Scale and Load Balance Your Architecture

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:

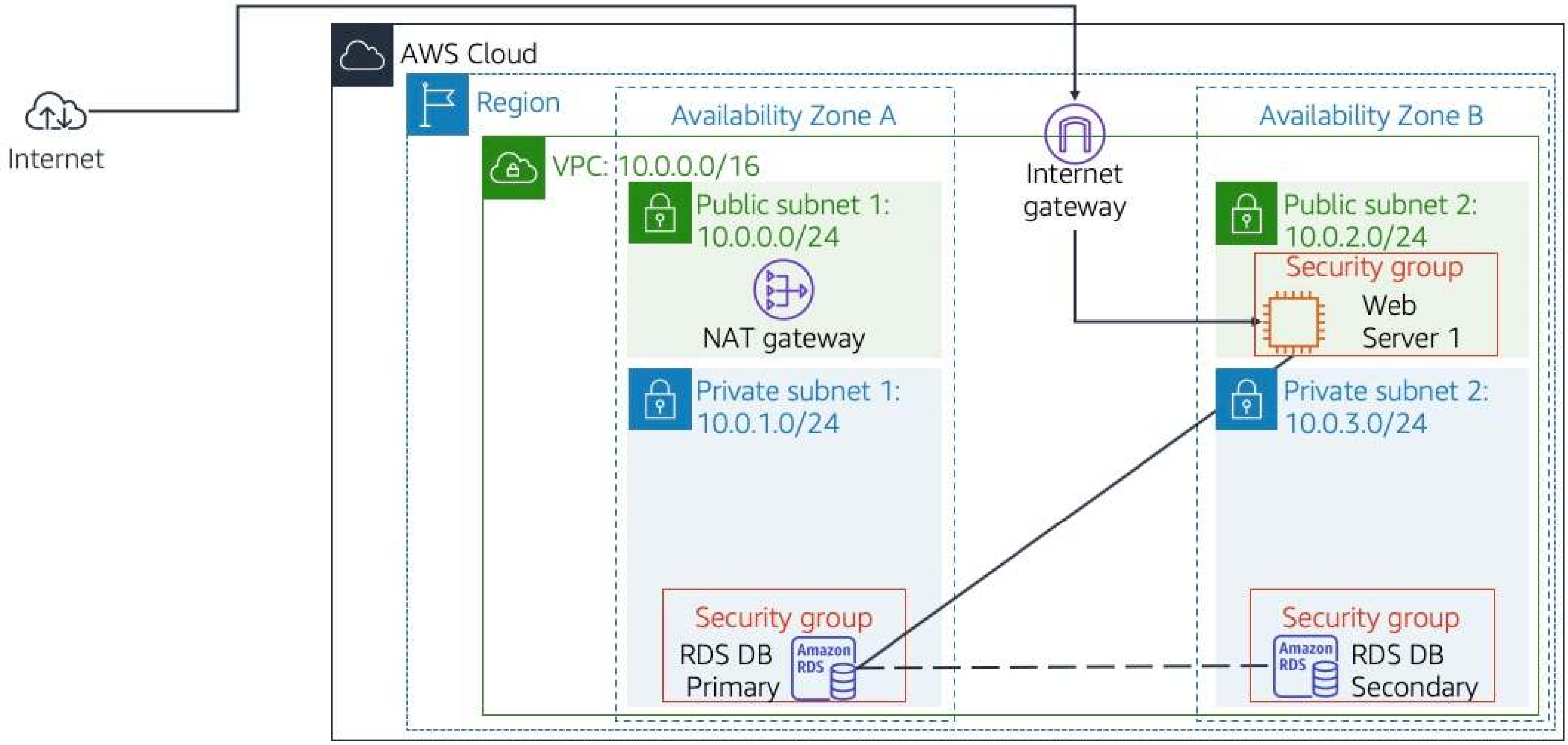
- Create an Amazon Machine Image (AMI) from a running instance.
- Create a load balancer.
- Create a launch template and an Auto Scaling group.
- Automatically scale new instances within a private subnet
- Create Amazon CloudWatch alarms and monitor performance of your infrastructure.

