Architecting on AWS
Lab Guide

Version 3.1

100-ARC-31-EN-LG

Corrections or feedback on the course, please email us at:
aws-course-feedback@amazon.com

For all other questions, please email us at:
aws-training-info@amazon.com

Table of Contents

Lab 1: Creating Your First Amazon Virtual Private Cloud 1
Login to the AWS Management Console 2
Creating the Base VPC 4
Creating a VPC Security Group 7
Launching a Web Server 7
Manually Creating Two More Subnets 10
Launching a Bastion Windows Host 15
Launching a Back-End Microsoft SQL Server 17
Connecting to the Bastion Host 20
Logging in to the Database Server 23
Additional Task (Optional) 24
Conclusion 25
Ending the Lab 25

Lab 2: Working with Amazon Identity and Access Management 26
Login to the AWS Management Console 26
Creating IAM users and policies 28
Testing IAM users 32
Conclusion 34

Lab 3: Getting Started with Auto Scaling 36
Login to the AWS Management Console 36
Overview of Auto Scaling 39
Creating an Elastic Load Balancer 40
Creating a Launch Configuration 41
Creating an Auto Scaling Group 42
Verifying and Testing Auto Scaling 43
Adding Auto Scaling Notifications 44
Creating Auto Scaling Policies 46
Testing Auto Scaling 48
Conclusion 51

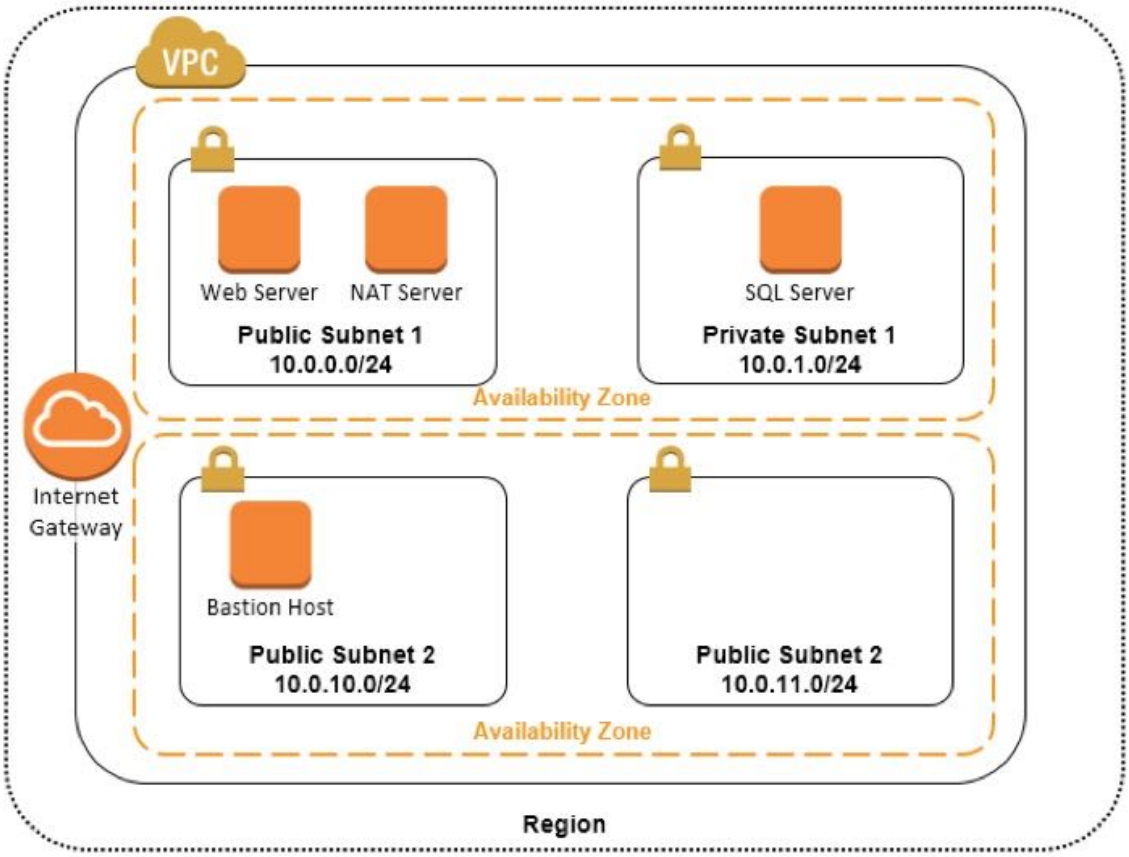
Lab 4: Creating a Batch Processing Cluster 52
Login to the AWS Management Console 53
Creating an IAM role 55
Creating a 'Master' EC2 Instance 56
Connecting to your instance 58
Creating an AMI from your batch processing instance 60
Creating two SQS task queues 61
Creating an S3 bucket 62
Launching worker nodes 62
Dispatching work and viewing results 65
Monitoring the cluster 66
Conclusion 69

Lab 1: Creating Your First Amazon Virtual Private Cloud

Overview

In this lab session, you will create a basic Amazon Virtual Private Cloud (VPC) and extend it to produce a customized network. You will do this with the AWS Management Console.

You will build the following architecture:



The overall VPC is designed to incorporate several basic features:

- **It spans two Availability Zones (AZs)** so you can distribute applications across these zones to architect for application durability and availability.
- **Within each Availability Zone (AZ) there are two subnets.** The **Public subnets** can route directly to the Internet. The **Private subnets** are able to communicate with any other subnet within the VPC, but there is no direct access between private subnets and the Internet.

What is Amazon Virtual Private Cloud?

Amazon Virtual Private Cloud (Amazon VPC) lets you provision a logically isolated section of the AWS Cloud where you can launch AWS resources in a virtual network that you define. You

have complete control over your virtual networking environment, including selection of your own IP address range, creation of subnets, and configuration of route tables and network gateways.

Topics Covered

This lab will take you through the creation of a VPC, including:

- Creating a Virtual Private Cloud (VPC) using the VPC Wizard
- Configuring Security Groups
- Launching EC2 instances
- Connecting to EC2 instances

Login to the AWS Management Console

Using qwikLABS to login to the AWS Management Console

Welcome to this self-paced lab! The first step is for you to login to Amazon Web Services.

1. To the right of the lab title, click the **Start Lab** button to launch your *qwik*LABS. If you are prompted for a token, use the one distributed to you (or the token you purchased).

Note: A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.

2. On the lab details page, notice the lab properties:

   a. **Duration -** The time the lab will run before automatically shutting down.

   b. **Setup Time -** The estimated time to set up the lab environment.

   c. **AWS Region** - The AWS Region in which the lab resources are created.

Note: The AWS Region for your lab may differ depending on your location and the lab setup.

3. Click the **Download PEM/PPK** button and then **Download PEM**. A file will download to your computer for use later in the lab, so remember where it was saved (eg Downloads folder):

4. In the AWS Management Console section of the *qwik*LABS page, **copy the Password to the clipboard**.

5. Click the **Open Console** button.

6. Log into the AWS Management Console using the following steps.

   a. In the **User Name** field type: `awsstudent`

   b. In the **Password** field, paste the password copied from the lab details page.

   c. Click **Sign In**:

The AWS account is automatically generated by *qwik*LABS. The login credentials for the `awsstudent` account are provisioned by *qwik*LABS using AWS Identity Access Management.

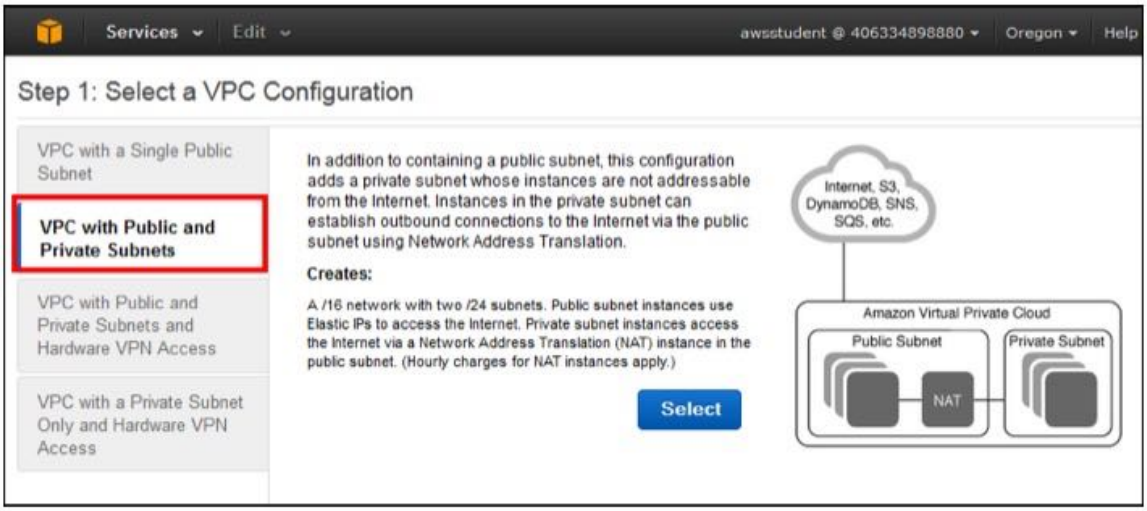
Creating the Base VPC

You will first create the VPC by using the VPC Wizard.

1. When you are logged into the console, click **VPC**.

2. Click the **Start VPC Wizard** button.

3. Click the **VPC with Public and Private Subnets** option:



4. Click **Select**.

The **VPC with Public and Private Subnets** panel contains several parameters. Depending on your professional background, the notation may appear different to that with which you are familiar. The VPC uses **CIDR block notation**, such as **`10.0.1.0/24`** which can also be expressed as **`10.0.1.0`** with a subnet mask of **`255.255.255.0`**.

The VPC itself is a Class B network in the **`10.0.0.0`** space. If you are familiar with the IPv4 address space, you will recognize this as one of the non-routable address blocks. The overall address space uses an IP CIDR block of **`10.0.0.0/16`**, which is the equivalent of a subnet mask of **`255.255.0.0`** (a full Class B network).

