* [Lab 2 - Making Your Environment Highly Available](https://globalknowledge.qwiklab.com/focuses/17008" \l "lab-2-making-your-environment-highly-available)
  + [Accessing the AWS Management Console](https://globalknowledge.qwiklab.com/focuses/17008#accessing-the-aws-management-console)
  + [Task 1: Inspect Your environment](https://globalknowledge.qwiklab.com/focuses/17008#task-1-inspect-your-environment)
    - [Task 1.1: Inspect Your VPC](https://globalknowledge.qwiklab.com/focuses/17008#task-1-1-inspect-your-vpc)
    - [Task 1.2: Inspect Your Amazon EC2 Instance](https://globalknowledge.qwiklab.com/focuses/17008#task-1-2-inspect-your-amazon-ec2-instance)
  + [Task 2: Start Your Web Server's PHP App](https://globalknowledge.qwiklab.com/focuses/17008#task-2-start-your-web-server-s-php-app)
    - [Task 2.1: Connect to Web Server 1 (Windows Only)](https://globalknowledge.qwiklab.com/focuses/17008#task-2-1-connect-to-web-server-1-windows-only)
    - [Task 2.2: Connect to Web Server 1 (Mac OSX/Linux only)](https://globalknowledge.qwiklab.com/focuses/17008#task-2-2-connect-to-web-server-1-mac-osx-linux-only)
    - [Task 2.3: Download, Install, and Launch Your Web Server's PHP Application](https://globalknowledge.qwiklab.com/focuses/17008#task-2-3-download-install-and-launch-your-web-server-s-php-application)
  + [Task 3: Launch a Duplicate of Web Server 1 into a Second Availability Zone](https://globalknowledge.qwiklab.com/focuses/17008#task-3-launch-a-duplicate-of-web-server-1-into-a-second-availability-zone)
    - [Task 3.1: Create an Amazon Machine Image of Web Server 1](https://globalknowledge.qwiklab.com/focuses/17008#task-3-1-create-an-amazon-machine-image-of-web-server-1)
    - [Task 3.2: Create PublicSubnet2](https://globalknowledge.qwiklab.com/focuses/17008#task-3-2-create-publicsubnet2)
    - [Task 3.3: Launch Your Second Web Server](https://globalknowledge.qwiklab.com/focuses/17008#task-3-3-launch-your-second-web-server)
  + [Task 4: Launch and Attach a Load Balancer](https://globalknowledge.qwiklab.com/focuses/17008#task-4-launch-and-attach-a-load-balancer)
    - [Task 4.1: Create and attach a Load Balancer with Amazon ELB](https://globalknowledge.qwiklab.com/focuses/17008#task-4-1-create-and-attach-a-load-balancer-with-amazon-elb)
  + [Task 5: Creating a Private Application Environment](https://globalknowledge.qwiklab.com/focuses/17008#task-5-creating-a-private-application-environment)
  + [Task 5.1: Create Two Private Subnets](https://globalknowledge.qwiklab.com/focuses/17008#task-5-1-create-two-private-subnets)
    - [Task 5.2: Create and Attach an AWS NAT Gateway](https://globalknowledge.qwiklab.com/focuses/17008#task-5-2-create-and-attach-an-aws-nat-gateway)
    - [Task 5.3: Launch Two Instances into Your Private Subnets](https://globalknowledge.qwiklab.com/focuses/17008#task-5-3-launch-two-instances-into-your-private-subnets)
    - [Task 5.4: SSH into Web Server 1 again (Windows Only)](https://globalknowledge.qwiklab.com/focuses/17008#task-5-4-ssh-into-web-server-1-again-windows-only)
    - [Task 5.5: Connect to Web Server 1 again (Mac OSX/Linux only)](https://globalknowledge.qwiklab.com/focuses/17008#task-5-5-connect-to-web-server-1-again-mac-osx-linux-only)
    - [Task 5.6: Test Your NAT Gateway](https://globalknowledge.qwiklab.com/focuses/17008#task-5-6-test-your-nat-gateway)
  + [Lab Complete](https://globalknowledge.qwiklab.com/focuses/17008#lab-complete)

