

Making Your Environment Highly Available

Critical business systems should be deployed as *Highly Available applications*, meaning that they can remain operational even when some components fail. To achieve High Availability in AWS, it is recommended to **run services across multiple Availability Zones**.

Many AWS services are inherently highly available, such as Load Balancers, or can be configured for high availability, such as deploying Amazon EC2 instances in multiple Availability Zones.

In this lab, you will start with an application running on a single Amazon EC2 instance and will then convert it to be Highly Available.

Objectives

After completing this lab, you will be able to:

- Create an image of an existing Amazon EC2 instance and use it to launch new instances.
- Expand an Amazon VPC to additional Availability Zones.
- Create VPC Subnets and Route Tables.
- Create an AWS NAT Gateway.
- Create a Load Balancer.
- Create an Auto Scaling group.

Duration

The lab requires approximately **60 minutes** to complete.

Access the AWS Management Console

Task 1: Inspect Your environment

This lab begins with an environment already deployed via Amazon CloudFormation including:

- An Amazon VPC
- A Public and Private Subnet in one Availability Zone
- An Internet Gateway associated with the Public Subnet
- A NAT Gateway in the Public Subnet
- An Amazon EC2 instance in the Public Subnet

Task 1.1: Inspect Your VPC

In this task, you will review the configuration of the VPC that has already been created.

4. [4]On the **AWS Management Console**, on the **Services** menu, click **VPC**.

5. [5]In the left navigation pane, click **Your VPCs**.

Here you can see the **Lab VPC** that has been created for you:

- In the **CIDR** column, you can see a value of **10.200.0.0/20**, which means this VPC includes 4,096 IPs between 10.200.0.0 and 10.200.15.255 (with some reserved and unusable).
- It is also attached to a **Route Table** and a **Network ACL**.
- This VPC also has a **Tenancy** of *default*, instances launched into this VPC will by default use shared tenancy hardware.

6. [6]In the navigation pane, click **Subnets**.

Here you can see the **Public Subnet 1** subnet:

- In the **VPC** column, you can see that this subnet exists inside of **Lab VPC**.
- In the **IPv4 CIDR** column, you can see a value of **10.200.0.0/24**, which means this subnet includes the 256 IPs (5 of which are reserved and unusable) between 10.200.0.0 and 10.200.0.255.
- In the **Availability Zone** column, you can see the Availability Zone in which this subnet resides.

7. [7]Click on the row containing **Public Subnet 1** to reveal more details at the bottom of the page.

8. [8]Click the **Route Table** tab in the lower half of the window.

Here you can see details about the Routing for this subnet:

- The first entry specifies that traffic destined within the VPC's CIDR range (**10.200.0.0/20**) will be routed within the VPC (**local**).
- The second entry specifies that any traffic destined for the Internet (**0.0.0.0/0**) is routed to the Internet Gateway (*igw-*). This setting makes it a *Public Subnet*.

9. [9]Click the **Network ACL** tab in the lower half of the window.

Here you can see the Network Access Control List (ACL) associated with the subnet. The rules currently permit *ALL Traffic* to flow in and out of the subnet, but they can be further restricted by using Security Groups.

10. [10]In the left navigation pane, click **Internet Gateways**.

Notice that an Internet Gateway is already associated with Lab VPC.

11. [11]In the navigation pane, click **Security Groups**.

12. [12]Click **Configuration Server SG**.

This is the security group used by the Configuration Server.

13. [13]Click the **Inbound Rules** tab in the lower half of the window.

Here you can see that this Security Group only allows traffic via SSH (TCP port 22) and HTTP (TCP port 80).

14. [14]Click the **Outbound Rules** tab.

Here you can see that this Security Group allows all outbound traffic.

Task 1.2: Inspect Your Amazon EC2 Instance

In this task, you will inspect the Amazon EC2 instance that was launched for you.

15. [15]On the **Services** menu, click **EC2**.
16. [16]In the left navigation pane, click **Instances**.