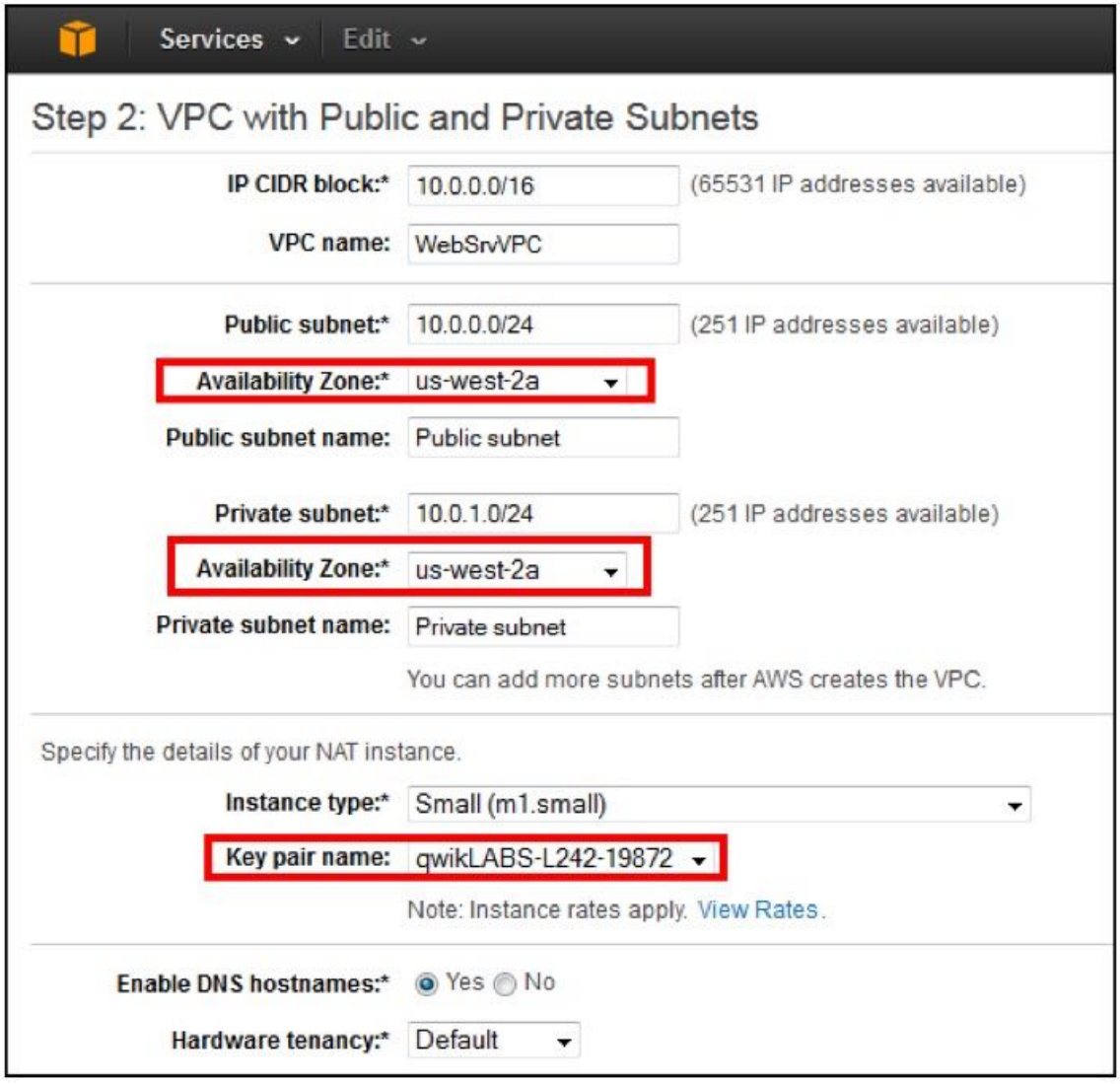
You will now edit the settings for the Public and Private subnets.

5. Choose an Amazon EC2 **Availability Zone** (for example, `us-west-2a`) from the drop down list under **Public subnet**.

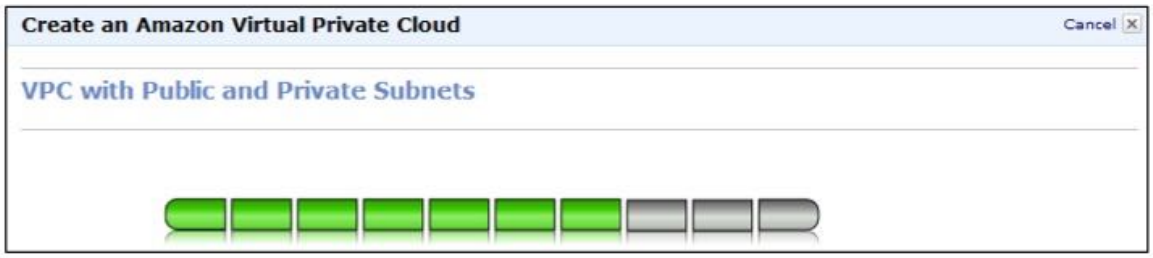
6. Choose the same **Availability Zone** for **Private subnet** as the prior step.

7. For the **Key pair name**, select the one with the same name as the PEM/PPK file that you downloaded earlier from the drop-down list. (The name should start with **qwikLABS**.)

Important: Be certain that the subnets are both in the same Amazon EC2 Availability Zone!

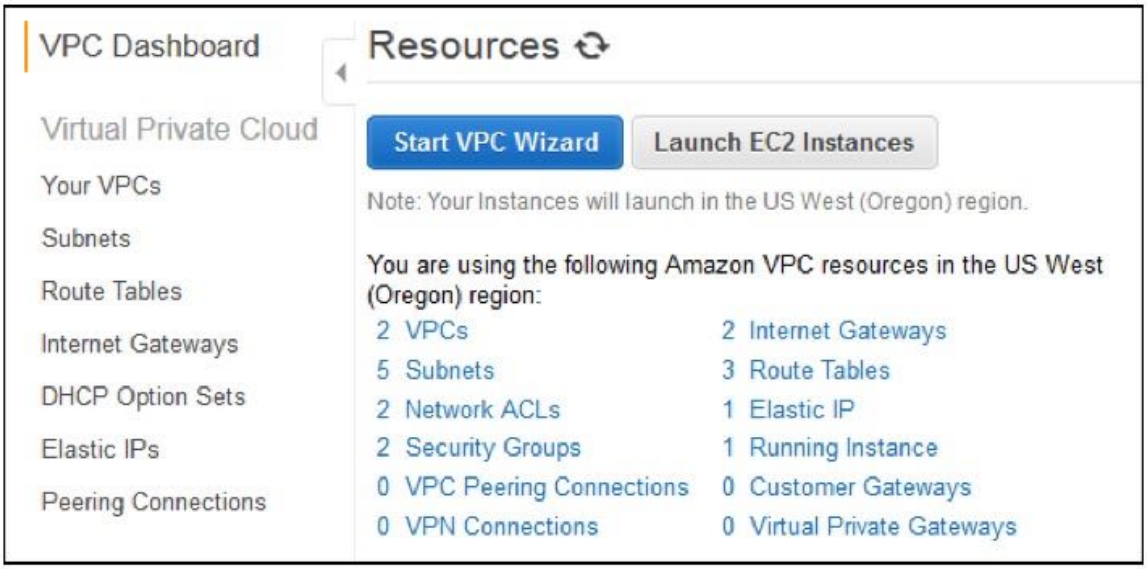


8. **Optional:** You can provide **VPC name** (e.g. WebSrvVPC).
9. Click the **Create VPC** button. This will open a dialog box indicating the progress of creating your VPC:



10. Once the VPC is created, click **OK**.

The VPC Dashboard displays a count of available network elements:



Your VPC has an important detail: everything is in a single Availability Zone. In order to optimize application availability you will need to distribute assets across multiple Availability Zones, which means that you need to add another pair of subnets. You will do that later in this lab.

Handling Outbound requests with NAT Servers

The VPC Wizard automatically launches a **NAT ("Network Address Translation") server**, which allows servers in the Private subnet to **initiate outbound connections to the Internet** to download software and access Internet services, such as Amazon S3. It does not allow systems on the Internet to initiate inbound connections to servers in the Private subnet. The VPC wizard assign a public **Elastic IP Address** to the NAT, which allows it to communicate with the Internet.

By default the NAT Server instance type is an `m1.small` and uses an EC2 Key Pair that was generated by *qwik*LABS, as was indicated in the VPC Wizard.

Creating a VPC Security Group

With the VPC created, your next task is to create a VPC security group that will permit access for web and SSH traffic.

1. Within the VPC Dashboard, click **Security Groups** in the left panel.
2. Click **Create Security Group** and then:
    a. **Name:** `Web`
    b. **Description:** `Web Security Group`
    c. **VPC:** Select the VPC you created.
    d. Click **Yes, Create**.

You should now see the new security group in the VPC Security Groups page. The security group can be edited to allow inbound HTTP and SSH traffic.

3. From the Details pane, click the **Inbound Rules** tab.
4. Click **Edit**.
5. From the **Type** list, select **HTTP**, and enter **0.0.0.0/0** for **Source**.
6. Click **Add another rule**.
7. From the **Type** list, select **SSH**, and enter **0.0.0.0/0** for **Source**.
8. Click **Save**.

Your security group is now ready.

Launching a Web Server

Your next task is to launch an EC2 instance and bootstrap it to act as a web server.