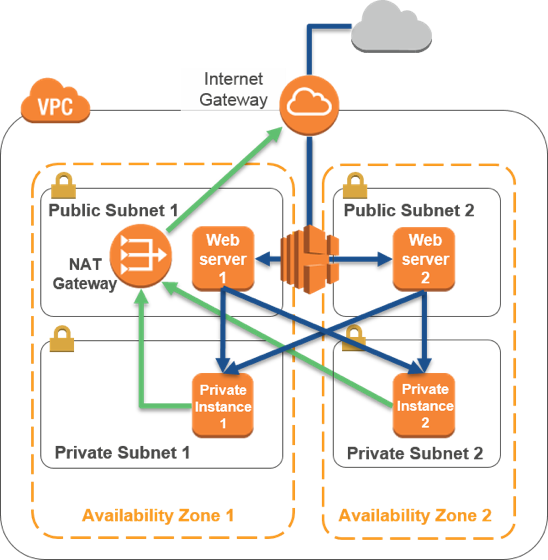
Lab 2 - Making Your Environment Highly Available

**Overview**  
In this lab, you will start with an Amazon EC2 instance inside of a public subnet (similar to what you created in the last lab), convert it into a basic PHP server, make your environment highly available, and add a highly available private tier that sits behind an AWS NAT Gateway.

**Objectives**

After completing **this** lab, you will be able to:

* Create an image of an existing Amazon EC2 instance and use it to launch a new instance.
* Create an Amazon ELB load balancer and attach it to Amazon EC2 instances.
* Create an AWS NAT Gateway.
* Create private subnets and launch Amazon EC2 instances into them.
* Edit private subnet route tables and security groups to intelligently control access.
* Test an AWS NAT Gateway.
* The final product of your lab will be this:



**Prerequisites**

This lab requires:

* Access to a notebook computer with Wi-Fi running Microsoft Windows, Mac OS X, or Linux (Ubuntu, SuSE, or Red Hat)
  + The qwikLABS lab environment is not accessible using an iPad or tablet device, but you can use these devices to access the student guide.
* For Microsoft Windows users: Administrator access to the computer
* An Internet browser such as Chrome, Firefox, or IE9 (previous versions of Internet Explorer are not supported)
* An SSH client such as PuTTY

**Duration**

The lab will require approximately **60 minutes** to complete.

Accessing the AWS Management Console

1. To the right of the lab title, click **Start Lab** to launch your Qwiklabs.

C:\Users\student\Desktop\architect\Architecting on AWS - Lab 2 - Making Your Environment Highly Available _ Qwiklabs + globalknowledge_files\start.png

1. On the **Connect** tab of the Qwiklabs page, copy the **Password** to the clipboard and then click **Open Console**.

C:\Users\student\Desktop\architect\Architecting on AWS - Lab 2 - Making Your Environment Highly Available _ Qwiklabs + globalknowledge_files\open.png

1. Sign in to the AWS Management Console using the following steps:
   * For **User Name**, type **awsstudent**
   * For **Password**, paste the password copied from the clipboard.
   * Click **Sign In**.

Task 1: Inspect Your environment

Unlike the last lab, this lab provides you with an environment already containing a few resources. Via CloudFormation, we’ve already launched and configured these things for you:

* A VPC (LabVPC) with a public subnet (PublicSubnet1)
* An Internet gateway attached to PublicSubnet1
* An Amazon EC2 instance (Web Server 1) inside of PublicSubnet1

**Task 1.1: Inspect Your VPC**

In this part of the lab, you will be guided through the VPC to see the various components that have already been set up.

1. On the **AWS Management Console** , on the **Services** menu, click **VPC**.
2. In the **VPC dashboard** , you can see what has already been set up for you. Take a look at the VPCs, Subnets, Route Tables, Internet gateways, Network ACLs, and Security Groups.

Confirm that you are seeing the stack create complete. This would ensure you are in the right region and all resources are created with any new AWS account, as well as the resources created via CloudFormation for the purposes of this lab.

1. In the navigation pane, click **Your VPCs**.
2. Here you can see the VPC that has been created for you via CloudFormation (LabVPC).

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 to 10.200.15.255 (with some reserved and unusable).

It is also attached to a route table, and includes a Network ACL.

This VPC also has a default tenancy value, which means unless otherwise specified, instances launched into this VPC will use shared tenancy hardware.

You can also see that this lab environment includes a default VPC. We **do not recommend** that you delete your default VPC from your own accounts, as having them restored is not a simple process.

1. In the navigation pane, click **Subnets**.
2. Here you can see the subnet created by this lab's CloudFormation template (PublicSubnet1).

In the **VPC** column, you can see that this subnet exists inside of **LabVPC**.

In the **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 AZ this subnet was placed inside of. It will vary based on the region in which your instructor launched your lab environments.