Duration

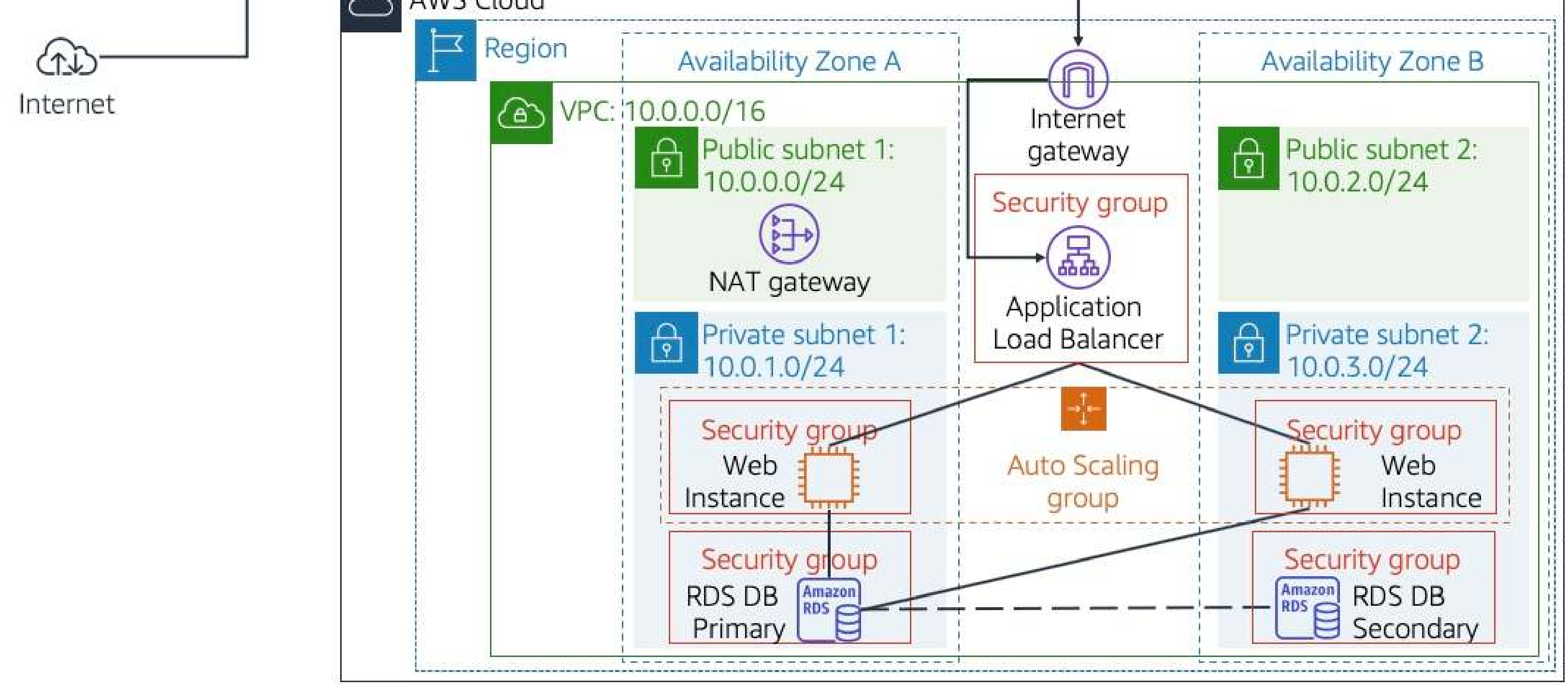
This lab takes approximately **45 minutes**.

Scenario

You start with the following infrastructure:



The final state of the infrastructure is:



Accessing the AWS Management Console

1. At the top of these instructions, click ▸ **Start Lab** to launch your lab.

   Tip: If you need more time to complete the lab, then restart the timer for the environment by choosing the ▸ **Start Lab** button again.

2. Lab resources will be displayed on the top left corner.

   Example:

   - **AWS** ● indicates that AWS lab resources are currently getting created.
   - **AWS** ● indicates that AWS lab resources are ready.

   Please wait for the lab to be ready, before proceeding.

3. At the top of these instructions, click **AWS** ●

   This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

   Tip: If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.

   ⚠ Do not change the lab region unless specifically instructed to do so.

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.

5. In the AWS Management Console, select the ▦ **Services** menu, and then select **EC2** under **Compute**.

6. In the left navigation pane, click **Instances**.

   First, you will confirm that the instance is running.

7. Wait until the **Status Checks** for **Web Server 1** displays *2/2 checks passed*. Click refresh ⟳ to update.

⚠ Proceeding before the instance is ready, would result in lab failure.

You will now create an AMI based upon this instance.

8. Select ☑ **Web Server 1**.
9. In the Actions ⌄ menu, click **Image and templates** > **Create image**, then configure:
   - **Image name:** `Web Server AMI`
   - **Image description:** `Lab AMI for Web Server`
10. Click **Create image**

    The confirmation screen displays the **AMI ID** for your new AMI.

11. Click **Close**

    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.

12. In the left navigation pane, click **Load Balancers**.
13. Click **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

14. Under **Application Load Balancer** click **Create** and configure:
    - **Name:** `LabELB`
    - **VPC:** *Lab VPC* (In the **Availability Zones** section)
    - **Availability Zones:** Select ☑ both to see the available subnets.
    - Select **Public Subnet 1** and **Public Subnet 2**

    This configures the load balancer to operate across multiple Availability Zones.