1. From the **Services** menu at the top of the screen, select **EC2.**
2. Click **Launch Instance**.
3. Locate the **Amazon Linux AMI** and click **Select**.
4. At the **Choose an Instance Type** panel, click **Next: Configure Instance Details**.
5. At the **Configure Instance Details** panel:
    a. Select the VPC you created from the drop-down list (`10.0.0.0/16`).
    b. Select your public subnet (`10.0.0.0/24`).
    c. Check the **Automatically assign a public IP address to your instances** check-box.

d. Expand the **Advanced Details** section. (You may need to scroll down to locate it.)

e. Enter the following text in the **User Data** field.
Note: Make sure the line-breaks appear in the correct positions. For your convenience, a **command reference text file** is attached to the *qwik*LABS page for this lab, which contains a text version of this script to simplify Copy & Paste.

```
#!/bin/sh
yum -y install httpd
chkconfig httpd on
/etc/init.d/httpd start
```

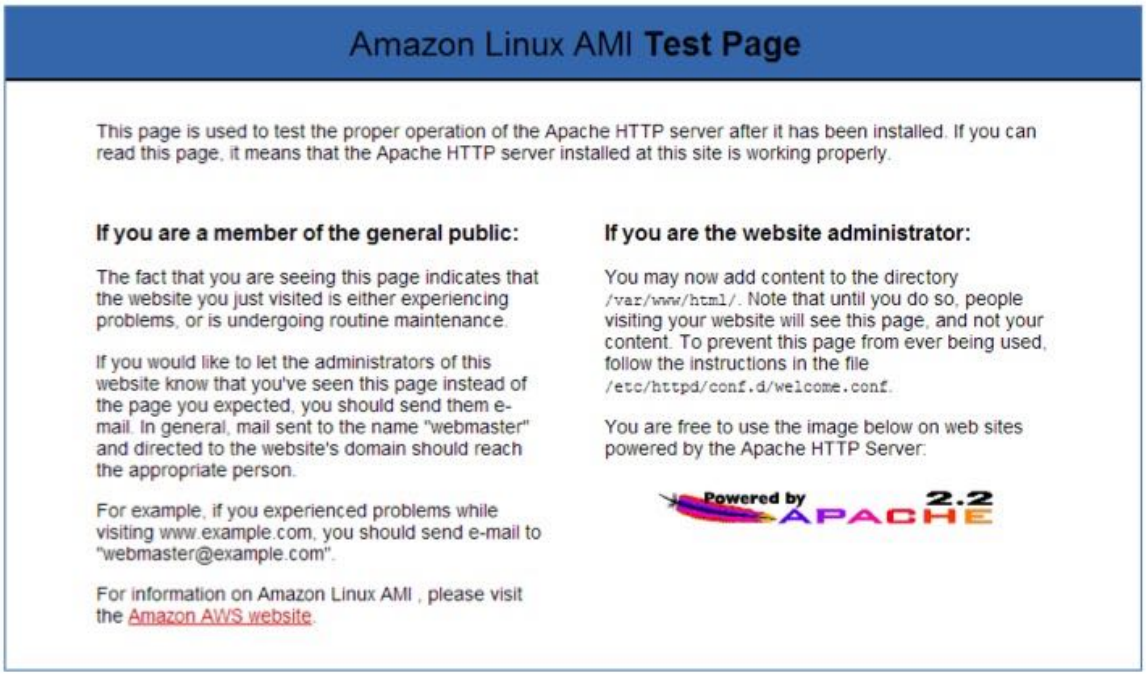
f. Click **Next: Add Storage**.

6. There are no modifications needed in the Add Storage panel. Click **Next: Tag Instance**.
7. At the **Tag Instance** panel, enter the **Value**: `Web Server 1`
8. Click **Next: Configure Security Group**.
9. Choose the **Select an existing security group** option.
10. Select the `Web` security group that you created.
11. Click **Review and Launch**.
12. Click **Launch**. You are presented with the **Select an existing key pair or create a new key pair** dialog.
13. Check the acknowledgement box and click **Launch Instances**.
14. Click **View Instances** (you might need to scroll down to see it).

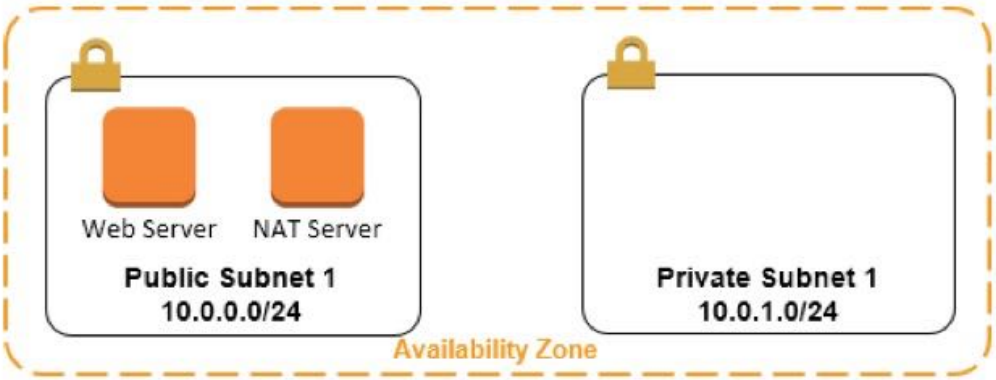
You should see two instances. The one labeled **`Web Server 1`**, is your web server. The other is the NAT instance that was launched by the VPC Wizard. Your web server is initially in the 'pending' state and will change to the 'running' state. After the instance is running, status checks are performed.

15. Wait until the web server shows `2/2 checks passed`.
16. Click on the web server and copy its **Public DNS** to your clipboard.

17. **Open a new tab** in your web browser, **paste the DNS address** and hit Enter. You should see something similar to the following:



The diagram below shows what you have configured thus far:



There is one other very important item missing from the environment: a second Availability Zone (AZ) with another web. AWS provides you access to multiple Availability Zones at no additional cost. A best practice is to mirror servers across two availability zones, and then use load balancing to distribute traffic between the availability zones.

AWS considers multi-AZ deployments essential to your welfare. AWS data centers are more reliable than typical Enterprise data centers, but outages can happen. If your environment is in a single AZ, you have no SLA protection. **The EC2 SLA is activated only if you are running instances in two or more Availbility Zones in an AWS Region that go offline at the same time.**

Manually Creating Two More Subnets

You will now create a public subnet and a private subnet in another Availability Zone. Unlike the previous subnets, you create these without the assistance of a wizard. They will be in the same Availability Zone as each other but in a different Availability Zone from the first two subnets you created. The original subnets were `10.0.0.0/24` (public), and `10.0.1.0/24` (private).

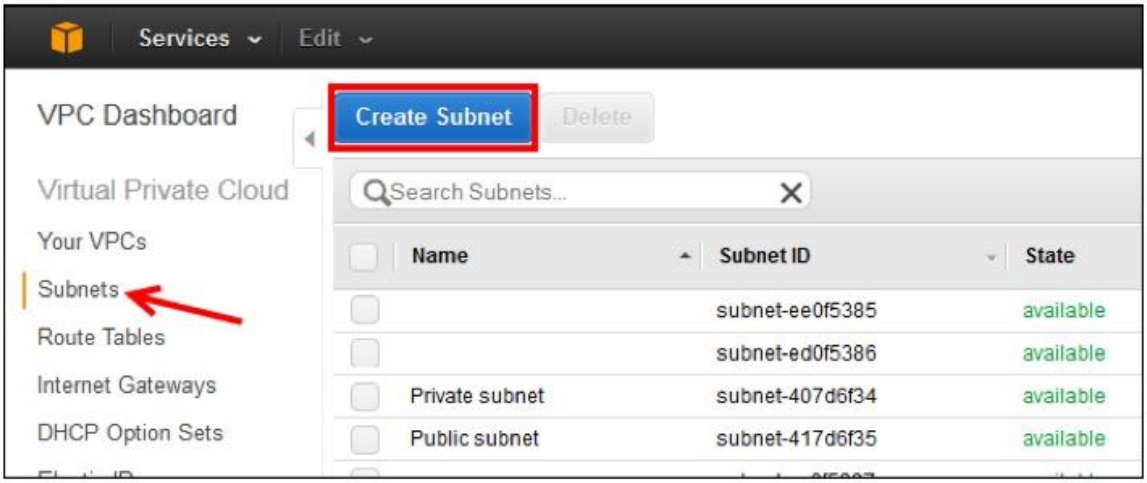
1. From the **Services** menu at the top of the screen, select **VPC.**

2. Click **Subnets** in the left panel.

You might see other Subnets that relate to a VPC with a CIDR range of `172.31.16.0/20`. This is the **Default VPC** that is automatically created for accounts. For this lab, concentrate only on the Subnets within our `10.0.0.0/16` VPC.

First, create the Public subnet:

3. Click **Create Subnet**:



4. In the **Create Subnet** dialog:

   a. **Name tag:** `Public subnet 2`

   b. **VPC:** `10.0.0.0/16`

   c. **Availability Zone:** Choose a different Availability Zone than the one used earlier.

   d. **CIDR Block:** `10.0.10.0/24`

   e. Click **Yes, Create**.

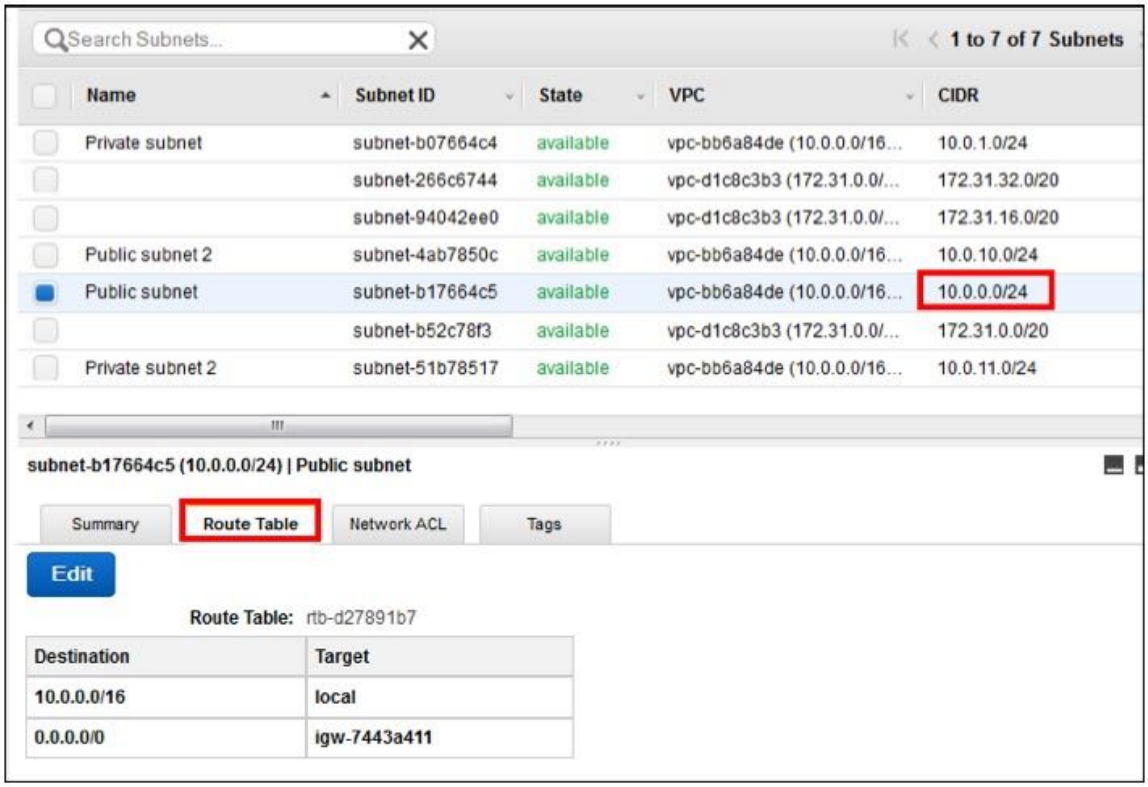
Now create the Private subnet:

5. Repeat the above steps to create a another subnet in the same Availability Zone with a **CIDR Block** of: `10.0.11.0/24`

What Determines Whether a Subnet is Public or Private?