1. Click the check box next to **PublicSubnet1**'s name to reveal more details about it at the bottom of the page.
2. Click the **Route Table** tab.
3. Here you can see details about this subnet's routing.

The first entry specifies that traffic destined within the VPC's CIDR (**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 created for your lab. This is what makes this a public subnet rather than a private one.

1. Click the **Network ACL** tab.
2. Here you can see the network ACL that has been associated with this subnet.

Even though by default the inbound and outbound rules for your network ACL are open to all traffic, you will be using security groups to control access to your instances.

1. Click the **Tags** tab. Here you can see the single tag for this subnet that specifies its name as **PublicSubnet1**.
2. In the navigation pane, click **Internet Gateways**. Here you can inspect the details for the Internet gateway that was created for this lab environment.
3. In the navigation pane, click **Security Groups**. There are three security groups, two of which were created by default for you by AWS, and one (with the name beginning with **qls**) created by CloudFormation.
4. Click the security group with the group name that begins with **qls**. This is the security group that **Web Server 1** belongs to.
5. Click the **Inbound Rules** tab. Here you can see that this security group only allows traffic via SSH (TCP port 22) and HTTP (TCP port 80).
6. Click the **Outbound Rules** tab. Here you can see that this security group allows all outbound traffic via TCP and UDP.

**Task 1.2: Inspect Your Amazon EC2 Instance**

In this part of the lab, you will inspect the Amazon EC2 instance that was already launched for you via CloudFormation.

1. On the **Services** menu, click **EC2**.
2. In the **EC2 Dashboard** , you can see you already have 1 running instance, 1 storage volume, 1 key pair, and 3 security groups available (these are the same security groups you inspected in the last task).
3. In the navigation pane, click **Instances**.
4. Here you can see **Web Server 1** is already running. If it is not already selected, click the box next to its name to select it.
5. In the **Description** tab, you can inspect the details of this instance, including its public and private IPs and which AZ, VPC, subnet, and security group(s) it has been attached to.
6. Copy and paste the **Public IP** of this instance into a text editor, such as Notepad.
7. In the **Actions** menu, click **Instance Settings** > **View/Change User Data**.

Note that no user data appears. This means that while your instance was launched for you, it has not yet been configured to run your server's application. When launching an Amazon EC2 instance, you can specify user data which would do this for you (as was done in the last lab), but for the purposes of this lab we will walk you through this process in the next task.

Task 2: Start Your Web Server's PHP App

Even though your Amazon EC2 instance has already been launched for you, it is not yet running your web application. To start the web application, you will need to SSH into the instance and run a few basic commands.

*While in a production environment, you would want to have the application automatically configured upon launch of a new instance, in a testing or development environment there may be circumstances where a more manual or dynamic control over what the instance launches and how it is configured is necessary.*

**Task 2.1: Connect to Web Server 1 (Windows Only)**

This section is for Windows users only. If you are running OSX or Linux, skip to **Task 2.2**.

In this section of the lab, you will download your keypair and use it to connect to your Amazon EC2 instance with PuTTY.

1. From the qwikLABS page in your browser, in the **Connect** section, click **Download PEM/PPK** > **Download PPK**.
2. Save the file to your **\Downloads** folder or any other easy to access location on your local computer.
3. Launch **PuTTY** by running the putty.exe file you downloaded in the last lab.
4. For **Host Name** , enter the public IP address from your **Web Server 1** instance which you copied into a text editor earlier in the lab.
5. In the **Connection** list, expand **SSH**.
6. Click **Auth**.
7. In the **Private key file for authentication** box, browse to the .ppk file that you downloaded earlier, then click Open.
8. In the **PuTTY Security Alert** dialog box that opens, click **Yes** to add the key to PuTTY's cache.

For **login as:** type and press **Enter**. You are now logged in to your **Web Server** instance.

**Note** The actual text you see may differ slightly from the above

1. Skip the next subtask and proceed to **Task 2.3**.

**Task 2.2: Connect to Web Server 1 (Mac OSX/Linux only)**

This section is for **Linux** and **Mac OSX** users only. If you are running **Windows** but have not yet connected to your instance, go back to **Task 2.1**. If you have already connected to your instance, skip ahead to **Task 2.3**.

In this section of the lab, you will download your keypair and use it to connect to your Amazon EC2 instance.