Here you can see that a **Configuration Server** is already running. In the **Description** tab in the lower half of the window, you can see the details of this instance, including its public and private IP addresses and the Availability zone, VPC, Subnet, and Security Groups.

17. [17]Copy the **IPv4 Public IP** value and paste it into a text editor, such as Notepad. You will use it later.
18. [18]In the **Actions** menu, click **Instance Settings > View/Change User Data**.

Note that no User Data appears! This means that the instance has not yet been configured to run your web application. When launching an Amazon EC2 instance, you can provide a **User Data script** that is executed when the instance first starts and is used to configure the instance. However, in this lab you will configure the instance yourself!

19. [19]Click **Cancel** to close the User Data dialog box.

Task 2: Login to your Amazon EC2 instance

Task 3: Download, Install, and Launch Your Web Server's PHP Application

In this task, you will be performing typical System Administrator activities to install and configure the web application. In a following task, you will create an *image* of this machine to automatically deploy the application on more instances to make it *Highly Available*.

The commands in this task will download, install, and launch your PHP web application. The instructions will step you through each command one at a time so you can understand exactly what you are doing to accomplish this task.

34. [34]To update the base software installed your instance, execute the following command:

```
sudo yum -y update
```

This will discover what updates are available for your Amazon Linux instance, download the updates, and install them.

Tip for PuTTY users: Simply right-click to Paste.

35. [35]To install a package that creates a web server, execute the following command:

```
sudo yum -y install httpd php
```

This command installs an Apache web server (httpd) and the PHP language interpreter.

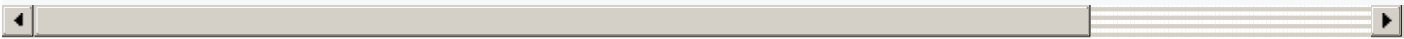
36. [36]Execute the following command:

```
sudo chkconfig httpd on
```

This configures the Apache web server to automatically start when the instance starts.

37. [37]Execute the following command:

```
wget https://us-west-2-aws-training.s3.amazonaws.com/awsu-ilt/AWS-100-ARC/v5.2/lab-2-ha,
```



This downloads a zip file containing the PHP web application.

38. [38]Execute the following command:

```
sudo unzip phpapp.zip -d /var/www/html/
```

This unzips the PHP application into the default Apache web server directory.

39. [39]Execute the following command:

```
sudo service httpd start
```

This starts the Apache web server.

You can ignore any warnings about "Could not reliably determine..."

Your web application is now configured! You can now access application to confirm that it is working.

40. [40]Open a new web browser tab, paste the **Public IP** address for your instance in the address bar and hit **Enter**. (That is the same IP address you copied into a Text Editor and used with ssh/PuTTY.)

The web application should appear and will display information about your location (actually, the location of your Amazon EC2 instance). This information is obtained from freegeoip.net.

If the application does not appear, ask your instructor for assistance in diagnosing the configuration.

41. [41]Close the web application browser tab that you opened in the previous step.

42. [42]Return to your SSH session, execute the following command:

```
exit
```

This ends your SSH session.

Task 4: Create an Amazon Machine Image (AMI)

Now that your web application is configured on your instance, you will **create an Amazon Machine Image (AMI)** of it. An AMI is a copy of the disk volumes attached to an Amazon EC2 instance. When a new instance is launched from an AMI, the disk volumes will contain exactly the same data as the original instance.

This is an excellent way to *clone* instances to run an application on multiple instances, even across multiple Availability Zones.

In this task, you will **create an AMI** from your Amazon EC2 instance. You will later use this image to launch additional, fully-configured instances to provide a Highly Available solution.

43. [43]Return to the web browser tab showing the EC2 Management Console.
44. [44]Ensure that your **Configuration Server** is selected, and click **Actions > Image > Create Image**

You will see that a Root Volume is currently associated with the instance. This volume will be copied into the AMI.