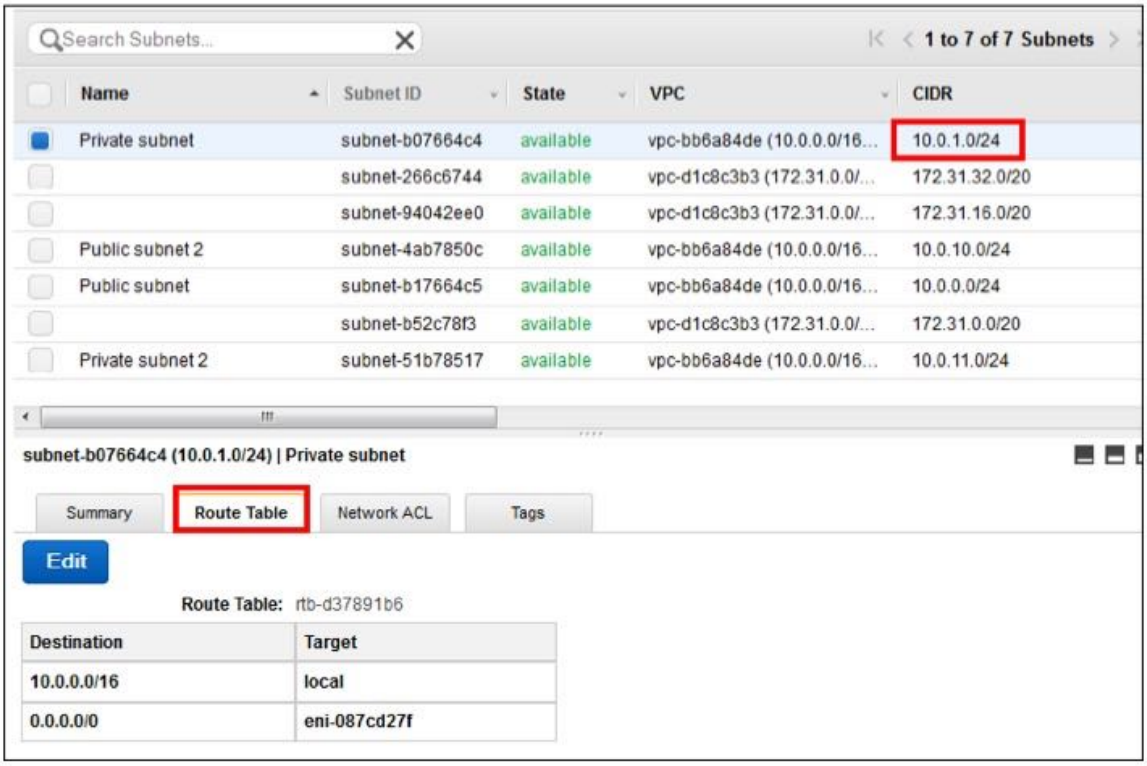
You now have two more subnets, but what makes them private or public? It is the routing rules!

1. Select the subnet using CIDR `10.0.0.0/24`, and note that there are two Routing Rules in the **Route Table**:
   - Any machine in this subnet can communicate with any other machine in `10.0.0.0/16`, which is the entire VPC. In other words, communication between all subnets is fully permitted. Later in this lab you examine security groups as a mechanism to restrict traffic.
   - Any traffic to/from the Internet (`0.0.0.0/0`) will be routed through the Internet Gateway. You have not yet looked at the Internet Gateway, but think of it as a router on the edge of the VPC. In fact, that is how it is depicted in the network diagrams.



The Details tab is also displaying Network ACLs ("Access Control Lists"), which can further limit network traffic.

2. Select the `10.0.1.0/24` subnet and note the following rules:
   - Traffic bound for any destination within the VPC (`10.0.0.0/16`) is permitted.
   - Traffic destined for the Internet (`0.0.0.0/0`) will flow to the NAT instance. The NAT is configured to only forward outbound traffic and replies to that traffic. It will not forward any other inbound traffic.



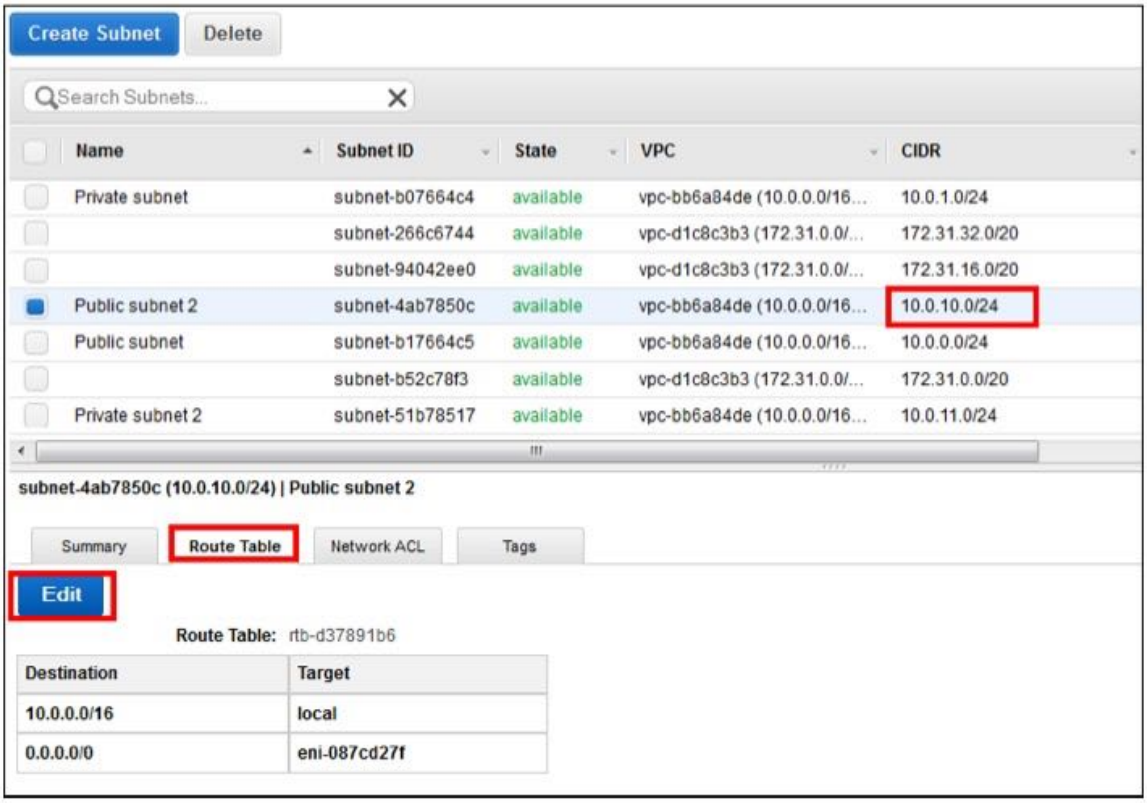
3. Click **Route Tables** in the left pane menu.

According to this view, only 1 subnet is associated with any routing rule in the `10.0.0.0/16` VPC, yet you have created 4 subnets! Why is this?

The Amazon VPC operates on a "safety first" principle. Note that one of the `10.0.0.0/16` rule sets is marked as "Main = Yes". If a subnet is not explicitly associated with a routing ruleset, it uses the Main ruleset, which happens to be the ruleset that does not talk directly to the Internet.

You need to associate the new public subnet (`10.0.10.0/24`) with the routing ruleset that routes bi-directionally to the Internet.