1. From the qwikLABS page in your browser, in the **Connect** section, click **Download PEM/PPK** > **Download PEM**.
2. Save the file to your local computer in a place where you can easily access it.
3. To connect to your EC2 instance, run the following commands in Terminal:

Copy Code Block

chmod 400 <path and name of pem>

ssh –i <path and name of pem> ec2-user@<Public IP>

For **<path and name of pem>**, substitute the path/filename to the .pem file you downloaded.

For **<Public IP>**, substitute the public IP address for your **Web Server 1** instance which you copied into a text editor earlier in the lab.

**Task 2.3: Download, Install, and Launch Your Web Server's PHP Application**

In this section of the lab, you will run a series of Linux commands on your instance which will download, install, and launch your server's PHP application. We will step you through each command one at a time so you can understand exactly what you are doing to accomplish this task.

1. To update your instance, execute the following command.

Copy Code Block

sudo yum -y update

This will run through a check of what updates are available for your instance, download the updates, and install them.

1. To install a package that creates a web server, execute the following command:

Copy Code Block

sudo yum -y install httpd php

This command installs an Apache web server and the PHP interpreter.

1. Execute the following command:

Copy Code Block

sudo chkconfig httpd on

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

1. Execute the following command

Copy Code Block

wget https://s3-us-west-2.amazonaws.com/us-west-2-aws-staging/awsu-ilt/academy-cca/v3.0/labs/lab8-ha/scripts/phpapp.zip

This downloads the sample PHP application into the current directory.

1. Execute the following command:

Copy Code Block

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

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

1. Execute the following command:

Copy Code Block

sudo service httpd start

This starts the Apache web server.

1. Open a new web browser or tab, paste the **Public IP** address for your instance in the address bar and hit **Enter**.

The sample PHP application is run and the information specific to your Amazon EC2 instance is displayed.

1. Close the web browser or window that you opened in the previous step.
2. Return to your SSH session, type

Copy Code Block

exit

and then press **Enter**. This ends your SSH session.

Task 3: Launch a Duplicate of Web Server 1 into a Second Availability Zone

To increase the availability of your application environment, in this task you will create a second web server in different Availability Zone.

To accomplish this, you will do the following:

* Create a snapshot of the storage volume for Web Server 1.
* Create an image (Amazon Machine Image, or AMI) based off of that snapshot.
* Create a second public subnet (PublicSubnet2) in a second AZ.
* Launch a new instance (Web Server 2) into PublicSubnet2 using the AMI you created in step 2.

**Scenario**

*Using your own custom AMIs to quickly provision new instances that are already configured for use is a key component to ensuring a consistent experience for users across your environment. It also allows you to easily replace unpatched instances with patched instances, without having to take your server down, patch it, and re-launch it, effectively eliminating downtime for patching and other maintenance.*

**Task 3.1: Create an Amazon Machine Image of Web Server 1**

In this subtask, you will create an Amazon Machine Image (AMI) of Web Server 1, so that it can be used to launch future instances that already have your web application running.

Typically, via the console, you can create an AMI by simple stopping the instance and creating the image from it. But in this circumstance, if your instance is using ephemeral storage and you stop it, you will lose the work you did earlier in the lab (installing and running your web server and simple PHP application).

But by instead snapshotting the storage volume of Web Server 1 and creating the AMI off of that, you can launch a new web server without stopping Web Server 1 and without having to run any commands you already ran earlier in this lab.

**Note**: When using Windows-based AMIs, snapshotting is a different process from Linux-based AMIs.

1. On the **Services** menu, click **EC2**.
2. In the navigation pane, click **Volumes**.
3. Ensure the storage volume attached to **Web Server 1** is already selected, and click **Actions** > **Create Snapshot**.
4. In the **Create Snapshot** window that appears, specify the following settings:
   * For **Name**, type 
   * For **Description**, type 
5. In the **Create Snapshot** window, click **Create**.
6. Click the link for your new snapshot next to **View snapshot**.
7. Within a minute, you should see your snapshot appear in the list. If it doesn't appear right away, periodically refresh the list with the refresh button in the upper right corner of the page until it appears.
8. Periodically refresh the list with the refresh button in the upper right corner of the page until **Status** value for the snapshot is *completed*. It may take a few minutes
9. With the WebServer1Snapshot selected, click **Actions** > **Create Image**.
10. In the **Create Image from EBS Snapshot** window that appears, specify the following settings:
    * For **Name**, type 
    * For **Description**, type 
    * For **Virtualization type**, click **Hardware-assisted virtualization** **Note** If when performing this operation outside of this operation, you're unsure which to use, we recommend using Hardware-assisted virtualization (HVM) rather than paravirtual (PV) machines as HVM provide the best performance in the widest range of scenarios.
11. In the **Create Image from EBS Snapshot** dialog box, click **Create**, then click **Close**.