45. [45]For **Image name**, type: `Web application`
46. [46]Click **Create Image**.
47. [47]Leave other values at their default settings and click **Close**.

The AMI will be created in the background and you will use it in a later step. There is no need to wait while it is being created.

Task 5: Configure a Second Availability Zone

To build a Highly Available application, it is best practice to launch resources in **multiple Availability Zones**. Availability Zones are physically separate data centers (or groups of data centers) within the same Region. Running your applications across multiple Availability Zones will provide greater *availability* in case of failure within a data center.

In this task, you will duplicate your network environment into a second Availability Zone. You will create:

- A second Public Subnet
- A second Private Subnet
- A second NAT Gateway
- A second Private Route Table

Task 5.1: Create a second Public Subnet

48. [48]On the **Services** menu, click **VPC**.
49. [49]In the left navigation pane, click **Subnets**.
50. [50]In the row for **Public Subnet 1**, take note of the value for **Availability Zone**. (You might need to scroll

sideways to see it.)

The name of an Availability Zone consists of the Region name (eg *us-west-2*) plus a zone identifier (eg *a*). Together, this Availability Zone has a name of *us-west-2a*.

51. [51]Click **Create Subnet**.
52. [52]In the **Create Subnet** dialog box, configure the following:

Name tag	Public Subnet 2
VPC	Lab VPC
Availability Zone	Choose a <i>different Availability Zone</i> from the existing Subnet (eg if it was <i>a</i> , then choose <i>b</i>).
IPv4 CIDR block	10.200.1.0/24

This will create a second Subnet in a different Availability Zone, but still within **Lab VPC**. It will have an IP range between 10.200.1.0 and 10.200.1.255.

53. [53]Click **Yes, Create**.
54. [54]With **Public Subnet 2** selected, click the **Route Table** tab in the lower half of the window. (*Do not* click the Route Tables link in the left navigation pane.)

Here you can see that your new Subnet has been provided with a default Route Table, but this Route Table does not have a connection to your Internet gateway.

55. [55]Click **Edit**.
56. [56]In the **Change to:** drop-down list, click **Public Route Table**.
57. [57]Click **Save**.

Public Subnet 2 is now a *Public Subnet* that can communicate directly with the Internet.

Task 5.2: Create a Second Private Subnet

Your application will be deployed in *private subnets* for improved security. This prevents direct access from the Internet to the instances. To be highly available, you require a second Private Subnet.

58. [58]Click **Create Subnet**.
59. [59]In the **Create Subnet** dialog box, configure the following:

Name tag	Private Subnet 2
VPC	Lab VPC

Availability Zone	Choose the same Availability Zone you just selected <i>Public Subnet 2</i> .
IPv4 CIDR block	<code>10.200.4.0/23</code>

60. [60]Click **Yes, Create**.

The Subnet will have an IP range between 10.200.4.0 and 10.200.5.255.

Task 5.3: Create a Second NAT Gateway

A NAT Gateway (*Network Address Translation*) is provisioned into a Public Subnet and provides **outbound Internet connectivity** for resources in a Private Subnet. Your web application requires connectivity to the Internet to retrieve geographic information, so you will need to route Internet-bound traffic through a NAT Gateway.

To remain *Highly Available*, your web application must be configured such that any problems in the first Availability Zone should not impact resources in the second Availability Zone, and vice versa. Therefore, you will create a second NAT Gateway in the second Availability Zone.

61. [61]In the left navigation pane, click **NAT Gateways**.

62. [62]Click **Create NAT Gateway**.

63. [63]For **Subnet**, select **Public Subnet 2**.

64. [64]Click **Create New EIP**.

An Elastic IP Address (EIP) is a static IP address that will be associated with this NAT Gateway. An Elastic IP address will remain unchanged over the life of the NAT Gateway.

65. [65]Click **Create a NAT Gateway**, then click **Close**.

66. [66]Click **View NAT Gateways**.