15. Click **Next: Configure Security Settings**

    💬 You can ignore the *"Improve your load balancer's security."* warning.

16. Click **Next: Configure Security Groups**

    A *Web Security Group* has already been created for you, which permits HTTP access.

17. Select ☑ **Web Security Group** and deselect ☐ **default**.
18. Click **Next: Configure Routing**

    Routing configures where to send requests that are sent to the load balancer. You will create a *Target Group* that will be used by Auto Scaling.

19. For **Name**, enter: `LabGroup`
20. Click **Next: Register Targets**

    Auto Scaling will automatically register instances as targets later in the lab.

21. Click **Next: Review**
22. Click **Create** then click **Close**

    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 Template and an Auto Scaling Group

In this task, you will create a *launch template* for your Auto Scaling group. A launch template is a template that an Auto Scaling group uses to launch EC2 instances. When you create a launch template, you specify information for the instances such as the AMI, the instance type, a key pair, security group and disks.

23. In the left navigation pane, click **Launch Templates**.
24. Click **Create launch template**
25. Configure these settings:
    - **Launch template name:** `LabTemplate`
    - **AMI:** `Web Server AMI`
    - **Instance type:** t3.micro.
    - **Key pair name:** vockey.
    - **Security groups:** Web Security Group. *Make sure the security group belongs to Lab VPC.*
    - Expand **Advanced details** and in the **Detailed CloudWatch monitoring** menu, select **Enable**

      This will capture metrics at 1-minute intervals, which allows Auto Scaling to react quickly to changing usage patterns.
26. Click **Create launch template** followed by **View launch templates**

    You will now create an Auto Scaling group that uses this Launch Template.
27. Select ◉ **LabTemplate** and then in the **Actions ⌄** menu, select **Create Auto Scaling group**
28. Configure the following settings:
    - **Auto Scaling group name:** `Lab Auto Scaling Group`
    - Click **Next**
    - **Network:** *Lab VPC*
    - **Subnet:** Select *Private Subnet 1 (10.0.1.0/24)* **and** *Private Subnet 2 (10.0.3.0/24)*
    - Click **Next**
    - **Load balancing - optional**
        - Select ☑ **Attach to an existing load balancer**
        - Select ◉ **Choose from your load balancer target groups**.
    - **Existing load balancer target groups** *LabGroup*
    - **Monitoring:** 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.
    - Click **Next**
    - **Group size:** Enter the below values
        - Desired capacity: `2`
        - Minimum capacity: `2`
        - Maximum capacity: `4`

      This will allow Auto Scaling to automatically add/remove instances, always keeping between 2 and 4 instances running.
    - **Scaling policies - optional**
        - Select ◉ **Target tracking scaling policy**
        - **Metric type:** *Average CPU Utilization*
        - **Target value:** `60`

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.

- Click **Next**
- Click **Next** again on the **Add notifications** section.
- **Add tags:** click **Add tag** and enter
  - **Key:** `Name`
  - **Value:** `Lab Instance`
- Click **Next**
- Finally click **Create Auto Scaling group**

  This will launch EC2 instances in private subnets across both Availability Zones.

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.
Note: If you experience an error related to the t3.micro instance type not being available, then rerun this task by selecting t2.micro instead.

Task 4: Verify that Load Balancing is Working

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

29. 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.

30. In the left navigation pane, click **Target Groups** (in the *Load Balancing* section).
31. Click **LabGroup** followed by the **Targets** tab.

    Two **Lab Instance** targets should be listed for this target group.

32. 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.

33. In the left navigation pane, click **Load Balancers**.
34. 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*

35. 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 four 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.

36. Return to the AWS management console, but do not close the application tab — you will return to it soon.
37. In the AWS Management Console, select the ▦ **Services** menu, and then select **CloudWatch** under **Management & Governance**.
38. In the left navigation pane, click **Alarms** (*not* **ALARM**).

    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.

39. 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.

40. Return to the browser tab with the web application.
41. 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.

42. 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 *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.

43. Wait until the **AlarmHigh** alarm enters the *ALARM* state.

    You can now view the additional instance(s) that were launched.

44. In the AWS Management Console, select the ▦ **Services** menu, and then select **EC2** under **Compute**.
45. 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.

46. Select ☑ **Web Server 1** (and ensure it is the only instance selected).
47. In the **Actions ⌄** menu, click **Instance State** > **Terminate**.
48. Click Yes, Terminate

Lab Complete 🎓

🏁 Congratulations! You have completed the lab.

49. Choose ■ **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

    A panel indicates that *DELETE has been initiated... You may close this message box now.*

50. A message *Ended AWS Lab Successfully* is briefly displayed, indicating that the lab has ended.