You've now started the process to create the AMI you will use to launch your second web server. While you wait for that process to complete, in the next task you will create the new public subnet where the new instance will be launched.

**Task 3.2: Create PublicSubnet2**

In this subtask, you will create PublicSubnet2, which will exist within LabVPC, but in a different Availability Zone from PublicSubnet1, to allow for cross-zone availability. You will then make the subnet publicly available by attaching it to a route table.

1. On the **Services** menu, click **VPC**.
2. In the navigation pane, click **Subnets**.
3. In the row for **PublicSubnet1** , note the value for **Availability Zone**.
4. Click **Create Subnet**.
5. Specify the following details:
   * For **Name tag**, type 
   * For **VPC**, type 
   * For **Availability Zone**, choose a different Availability Zone from the one where PublicSubnet1 resides.
   * For **CIDR block**, type This will create a second subnet in a different Availability Zone, but still within **LabVPC** , with an IP range between 10.200.1.0 and 10.200.1.255.
6. Click **Yes, Create**. In a few seconds, **PublicSubnet2** will appear in your subnet list.
7. With **PublicSubnet2** selected, click **Subnet Actions** > **Modify auto assign IP settings**.
8. In the **Modify auto assign IP settings** dialog box that appears, select the check box for **Enable auto-assign public IPv4 address**.
9. Click **Save**. For instances launched into this subnet, they will now be assigned a public IP by default, unless specified otherwise during launch. Click **Close**.
10. With **PublicSubnet2** still selected, click the **Route Table** tab.
11. 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.
12. Click **Edit**.
13. In the **Change to:** drop-down list, click the entry that includes **Public** in the listing.
14. Click **Save**.

**PublicSubnet2** is now publicly available.

**Task 3.3: Launch Your Second Web Server**

In this subtask, you will use your previously created AMI to launch a new instance into PublicSubnet2.

1. On the **Services** menu, click **EC2**.
2. In the navigation pane, click **AMIs**.
3. With **WebServerImage** already selected, click **Launch**.
4. If it is not already selected, select **t2.micro** as your instance type.
5. Click **Next: Configure Instance Details**.
6. Specify the following settings if they are not already specified by default:
   * For **Network**, type 
   * For **Subnet**, type 
   * For **Auto-assign Public IP**, click **Use subnet setting (Enable)**.
7. Leave the remaining settings as their default and click **Next: Add Storage**.
8. Leave the storage settings as their default and click **Next: Add Tags**.
9. Click **click to add a Name tag**.
10. In the **Value** column for **Name** , type .
11. Click **Next: Configure Security Group**.
12. Select **Select an existing security group**.
13. Select the check box next to the security group whose name starts with **qls**.

**Note** This is the same security group that **Web Server 1** resides in.

1. Click **Review and Launch**.
2. Review that your settings are as they should be according to the steps above and click **Launch**.
3. When prompted, ensure that the default **qwikLABS** key provided for this lab environment is selected for **Select a key pair** , and select the acknowledgement check box.

**Note** You need to use a *different* key pair from the one you downloaded in the first lab.

1. Click **Launch Instances**.
2. Click **View Instances**.
3. Wait until the instance state of **Web Server 2** changes to *running*. This may take a few minutes. You can click the refresh button in the upper right corner to refresh the status of your instances. Once the instance state changes to *running*,proceed to the next task.

Task 4: Launch and Attach a Load Balancer

While you now have two separate web server instances, your environment is not truly highly available as those instances are only accessible via two separate public IPs. In addition to being a confusing experience for users, traffic will not be balanced across the instances automatically, which could result in one instance being overutilized while the other is empty.

**Task 4.1: Create and attach a Load Balancer with Amazon ELB**

In this subtask, you will use Amazon ELB to create a load balancer and attach it to your two web servers. Then you will test the load balancer to show that it is distributing load between them.