You will now see two NAT Gateways. (If you only see one, click the refresh icon in the top-right until both appear.)

67. [67]Find the NAT Gateway that you just created -- it will have a status of **Pending** and a Private IP Address starting with `10.200.1`.

68. [68]Copy the **NAT Gateway ID** show in the first column, starting with `nat-`. Paste it into a text document for use in the next task.

You must now configure your network to use the second NAT Gateway.

Task 5.4: Create a Second Private Route Table

A Route Table defines how traffic flows into and out of a Subnet. You will now create a Route Table for *Private Subnet 2* that sends Internet-bound traffic through the NAT Gateway that you just created.

69. [69]In the navigation pane, click **Route Tables**.

70. [70]Click **Create Route Table**.

71. [71]In the **Create Route Table** dialog box, configure the following:

Name tag	Private Route Table 2
VPC	Lab VPC

72. [72]Click **Yes, Create**.

73. [73]Click the **Routes** tab in the lower half of the window.

The Route Table currently only sends traffic within the VPC, as identified by the Target of *local*. You will now configure the Route Table to send Internet-bound traffic (identified with the wildcard **0.0.0.0/0**) through the second NAT Gateway.

74. [74]Click **Edit**.

75. [75]Click **Add another route**.

76. [76]For **Destination**, type: **0.0.0.0/0**

77. [77]Click in the **Target** box, and then click the NAT Gateway with the ID you copied earlier. (Check your text editor for the *nat*-ID you saved earlier.)

78. [78]Click **Save**.

You can now associate this Route Table (*Private Route Table 2*) with the second *Private Subnet 2* that you created earlier.

79. [79]With **Private Route Table 2** still selected, click the **Subnet Associations** tab at the bottom of the screen.

80. [80]Click **Edit**.

81. [81]Select (tick) the checkbox beside **Private Subnet 2**.

82. [82]Click **Save**.

Private Subnet 2 will now route Internet-bound traffic through the second NAT Gateway.

Task 6: Create an Application Load Balancer

In this task, you will create an **Application Load Balancer** that distributes requests across multiple Amazon EC2 instances. This is a critical component of a Highly Available architecture because the Load Balancer performs health checks on instances and only sends requests to healthy instances.

You do not have any instances yet – they will be created by the Auto Scaling group in the next task.

83. [83]On the **Services** menu, click **EC2**.

84. [84]In the left navigation pane, click **Load Balancers** (you might need to scroll down to find it).

85. [85]Click **Create Load Balancer**.

Several types of Load Balancers are displayed. Read the descriptions of each type to understand their

capabilities.

86. [86]Under **Application Load Balancer**, click **Create**.
87. [87]For **Name**, type: LB1
88. [88]Scroll down to the **Availability Zones** section.
89. [89]For **VPC**, select **Lab VPC**.

You will now specify which subnets the Load Balancer should use. It will be an Internet-facing load balancer, so you will select both Public Subnets.

90. [90]Click the **first** displayed Availability Zone, then click the **Public Subnet** displayed underneath.
91. [91]Click the **second** displayed Availability Zone, then click the **Public Subnet** displayed underneath.

You should now have two subnets selected: **Public Subnet 1** and **Public Subnet 2**. (If not, go back and try the configuration again.)

92. [92]Click **Next: Configure Security Settings**.

A warning is displayed, which recommends using HTTPS for improved security. This is good advice, but is not necessary for this lab.

93. [93]Click **Next: Configure Security Groups**.
94. [94]Select the Security Group with a Description of **Security group for the web servers**.

This Security Group permits only HTTP incoming traffic, so it can be used on both the Load Balancer and the Web Servers.

95. [95]Click **Next: Configure Routing**.

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. Your web application will use only one Target Group.

96. [96]For **Name**, type: Group1
97. [97]Click **Advanced health check settings** to expand it.