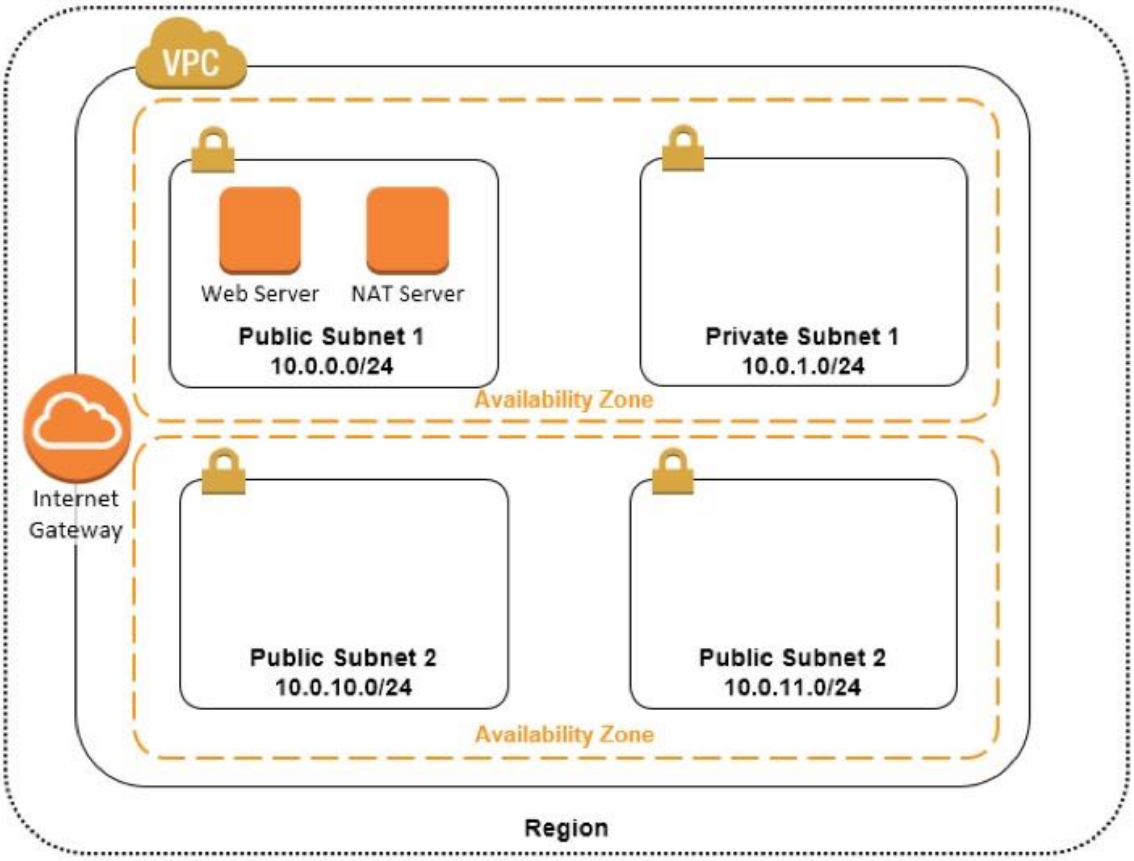
4. Click **Subnets** in the left panel.
5. Select the `10.0.10.0/24` subnet.
6. On the **Details** tab, to the right of **Route Table**, click **Edit** to replace the ruleset.



7. From the drop-down list, select the **Route Table** which is different from the current setting.

8. Click **Save**.

Your VPC now looks like this:

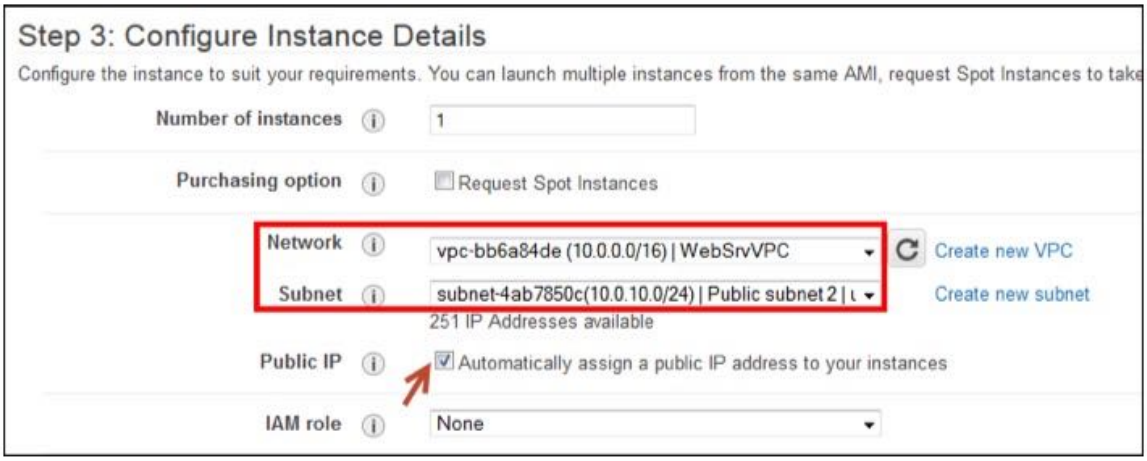


Launching a Bastion Windows Host

A bastion host (or "jump box") is a computer that is configured to prevent unauthorized network access. The bastion host is typically in front of a firewall or in a corporate DMZ. The bastion host usually runs a very limited set of services (such as a proxy server) so there are fewer network entry points that can be exploited.

You will create your bastion host in your new public subnet, though the original public subnet would also work.

1. From the **Services** menu at the top of the screen, select **EC2.**
2. Click **Launch Instance**.
3. Select the **Microsoft Windows Server 2008 R2 Base** AMI.
4. At the **Choose and Instance Type** panel, click **General Purpose > m1.small**.
5. Click **Next: Configure Instance Details**.
6. At the **Configure Instance Details** panel:
   a. **Network:** `10.0.0.0/16`
   b. **Subnet:** `10.0.10.0/24`



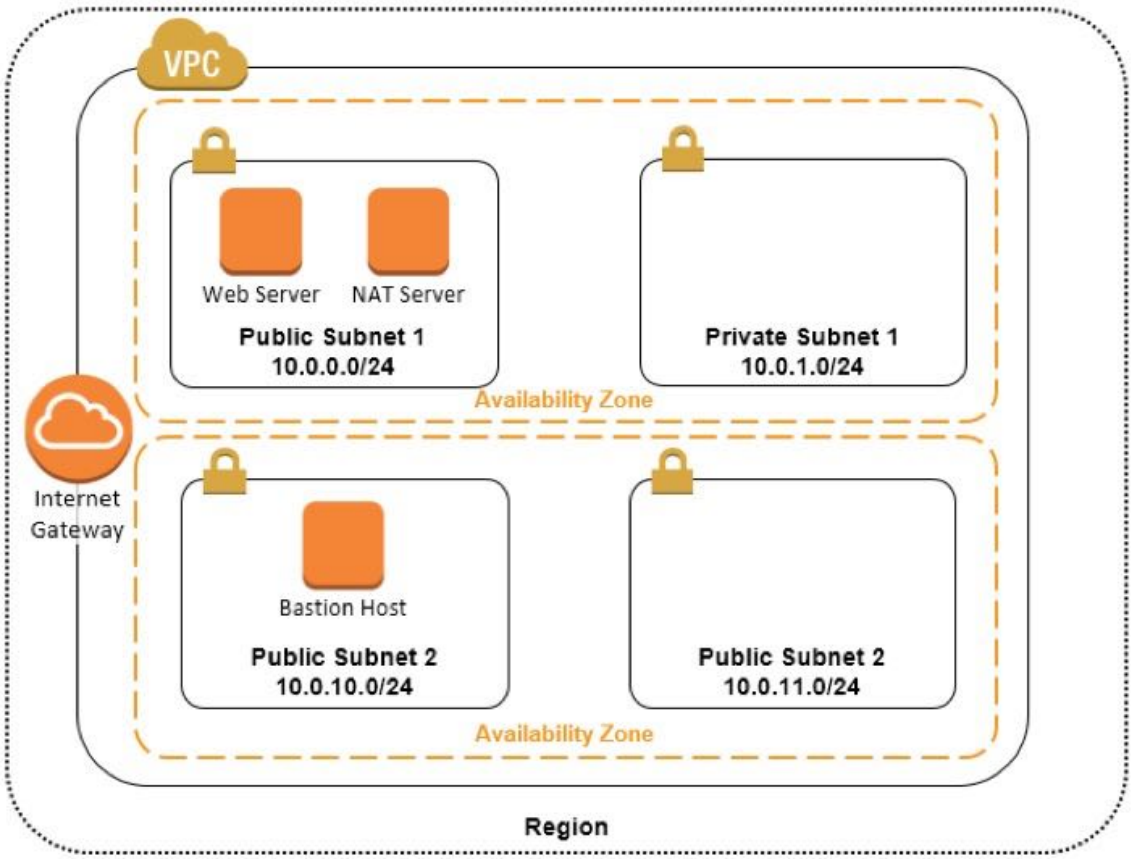
   c. Check the **Public IP** box to give this instance a public IP address automatically.
   d. Click **Next: Add Storage**.
7. There are no modifications needed in the Add Storage panel. Click **Next: Tag Instance**.
8. At the **Tag Instance** panel, enter the **Value**: `Bastion Host`
9. Click **Next: Configure Security Group**.

10. At the **Configure Security Group** panel:
    a. **Security group name:** `Bastion`
    b. **Description:** `Bastion Host Security Group`
    c. Verify there are is an existing rule for ports **3389** (RDP).

This will only allow access to port 3389, which is the Windows Remote Desktop Protocol (RDP). For this lab it is allowing access from any IP address on the Internet. Normally you will want to restrict access to the address ranges specifically required for administration.

11. Click **Review and Launch**.
12. Click **Launch**. You are presented with the **Select an existing key pair or create a new key pair** dialog.
13. Check the acknowledgement box and click **Launch Instances**.
14. Click **View Instances** (you might need to scroll down to see it).

Your VPC now looks like this:



Launching a Back-End Microsoft SQL Server

Database security is a serious subject. You will place your database in a private subnet, away from Internet traffic. You do not use the database in this lab. Rather, the objective is to create a "pot of gold" where the server is reachable via RDP under a limited set of conditions.

To secure the SQL Server, you will only open RDP access to the Bastion Host. Therefore, you must first obtain the ID of the Bastion Host Security Group.

1. Click **Security Groups** in the left panel.
2. Select the **Bastion** Security Group.
3. Make a note of the **Group ID** displayed in the lower panel. It will be used later.

You can now launch the SQL Server.

4. Click **Instances** in the left panel.
5. Click **Launch Instance**.
6. In the **Choose an Amazon Machine Image** panel, select the **Microsoft Windows Server 2008 R2 with SQL Server Web** AMI.
7. At the **Choose and Instance Type** panel, click **General Purpose > m3.medium**
8. Click **Next: Configure Instance Details**.
9. At the **Configure Instance Details** panel:
    a. **Network:** `10.0.0.0/16`
    b. **Subnet:** `10.0.1.0/24`
    c. Scroll down to find the **Network Interfaces** section. In the **Primary IP** field for the **eth0** device, type the following IP address: `10.0.1.99`

    d. Click **Next: Add Storage**.
10. At the **Add Storage** panel, click **Next: Tag Instance**.
11. At the **Tag Instance** panel, enter the **Value**: `SQL Server`
12. Click **Next: Configure Security Group**.

13. At the **Configure Security Group** panel:

    a. **Security group name:** `SQL Server`

    b. **Description:** `SQL Server Security Group`

    c. In the **RDP** line, set the **Source** to `Custom IP`. In the text box, type the Bastion Security Group ID that you noted earlier (it will be similar to `sg-a1b2c3d4`).