1. If you are not already in the **EC2 Dashboard** , on the **Services** menu, click **EC2**.
2. In the navigation pane, click **Load Balancers**.
3. Click **Create Load Balancer**, click **Classic Load Balancer**, then click **Continue**.
4. Specify the following settings:
   * For **Load Balancer name**, type 
   * For **Create LB Inside**, type 
5. When you select **LabVPC** , you should see the **Select Subnets** menu appear at the bottom of the page.
6. In the **Select subnets** menu, click the plus symbols in the rows for both subnets ( **PublicSubnet1** and **PublicSubnet2** ).
7. With both subnets now listed under **Selected Subnets** , click **Next: Assign Security Groups**.
8. Select **Create a new security group**.
9. Specify the following settings:
   * For **Security group name**, type 
   * For **Description**, type\**Security group for the load balancer for the web servers*\*
   * For **Type**, click **HTTP**.
   * For **Source**, click **Anywhere**.
10. Click **Next: Configure Security Settings**.
11. Since traffic to this public-facing load balancer does not need to be secure, the load balancer does not need a secure listener. Click **Next: Configure Health Check**.
12. Specify the following settings:
    * For **Ping Path**, type **/index.php** This tells the load balancer to ping index.php when it does a health check on an instance.
    * For **Response Timeout**, type This tells the load balancer to wait 5 seconds for a response to its health check before timing out.
    * For **Interval**, type\**10 seconds*\*. This tells the load balancer to wait 10 seconds between health checks.
    * For **Unhealthy threshhold**, type\**2*\* This tells the load balancer that 2 consecutive failed health checks are required to identify the instance as unhealthy and stop sending traffic to it until it is healthy again.
    * For **Healthy threshhold**, type This tells the load balancer that 5 consecutive successful health checks are required to identify the instance as healthy and start sending traffic to it.
13. Click **Next: Add EC2 Instances**.
14. Select the check boxes next to both of your web server instances ( **Web Server 1** and **Web Server 2** ).
15. Leave the remaining settings as their default and click **Next: Add Tags**.
16. Create a tag with a following settings:
    * For **Key**, type 
    * For **Value**, type 
17. Click **Review and Create**.
18. Confirm that your load balancer's settings are as they should be based on the above steps and click **Create**.
19. Click **Close**.
20. Your new load balancer has already begun performing health checks on the web servers it is attached to. After a minute has passed, start periodically refreshing the page with the refresh button in the upper right corner of the page until its **Status** is *2 of 2 instances in service*. It should take no more than 2 minutes total.
21. Copy the value next to **DNS Name** , paste it into a new browser tab or window, and hit **Enter** to navigate to your load balancer's public DNS address.
22. The landing page for your PHP app should appear. Note the IP address that appears on the page.
23. Refresh the page. The IP address should change. If it doesn't change, hit refresh again until it does. This shows that your load balancer is successfully balancing traffic between your two instances in your two different subnets.

You have now completed the process of launching your web server into a second AZ and putting your servers behind a load balancer in order to make your application more highly available. In the next Task, you will create a private application environment that sits behind an AWS NAT Gateway.

Task 5: Creating a Private Application Environment

So far, your environments have only used public subnets, however most application environments use private subnets as well, where instances are cut off from direct Internet access. In this task, you will create a simple private application environment that routes traffic from two private instances out through an AWS NAT gateway.

**Scenario**  
*Most application environments do not need all of their resources directly accessible via the Internet. Applications are instead configured to control access to certain resources by keeping them in a private layer. This is typical for applications that have sensitive data that should only be accessible via approved applications in approved ways.*

Task 5.1: Create Two Private Subnets

In this subtask, you will create two private subnets within LabVPC.

1. On the **Services** menu, click **VPC**.
2. In the navigation pane, click **Subnets**.
3. Click **Create Subnet**.
4. Specify the following settings:
   * For **Name tag**, type 
   * For **VPC**, type 
   * For **Availability Zone**, select any AZ, but remember which AZ you picked.
   * For **CIDR block**, type **10.200.2.0/23**
5. Click **Yes, Create**.

Your first private subnet has now been created.

1. Click **Create Subnet**.
2. Specify the following settings:
   * For **Name tag**, type 
   * For **VPC**, type 
   * For **Availability Zone**, select any AZ other than the one you picked for Private Subnet 1
   * For **CIDR block**, type **10.200.4.0/23**
3. Click **Yes, Create**.

You now should have two subnets (without routes to an Internet gateway), each in a different Availability Zone.

**Task 5.2: Create and Attach an AWS NAT Gateway**

In this subtask, you will create an AWS NAT gateway and route your subnets through it.

1. In the navigation pane, click **NAT Gateways**.
2. Click **Create NAT Gateway**.
3. Click inside of the text box for **Subnet**. A list of your subnets will appear.
4. Click on the entry for **PublicSubnet1**.