The Application Load Balancer automatically performs Health Checks on all instances to ensure that they are healthy and are responding to requests. The default settings are recommended, but you will make them slightly faster for use in this lab.

98. [98]For **Healthy threshold**, type: 2
99. [99]For **Interval**, type: 10

This means that the Health Check will be performed every 10 seconds and if the instance responds correctly twice in a row, it will be considered healthy.

100. [100]Click **Next: Register Targets**.

Targets are 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.

101. [101]Click **Next: Review**.
102. [102]Review the settings and click **Create**.
103. [103]Click **Close**.

Your Load Balancer will now be provisioned in the background. You can now create an Auto Scaling group to launch your Amazon EC2 instances.

Task 7: Create an Auto Scaling Group

Auto Scaling is a service designed to launch or terminate Amazon EC2 instances automatically based on user-defined policies, schedules, and health checks. It also **automatically distributes instances across multiple Availability Zones** to make applications *Highly Available*.

In this task, you will create an Auto Scaling group that deploys Amazon EC2 instances across *you***Private Subnets**. This is best practice security for deploying applications because instances in a private subnet cannot be accessed from the Internet. Instead, users will send requests to the Load Balancer, which will forward the requests to Amazon EC2 instances in the private subnets.

104. [104]In the left navigation pane, click **Auto Scaling Groups** (you might need to scroll down to find it).
105. [105]Click **Create Auto Scaling group**.
106. [106]Click **Create launch configuration**.

A *Launch Configuration* defines what type of instances should be launched by Auto Scaling. The interface looks similar to launching an Amazon EC2 instance, but rather than launching an instance it stores the configuration for later use.

You will configure the Launch Configuration to use the AMI that you created earlier. It contains a copy of the software that you installed on the Configuration Server.

107. [107]In the left navigation pane, click **My AMIs**.
108. [108]In the row for **Web application**, click **Select**.
109. [109]Click **Next: Configure Details** to accept the default (**t2.micro**).
110. [110]For **Name**, type: `Web-Configuration`
111. [111]Click **Next: Add Storage**.

You do not require additional storage on this instance, so you are leaving the settings as their default.

112. [112]Click **Next: Configure Security Group**.
113. [113]Click **Select an existing security group**
114. [114]Select the Security Group with a Description of **Security group for the web servers**.
115. [115]Click **Review**.

You may receive a warning that you will not be able to connect to the instance via SSH. This is acceptable because the server configuration is already defined on the AMI and there is no need to login to the instance.

116. [116]Click **Continue** to dismiss the warning.
117. [117]Review the settings, then click **Create launch configuration**.
118. [118]When prompted, accept the keypair you have download earlier, select the acknowledgement check box, then click **Create launch configuration**.

You will now be prompted to create the Auto Scaling group. This includes defining the *number* of instances and *where* they should be launched.

119. [119]In the **Create Auto Scaling Group** page, configure the following settings:

Group Name	Web application
Group Size	Start with 2 instances
Network	Lab VPC
Subnet	Click in the box and select both Private Subnet 1 and Private Subnet 2

Auto Scaling will automatically distribute the instances amongst the selected Subnets, with each Subnet in a different Availability Zone. This is excellent for maintaining High Availability because the application will survive the failure of an Availability Zone.

120. [120]Click the **Advanced Details** heading to expand it.
121. [121]Select (tick) the **Load Balancing** checkbox.
122. [122]Click in **Target Groups**, then select **Group1**.
123. [123]**Next: Configure scaling policies**.
124. [124]Ensure **Keep this group at its initial size** is selected.

This configuration tells Auto Scaling to always maintain two instances in the Auto Scaling group. This is ideal for a Highly Available application because the application will continue to operate even if one instance fails. In such an event, Auto Scaling will automatically launch a replacement instance.

125. [125]Click **Next: Configure Notifications**.

You will not be configuring any notifications.

126. [126]Click **Next: Configure Tags**.

Tags placed on the Auto Scaling group can also automatically propagate to the instances launched by Auto Scaling.