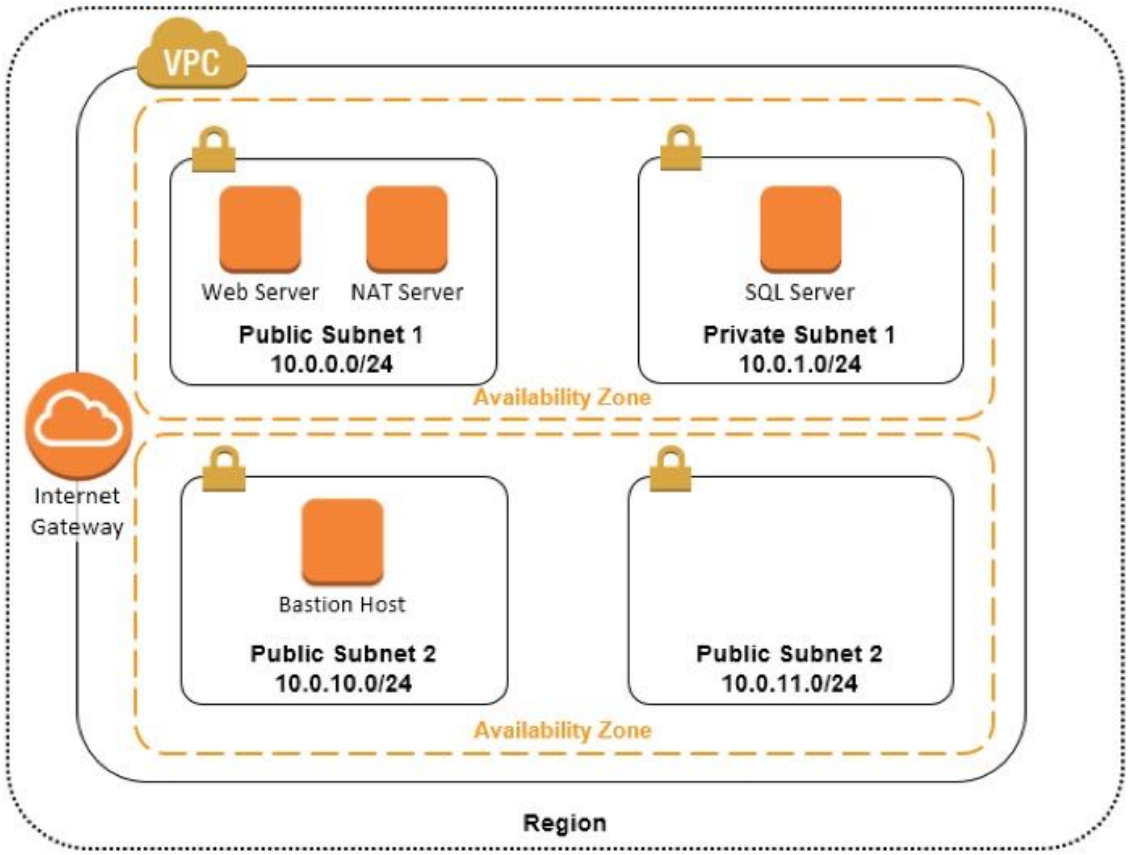
14. Click **Review and Launch**.

15. Click **Launch**. You are presented with the **Select an existing key pair or create a new key pair** dialog.

16. Check the acknowledgement box and click **Launch Instances**.

17. Click **View Instances** (you might need to scroll down to see it).

Your VPC now looks like this:



Connecting to the Bastion Host

To ensure the security of your instances, the EC2 service generates a unique, random Administrator password for each instance and encrypts it using the Key pair nominated when an instance is launched. To access the instances, you will need to decrypt the password by providing the private half of the Key pair.

You have already downloaded this key pair at the start of this lab, when you first accessed *qwik*LABS.

1. Click on your **Bastion Host** instance.
2. From the **Actions** menu, select **Get Windows Password**.

Note: It can take several minutes for a Windows Password to be generated. If the password is not ready yet, you will receive a "not available yet" message. Click **Close**, wait several minutes, and choose **Get Windows Password** again.

3. Click **Choose File** and navigate to the file that you downloaded at the start of the lab. It will be named similar to `qwikLABS-L123-45678.pem`.

If you are unable to locate the `pem` file, you can download it again from the *qwik*LABS tab in your browser, by clicking the **Download PEM** button.

4. Click **Decrypt Password**.
5. Make a note of the Public IP, user name, and password. Because you need these items later, consider pasting them into a text file. They will look similar to this:

Connecting to the Bastion Server (Windows)

Important: This section is for Windows users only. If you are using Mac or Linux, please skip ahead to the appropriate section.

You will be connecting to the Bastion Host by using **Remote Desktop Connection**.

1. On your local computer click **Start > Run**, type `mstsc` and click **OK** to start Remote Desktop Connection.
2. Click **Show Options** and enter:
    a. **Computer:** Type or paste the Public IP you noted above.

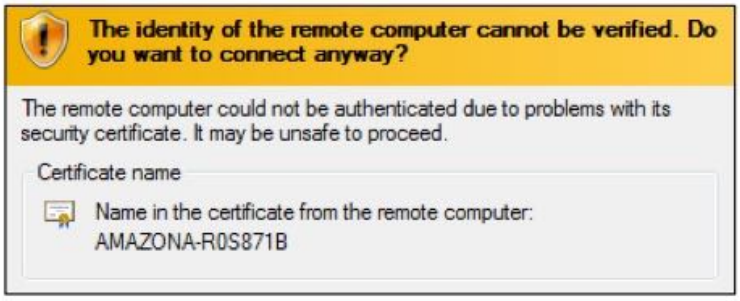
b. **User name:** `.\Administrator`

c. Click **Connect**.

The '`.\`' in the User Name instructs Remote Desktop Connection to login as a local account, just in case your system is configured to use a different domain.

3. When prompted, type the **Password** you noted.

4. Click **Yes** if you see a certificate verification message similar to "the identity of the remote computer cannot be verified":



5. Proceed to the section entitled **Connecting to the database server** later in these instructions.

Connecting to the Bastion Server (Mac)

Important: This section is for Mac users only. If you are using Linux, please skip ahead to the appropriate section.

You will connect to the Bastion Host using **Remote Desktop Connection for Mac**. This software might already be installed on your computer if you have Microsoft Office. Alternatively, it can be downloaded for free from:

http://www.microsoft.com/en-au/download/details.aspx?id=18140

1. Open the **Remote Desktop Connection for Mac** application.

2. Type the Bastion host's Public IP address and click **Connect**:

3. When prompted for credentials:

a. **User name:** `Administrator`

b. **Password:** Type or paste the password you noted earlier

c. **Domain:** Leave blank

d. Click **OK**.

4. Click **Connect** if you see a verification message similar to "the server name is incorrect."

5. Proceed to the section entitled **Connecting to the database server** later in these instructions.

Connecting to the Bastion Server (Linux)

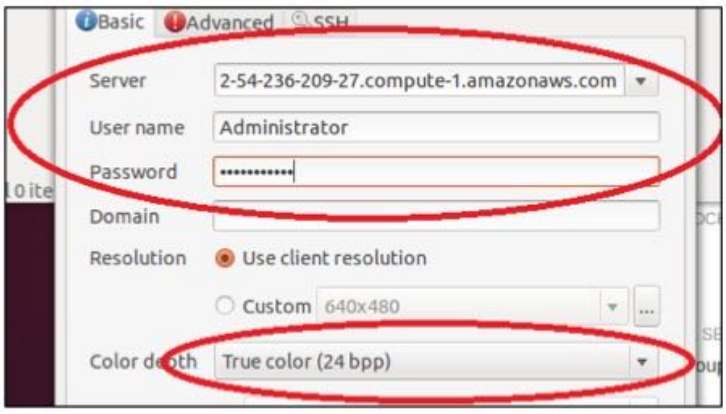
Important: This section is for Linux users only.

1. Open the **Remmina Remote Desktop Client**.

2. Type the bastion host computer's IP address, and then type the **User name** and **Password**.

3. Optionally, choose a **Color depth** that your bandwidth supports, such as 'True color (24 bpp)'.

4. Click **Connect**.



5. Click **OK** when prompted to accept the remote certificate.

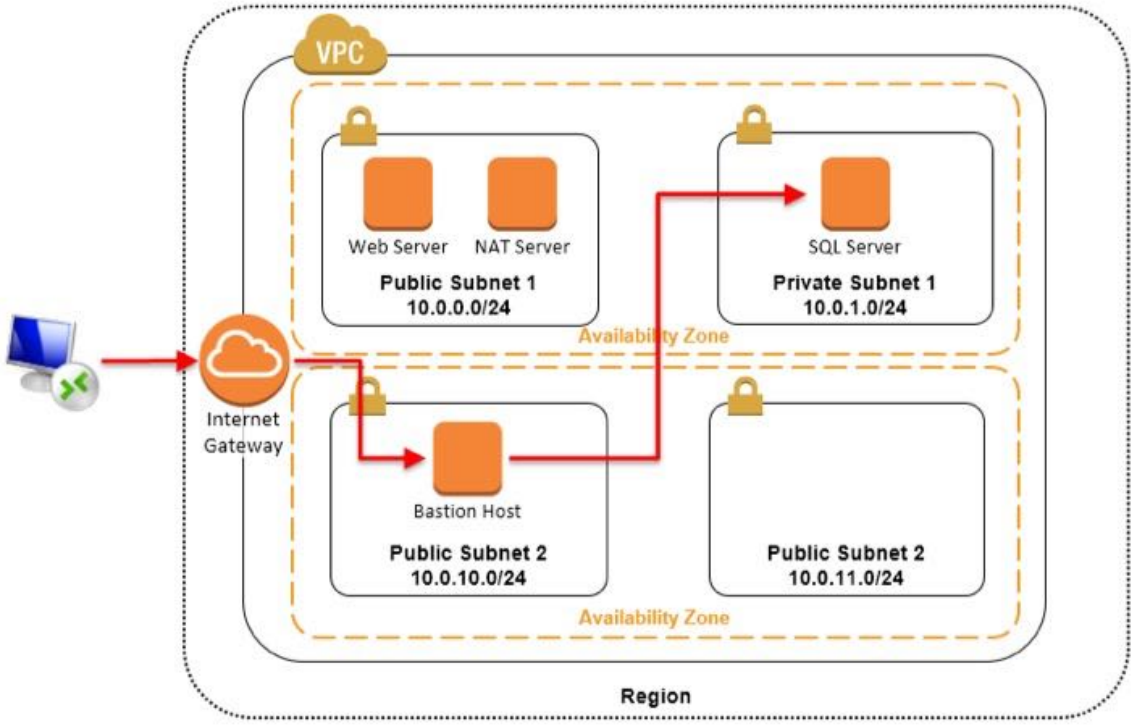
Logging in to the Database Server

You have now connected to the Bastion Host computer, which is located in the Public subnet. The next step is to connect to the SQL Server from the Bastion host.