**Note** While you are required to place an AWS NAT gateway inside of one subnet, it is accessible by any instances within your VPC, regardless of which subnet they reside in. AWS NAT gateways are also fully managed, which means they are inherently highly available, and do not need to be placed in more than one Availability Zone.

1. An AWS NAT gateway requires an Elastic IP so that in the event that the underlying hardware fails, the replacement hardware can attach to the same IP without an interruption in service.

Click **Create New EIP**.

1. Click **Create a NAT Gateway**.
2. You must now route your private subnets to the NAT gateway for it to work. Click **Edit Route Tables**.
3. Click **Create Route Table**.
4. Specify the following settings:
   * For **Name tag**, type 
   * For **VPC**, type 
5. Click **Yes, Create**.
6. With your **Private** route table selected, click the **Routes** tab.
7. This table already includes a local route, but it needs a route to the NAT gateway for traffic headed to the Internet.

Click **Edit**.

1. Click **Add another route**.
2. Under **Destination** , type **0.0.0.0/0**
3. Click the text box under **Target**. A list appears. In that list, click the entry that starts with *nat-*.
4. Click **Save**.
5. Click the **Subnet Associations** tab.
6. Click **Edit**.
7. Select the check boxes for **PrivateSubnet1** and **PrivateSubnet2**.
8. Click **Save**.

Your private subnets will now route traffic bound for the Internet through your AWS NAT gateway.

**Task 5.3: Launch Two Instances into Your Private Subnets**

In this subtask, you will launch two Amazon EC2 instances into your private subnets so you can test your AWS NAT gateway.

1. On the **Services** menu, click **EC2**.
2. Click **Launch Instance**.
3. Click **Select** next to the entry for **Amazon Linux AMI** at the top of the list.
4. Click **Next:Configure Instance Details** to use a t2.micro instance type.
5. Specify the following settings:
   * For **Network**, type 
   * For **Subnet**, type 
   * For **Auto-assign Public IP**, click **Use subnet setting (Disable)**.
6. Leave the remaining settings as their default and click **Next: Add Storage**.
7. Leave these settings as their default and click **Next: Add Tags**.
8. Click **click to add a Name tag**.
9. For **Value** , type .
10. Click **Next: Configure Security Group**.
11. Select **Create a new security group**.
12. Specify the following settings:
    * For **Security group name**, type 
    * For **Description**, type 
13. For **Source** , click **Custom**.
14. In the text box next to **Custom** for your SSH rule, type . A list of your security groups will appear.
15. Click the security group that includes **WebServerSG** in the name.

**Note** This means that in order to access instances in your **PrivateSG** security group, you will first need to SSH into one of the instances in your **WebServerSG** security group and then SSH from there into a private instance.

1. Click **Review and Launch**.
2. Ensure your instance's settings are correct and click **Launch**.
3. Ensure that your qwikLABS key pair is selected, select the acknowledgement check box, and click **Launch Instances**.
4. Click **View Instances**.
5. Follow steps **145-160** to create a second private instance, this time named **Private Instance 2** , and located in PrivateSubnet2. (You do not need to create a new security group, you can just use the same security group you created for Private Instance 1)
6. You should now have two private instances, each one in a different private subnet.
7. Click the check box next to **Private Instance 1**.
8. In the **Description** tab, find the entry for **Private IPs** , copy it, and paste it into the same text file on a new line.
9. Once **Private Instance 2** is in the *running\_state*, click the check box next to its name and copy its private IP into the same text file for easy retrieval.

**Task 5.4: SSH into Web Server 1 again (Windows Only)**

**Note** This section is for Windows users only. If you are running Mac OSX or Linux skip ahead to Task 5.5.

**In this section of the lab, connect to your Amazon EC2 instance via PuTTY again.**

1. If you are not already there, navigate to your list of Amazon EC2 instances in the **EC2 Dashboard**.
2. Click the check box next to **Web Server 1**.
3. In the **Description** tab, find the entry for **Public IP** , copy it, and paste it into a text editor, such as Notepad.
4. Download **Pageant** from:

<http://the.earth.li/~sgtatham/putty/latest/x86/pageant.exe>

1. Launch **Pageant**.

If Pageant doesn't appear, find the icon for it in your task bar (a computer terminal with a hat-like object on top) and double-click it to open it.

1. Click **Add Key**.
2. Select the .ppk file you downloaded earlier in the lab, click **OK** , and close the Pageant window.

Since you will need to SSH into your private instances from your public instances, This will allow you to forward the key pair necessary for authenticating with your private instances.