127. [127]For **Key**, type: Name
128. [128]For **Value**, type: Web application
129. [129]Click **Review**.
130. [130]Review the settings, then click **Create Auto Scaling group**.
131. [131]Click **Close**.

Your Auto Scaling group will initially show zero instances. This should soon update to two instances. (Click the refresh icon in the top-right to update the display.)

Your application will soon be running across two Availability Zones and Auto Scaling will maintain that configuration even if an instance or Availability Zone fails.

Task 8: Test the Application

In this task, you will confirm that your web application is running and you will test that it is Highly Available.

132. [132]In the left navigation pane, select **Target Groups**.

133. [133]Click the **Targets** tab in the lower half of the window.

You should see two Registered instances. The Status column shows the results of the Load Balancer Health Check that is performed against the instances.

134. [134]Occasionally click the refresh icon in the top-right until the **Status** for both instances appears as *healthy*.

If the status does not eventually change to *healthy*, ask your instructor for assistance in diagnosing the configuration. Hovering over the icon in the Status column will provide more information about the status.

You will be testing the application by connecting to the Load Balancer, which will then send your request to one of the Amazon EC2 instances. You will need to retrieve the DNS Name of the Load Balancer.

135. [135]In the left navigation pane, click **Load Balancers**.

136. [136]In the **Description** tab in the lower half of the window, copy the **DNS Name** to your clipboard, but do not copy "(A Record)". It should be similar to: *LB1-xxxx.elb.amazonaws.com*

137. [137]Open a new web browser tab, paste the DNS Name from your clipboard and hit Enter.

The Load Balancer forwarded your request to one of the Amazon EC2 instances. The Instance ID and Availability Zone are shown at the bottom of the web application.

138. [138]Reload the page in your web browser. You should notice that the Instance ID and Availability Zone sometimes changes between the two instances.

The flow of information when displaying this web application is:

- You sent the request to the Load Balancer, which resides in the *public subnets* that are connected to the Internet.
- The **Load Balancer** chose one of the Amazon EC2 instances that reside in the *private subnets* and forwarded the request to it.
- The **Amazon EC2** instance requested geographic information from freegeoip.net. This request went out to the Internet through the **NAT Gateway** in the same Availability Zone as the instance.
- The Amazon EC2 instance then returned the web page to the Load Balancer, which returned it to your web browser.

Task 9: Test High Availability

Your application has been configured to be Highly Available. This can be proven by stopping one of the Amazon EC2 instances.

139. [139]Return to the EC2 Management Console tab in your web browser (but do not close the web application tab - you will return to it soon).
140. [140]In the left navigation pane, click **Instances**.

First, you do not require the Configuration Server any longer, so it can be terminated.

141. [141]Select the **Configuration Server**.
142. [142]Click **Actions > Instance State > Terminate** then click **Yes, Terminate**.

Next, stop one of the Web application instances to simulate a failure.

143. [143]Select one of the instances named **Web application** (it does not matter which one you select).
144. [144]Click **Actions > Instance State > Stop** then click **Yes, Stop**.

In a short time, the Load Balancer will notice that the instance is not responding and will automatically route all requests to the remaining instance.

145. [145]Return to the Web application tab in your web browser and reload the page several times.

You should notice that the **Availability Zone** shown at the bottom of the page stays the same. Even though an instance has failed, your application remains available.

After a few minutes, Auto Scaling will also notice the instance failure. It has been configured to keep two instances running, so Auto Scaling will **automatically launch a replacement instance**.

146. [146]Return to the EC2 Management Console tab in your web browser. Click the refresh icon in the top-right occasionally until a new Amazon EC2 instance appears.

After a few minutes, the Health Check for the new instance should become healthy and the Load Balancer will continue sending traffic between two Availability Zones. You can reload your Web application tab to see this happening.

This demonstrates that your application is now Highly Available.

Lab Complete.

Congratulations! You have completed the lab.