1. In the EC2 Management Console, click on your **SQL Server** instance.
2. From the Actions menu, select **Get Windows Password**. Make a note of the Public IP and Password.
3. Return to the **Remote Desktop** session.
4. In the remote session click **Start > Run**, type `mstsc` and click **OK** to start Remote Desktop Connection.
5. Click **Show Options** and enter:
    a. **Computer:** `10.0.1.99`
    b. **User name:** `Administrator`
    c. Click **Connect**.
6. When prompted, type the **Password** you noted above.

Congratulations! You should now be connected to the SQL Server via the Bastion Host. While the SQL Server is located in a private subnet, this connection was made possible by connecting via the Bastion Server.

Your completed environment should look like the following diagram. The line shows how your connection flows through the Internet Gateway, through the Bastion Host and then to the SQL Server.



You might wonder why you created the second private subnet (labeled Private Subnet 2). This subnet is where you could create a replica of the SQL Server.

Additional Task (Optional)

Now that you are connected to the SQL Server, try opening a web browser and connecting to a website. You will find that the connection does not work. Your extra challenge is to find out why this connection is refused.

Hint: It has to do with the AWS security group settings on the NAT Server and SQL Server.

Conclusion

As you just learned, there are multiple ways to control how to access servers that are kept in private subnets. To help ensure that your network is secure, pay attention to the subnets containing your servers.

Bastion Hosts and VPN tunnels each have an advantage. Bastion hosts allow you a secure method of logging in to manage servers, especially if only a few people need to perform this activity. If you want the VPC to act as a virtual extension to your corporate network, then a VPN may make more sense.

Ending the Lab

End Your Lab

1. To log out of the AWS Management Console, from the menu, click `awsstudent @ [YourAccountNumber]` and choose **Sign out**:

2. Close any active SSH client sessions or remote desktop sessions.
3. Click the **End Lab** button on the *qwik*LABS lab details page.

4. When prompted for confirmation, click **OK**.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com

Lab 2: Working with Amazon Identity and Access Management

Overview

In this lab, you will use the Amazon Identity and Access Management service to create users and roles within an AWS environment. You will then test the permissions of these users and roles to verify that they can only perform specific actions within the AWS environment.

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with:

- Navigating the AWS Management Console
- Launching and terminating EC2 instances
- Connecting to and launching commands from an EC2 instance

Topics Covered

Here's what you'll do during this lab:

1. Create and test a user that only has full access to Amazon S3.
2. Create and test a user that only has full access to Amazon EC2.
3. Create a role that allows only describe permissions to EC2 instances.

Login to the AWS Management Console

Using qwikLABS to login to the AWS Management Console

The first step is for you to login to Amazon Web Services.

1. To the right of the lab title, click the **Start Lab** button to launch your *qwik*LABS. If you are prompted for a token, use the one distributed to you (or the token you purchased).

2. On the lab details page, notice the lab properties.

    a. **Duration -** The time the lab will run before automatically shutting down.

    b. **Setup Time -** The estimated time to set up the lab environment.

    c. **AWS Region** - The AWS Region in which the lab resources are created.

The AWS Region for your lab will differ depending on your location and the lab setup.

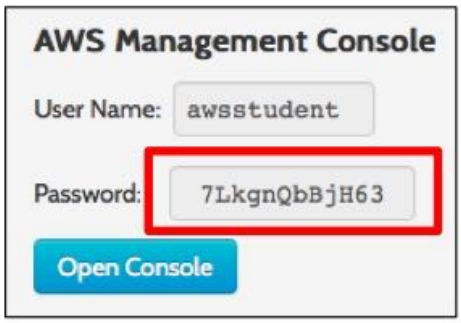
3. Click the **Download PEM/PPK** button and then:

    a. Windows users: Select **Download PPK**

    b. Mac & Linux users: Select **Download PEM**

A file will download to your computer for use later in the lab, so remember where it was saved (such as the Downloads folder):

4. In the AWS Management Console section of the *qwik*LABS page, copy the **Password** to the clipboard.



5. Click the **Open Console** button:

6. Log into the AWS Management Console using the following steps.

    a. In the **User Name** field type `awsstudent`.

    b. In the **Password** field, paste the password copied from the lab details page.

    c. Click **Sign in**.

Note: The AWS account is automatically generated by *qwik*LABS. Also, the login credentials for the `awsstudent` account are provisioned by *qwik*LABS using AWS Identity Access Management.

Creating IAM users and policies

In this part of the lab, you will:

- Create an IAM user that has full permissions only to Amazon S3
- Create an IAM user that has full permissions only to Amazon EC2
- Create a role that allows only describe permissions to Amazon EC2

Create users in IAM

To make things more efficient, you'll first create two users in IAM, and then assign specific security policies to each of them. To start you first need to create the users.

To create users:

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Users**.
3. Click **Create New Users**.
4. In the first **Enter User Names** field, type `AsperatusTechS3TestUser`.
5. In the second **Enter User Names** field, type `AsperatusTechEC2TestUser`.
6. Verify that the Generate an access key for each user option is enabled.
7. Click **Create**.

Amazon IAM creates the users for you, and gives you the option to download the users' credentials.

To download the users' credentials:

1. Click **Download Credentials**.
2. Open the file and re-save it as `AsperatusTechTestUserCredentials`.
3. From the AWS Management Console, click **Close Window**.

Take a moment to open the credential file. Notice that you have three columns: **User Name**, **Access Key Id**, and **Secret Access Key**. In addition to these columns, add a fourth: Password, which you'll use in the next set of steps.

By default, users that you create do not have access to the AWS Management Console. To grant this access, you need to create a password for the user.

To create a password:

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Users**.
3. Select **AsperatusTechS3TestUser**.
4. From the **User Actions** menu, select **Manage Password**.
5. Select **Assign an auto-generated password** and click **Apply**.
6. A new dialog box appears, confirming that IAM has generated the user's password.
7. Click **Show User Security Credentials**.
8. Open AsperatusTechTestUserCredentials file you created earlier.
9. Add the password to the Password column of the **AsperatusTechS3TestUser** entry.
10. Click **Close Window**.
11. IAM asks you to confirm that you want to close the window, as you haven't downloaded the user's password. As you've copied the password to an existing file, it's okay to click **Close Window** again to close the window.
12. Repeat this process for the AsperatusTechEC2TestUser.

Creating an IAM user that has full permissions to Amazon S3 only

You now have two users in IAM. Take a look at the user, AsperatusTechS3TestUser.

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Users**.
3. From the list of users, select **AsperatusTechS3TestUser**.

4. In the Details pane, select the Permissions tab.

Notice that this user does not have any permissions for AWS. You need to change this so the user can access S3 buckets.

To add S3 permissions to the user:

1. From the Permissions tab for the user, click **Attach User Policy**.
2. Choose **Select Policy Template**.
3. Locate the **Amazon S3 Full Access** policy template and click **Select**.
4. A new dialog box appears that allows you to customize the policy further. Take a look at this policy as it is now. Notice that it only allows access to Amazon S3. As this is exactly what you want, you can leave the policy unchanged.
5. Click **Apply Policy**.

Create an IAM user that has full permissions to Amazon EC2 only

Your next task is to create a user that has full permissions (Start, Stop, Terminate, and so on) to Amazon EC2 instances. This time, instead of creating a single user with these permissions, you need to create a user group that has these permissions, and then create a user that belongs to that group.

To create a user group in IAM:

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Groups**.
3. Click **Create New Group**.
4. In the **Group Name** box, type AsperatusTechEC2TestGroup.
5. In the **Set Permissions** dialog box, choose **Select Policy Template**.
6. Locate the **Amazon EC2 Full Access** policy and click **Select**.
7. A new dialog box appears that allows you to customize the policy further. Take a look at this policy as it is now. Notice that it not only allows full access for EC2, but also to Elastic Load Balancers and CloudWatch. These additional services are added because they are closely tied to Amazon EC2. In this case, your security team has no objections to having these additional services available to these users, so you can leave the policy unchanged.
8. Click **Continue**.
9. Click **Create Group**.

With your group created, your next step is to add a user to the group.

To add a user to a group:

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Groups**.
3. Select **AsperatusTechEC2TestGroup**.
4. From the **Group Actions** menu, select **Add Users to Group**.
5. From the dialog box that appears, select **AsperatusTechEC2TestUser** and then click **Add Users**.

Creating a role that has EC2 describe permissions only

Next, you've been asked to create a role within AWS. This role, when assigned to a resource (like an EC2 instance), allows anyone using that resource with describe permissions to Amazon EC2 instances. This means they can list what EC2 instances are running, but cannot start, stop, or otherwise change them.

To create a role:

1. From the AWS Management Console, access the IAM dashboard.
2. Click **Roles**.
3. Click **Create New Role**.
4. In the **Role Name** box, type `AsperatusTechEC2Describe`.
5. Click **Continue**.
6. A dialog box appears, allowing you to select a role.
7. Choose the **AWS Service Roles** option.
8. Locate the **Amazon EC2** role and click **Select**.
9. Another dialog box appears, allowing you to select a policy template for the role.
10. Choose the **Select Policy Template** option.
11. Locate the **Amazon EC2 Read Only Access** policy and click **Select**.
12. A new dialog box appears that allows you to customize the policy further. Take a look at this policy. Just as when you created your AsperatusTechEC2Group earlier, this policy allows describe actions not only on EC2 instances, but on Elastic Load Balancing, CloudWatch, and Auto Scaling.
13. Click **Continue**.
14. Click **Create Role**.