1. Launch **PuTTY** by running the putty.exe file you downloaded previously.
2. In the Host Name box, enter the public IP address from your **Web Server 1** instance which you copied into a text editor earlier in the lab.
3. In the **Connection** list, expand **SSH**.
4. Click **Auth**.
5. Select **Allow agent forwarding**.
6. In the **Private key file for authentication box** , browse to the .ppk file that you downloaded earlier, then click **Open**.
7. For **Log in as** : type and press Enter.
8. Skip the next subtask and proceed to **Task 5.6**.

**Task 5.5: Connect to Web Server 1 again (Mac OSX/Linux only)**

**Note** This section is for **Linux** and **Mac OSX** users only. If you are running **Windows** but have not yet connected to your instance, go back to **Task 5.4**. If you have already connected to your instance, skip ahead to **Task 5.6**.

**In this section of the lab, you will connect to your Amazon EC2 instance.**

1. To connect to **Web Server 1** with SSH agent forwarding enabled, run the following commands in Terminal:  
   

**MacOS only:**  


**Linux only:**  


**All:**  
  
For **<path and name of pem>**, substitute the path/filename to the .pem file you downloaded earlier.

For **<Public IP>**, substitute the public IP address you copied earlier.

**Task 5.6: Test Your NAT Gateway**

**In this subtask, you will SSH from your Web Service 1 instance into your private instances, where you will test your NAT gateway by pinging a website.**

1. Return to your **EC2 Dashboard**.
2. Click the check box next to **Private Instance 1** to select it.
3. In the **Description** tab, find the entry for **Private IPs** , copy it, and paste it into a text file.
4. Return to your open SSH connection and run the following command:

Copy Code Block

ssh ec2-user@<Private IP>

For **<Private IP>** ,substitute the private IP address you copied in the previous step.

1. When prompted for a response, type and press **Enter**.

You should be logged in to your private instance now.

1. To verify that your private instance can connect to the Internet, run the following command:

Copy Code Block

ping ietf.org

1. You should receive a series of continuous responses that look similar to this:

Copy Code Block

PING ietf.org (4.31.198.44) 56(84) bytes of data.

64 bytes from mail.ietf.org (4.31.198.44): icmp\_seq=1 ttl=48 time=74.9 ms

If you don't receive the expected response, check with your instructor to determine the problem.

**Note** This test will only work on websites which have ICMP enabled, so websites besides ietf.org may not work.

1. Once you've received the expected response, press **Ctrl+C** to end the ping command.
2. To end your SSH session with **Private Instance 1** , type and press **Enter**.
3. Return to your **EC2 Dashboard**.
4. Click the check box next to **Private Instance 2** to select it.
5. In the **Description** tab, find the entry for **Private IPs** , copy it, and paste it into a text file.
6. Follow steps **186-191** to test the Internet connection for **Private Instance 2**.

Lab Complete

Congratulations! You have successfully completed the lab. To clean up your lab environment, do the following:

1. To sign out of the **AWS Management Console** click **awsstudent** in the navigation bar, and then click **Sign Out**.
2. Return to the **qwikLABS** page where you launched your lab and click **End**.

© for the use of: shweta\_misra@fanniemae.com

* Connect
* Lab Details
* Addl. Info
* **Lab Connection**   
  Please follow the lab instructions to connect to your lab

**Warning:** Do not transmit data into the AWS Console that is not related to Qwiklabs or the lab you are taking.

* + **AWS Console Details**
  + **Access Key Details**
  + [Show Access Keys](https://globalknowledge.qwiklab.com/focuses/17008#accessKeyIdModal)
  + **Key Pair Details**

|  |  |
| --- | --- |
| **Setup Time (min.)** | 3 |
| **Duration (min.)** | 150 |
| **Access (min.)** | 150 |

|  |
| --- |
| **Tags:** [**AWS-100-ARC-51-lab-2-ha**](https://globalknowledge.qwiklab.com/tags/AWS-100-ARC-51-lab-2-ha)  **Lab Description:** In this lab, you will take the single web server environment from your last lab and make it more highly available using another AZ, a NAT gateway, and an Elastic Load Balancer.  **Lab Creator:** qwikLABS publisher  **Date Created:** May 17, 2017 08:25  **AWS Region::** [us-west-2] **US West (Oregon)** |

Additional Lab Information:

×

**Access Key Details**

Use access keys to make secure REST or Query protocol requests to any AWS service API.

Close