** (and ensure it is the only instance selected).
63. In the **Instance state** menu, click **Instance State** > **Terminate Instance**.
64. Choose **Terminate**

Lab Complete

Congratulations! You have completed the lab.