Testing IAM users

You should now have the following:

- A user that has full access to S3 resources only
- A user that has full access to EC2 resources only
- A role that has read access to EC2 resources only

Next, you'll test each of these to see how they function. Before you start, first get the URL associated with your main AWS account for this lab.

To access the URL:

1. From the AWS Management Console, access the **IAM Dashboard**.
2. Locate the section, **IAM User Sign-In URL**.
3. Copy the URL shown under **IAM users sign-in link**.

Test the S3 user account

For this test, you'll need the password you created for your AsperatusTechS3TestUser account.

To test the user:

1. Open a new browser window. We recommend using a private browsing mode, such as Internet Explorer's InPrivate mode or Chrome's Incognito mode.
2. Navigate to your AWS Account Alias URL.
3. In the **Username** box, type `AsperatusTechS3TestUser`.
4. In the **Password** box, type the password that you generated for this account.
5. You should now see the main screen of the AWS Management Console.
6. Access the S3 dashboard.
7. Create a bucket and add a file to it.
8. You've now confirmed that you can view and create S3 resources. Next, let's look at what the account cannot do.
9. Access the EC2 dashboard.
10. Notice that, in the main screen of the dashboard, there are messages stating that you cannot describe various aspects of EC2 instances.
11. Click **Instances**.
12. Notice that you see a message: `An error occurred fetching Instance data.` Try other AWS services (such as DynamoDB) and confirm that this account can only access and modify S3 resources. When you are finished, sign out of the account.

Test the EC2 account

Next, take a look at what you can do with the AsperatusTechEC2TestUser account.

To test the user:

1. Open a new browser window. We recommend using a private browsing mode, such as Internet Explorer's InPrivate mode or Chrome's Incognito mode.
2. Navigate to your AWS Account Alias URL.
3. In the **Username** box, type `AsperatusTechEC2TestUser`.
4. In the **Password** box, type the password that you generated for this account.
5. You should now see the main screen of the AWS Management Console.
6. Access the EC2 dashboard.
7. Launch an instance.
8. You've now confirmed that you can view and create EC2 resources. Next, let's look at what the account cannot do.
9. Access the S3 dashboard.
10. Notice that you cannot access any S3 resources.

As with the AsperatusTechS3TestUser account, try a few other AWS products, noting what you can and cannot do with this user account. When you're finished, sign out.

Test the EC2 role

As a final test, let's see the capabilities of the EC2 role.

To test the EC2 role:

1. Return to your main AWS Management Console. (This is the console that is using your qwikLab credentials.)
2. From the AWS Management Console, access the EC2 Dashboard.
3. Click **Instances**.
4. Click **Launch Instance**.
5. Locate the Amazon Linux AMI and click **Select**.
6. From the All Instances Types tab, choose **m1.small** and then click **Next: Configure Instance Details.**
7. From the **IAM Role** list, select **AsperatusTechEC2Describe**.
8. Click **Review and Launch**.

9. Click **Launch**. You are presented with the **Select an existing key pair or create a new key pair** dialog.
10. Check the acknowledgement box and click **Launch Instances**.
11. Click **View Instances** (you might need to scroll down to see it).
12. Connect to the new instance. You'll need to use the new key pair to access it.
13. Attempt to run an EC2 describe command, such as `aws ec2 describe-instances`. (Review the command-line documentation for assistance: http://docs.aws.amazon.com/cli/latest/reference/ec2/describe-instances.html.) Be sure to specify a region when you use this command.
14. You should be able to get information about the instances running in your region.
15. Attempt to launch a new instance using the `aws ec2 run-instances` command. (Review the command-line documentation for assistance: http://docs.aws.amazon.com/cli/latest/reference/ec2/run-instances.html.) You'll need an AMI ID to use this command—a list of AMIs is available here: http://aws.amazon.com/amazon-linux-ami/.

Notice that you are not allowed to run this command with your current set of permissions.

Conclusion

Congratulations! You have now used IAM to create two users, each with a different set of permissions. You've also created an IAM role that you can assign to an EC2 resource.

End Your Lab

1. To log out of the AWS Management Console, from the menu, click `awsstudent @ [YourAccountNumber]` and choose **Sign out**:

2. Close any active SSH client sessions or remote desktop sessions.
3. Click the **End Lab** button on the *qwik*LABS lab details page.

4. When prompted for confirmation, click **OK**.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com

Lab 3: Getting Started with Auto Scaling

Overview

Auto scaling represents more than a way to add and subtract servers. It is also a mechanism to handle failures similar to the way that Load Balancing handles unresponsive servers. This lab demonstrates how to configure Auto Scaling to automatically launch and monitor Amazon EC2 instances, and update an associated Elastic Load Balancer (ELB).

What is Auto Scaling?

Auto Scaling allows you to scale your Amazon EC2 capacity up or down automatically according to conditions you define. With Auto Scaling, you can ensure that the number of Amazon EC2 instances you're using increases seamlessly during demand spikes to maintain performance, and decreases automatically during demand lulls to minimize costs. Auto Scaling is particularly well suited for applications that experience hourly, daily, or weekly variability in usage.

Auto Scaling is enabled by Amazon CloudWatch and available at no additional charge beyond Amazon CloudWatch fees.

Topics Covered

This lab will take you through Auto Scaling, including:

- Creating a Launch Configuration
- Creating an Auto Scaling Group
- Activating Auto Scaling Notifications
- Creating Scaling Policies
- Testing Auto Scaling by triggering Scaling Policies

Login to the AWS Management Console

Using qwikLABS to login to the AWS Management Console

Welcome to this self-paced lab! The first step is for you to login to Amazon Web Services.

1. To the right of the lab title, click the **Start Lab** button to launch your *qwik*LABS. If you are prompted for a token, use the one distributed to you (or the token you purchased).

Note: A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.

2. On the lab details page, notice the lab properties.
    a. **Duration -** The time the lab will run before automatically shutting down.
    b. **Setup Time -** The estimated time to set up the lab environment.
    c. **AWS Region** - The AWS Region in which the lab resources are created.

Note: The AWS Region for your lab will differ depending on your location and the lab setup.

3. In the AWS Management Console section of the *qwik*LABS page, copy the **Password** to the clipboard.

4. Click the **Open Console** button:

5. Log into the AWS Management Console using the following steps.
    a. In the **User Name** field type `awsstudent`.
    b. In the **Password** field, paste the password copied from the lab details page.
    c. Click **Sign in**.

Note: The AWS account is automatically generated by *qwik*LABS. Also, the login credentials for the `awsstudent` account are provisioned by *qwik*LABS using AWS Identity Access Management.

Overview of Auto Scaling

Auto Scaling Principles

First, auto scaling is a way to set the "cloud temperature". You use rules to "set the thermostat", and, under the hood, Auto Scaling "controls the heat" by adding and removing EC2 instances on an as-needed basis, in order to maintain the "temperature" (capacity).

Second, Auto Scaling assumes a set of homogeneous servers. That is, Auto Scaling does not know that Server A is a 64-bit extra-large instance and is more capable than a 32-bit small instance. In fact, this is a core tenet of cloud computing: **scale horizontally using a fleet of fungible resources**. An individual resource is not important, and accordingly the philosophy is "easy come, easy go".

The Key Components of Auto Scaling

When you launch a server manually, you provide parameters such as the Amazon Machine Image (AMI), instance type and security group. Auto scaling calls this a **Launch Configuration**. It is simply a set of parameters.

Auto Scaling Groups tell the system what to do with an instance once it launches. This is where you specify which Availability Zones it should use, which Elastic Load Balancer it will receive traffic from and, most importantly, this is where you specify the minimum and maximum number of instances to run at any given time.

Scaling Policies tell Auto Scaling when to add or remove instances. They have rules such as "scale the fleet out by 10%" or "scale in by 1 instance."

Scaling Policies can be triggered manually, on a schedule or, most powerfully, by **CloudWatch Alarms**. Alarms inspect CloudWatch metrics and change state when conditions change, such as:

- "When average CPU across instances in the Auto Scaling group drops below 40% for 15 minutes", or
- "When average CPU across instances in the Auto Scaling group exceeds 65% for 10 minutes".

An alarm can be in the **Alarm**, **OK**, or **Insufficient Data** states. The last is a special state for when there is no data available to determine the state of the alarm.

Timing Matters

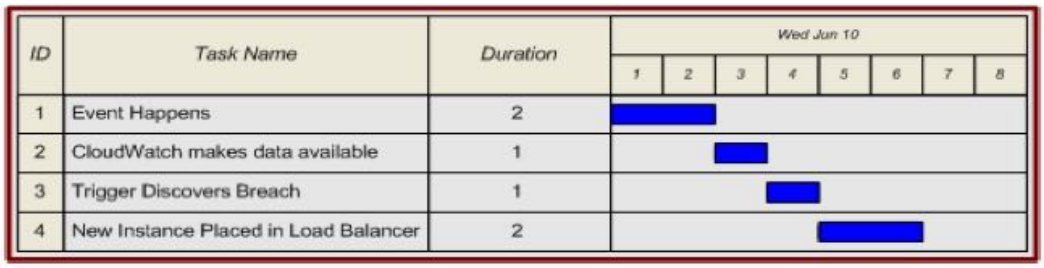
There are costs related to using Auto Scaling. There are two important concepts that directly affect the cost of AWS and the manner in which your application scales.

The Minimum Unit of Cost for EC2 is 1 Hour

It does not matter whether an EC2 instance runs for 60 seconds or 60 minutes – AWS bills for the full hour. Accordingly it is very important to avoid a short-cycle situation where a server is added to the fleet for 10 minutes, decommissioned and then another server is added a few minutes later.

Scaling Takes Time

Consider the graph below. In most situations a considerable amount of time passes between when there is the **need** for a scaling event, and **when** the event happens.



- In this example, the rule says that you must be in a particular condition for at least 2 minutes.
- CloudWatch is the underlying data collection system that monitors statistics such as CPU utilization. It is a polling protocol and in general takes 60 seconds to aggregate the data.
- Auto Scaling is also a polling system and it takes another 60 seconds.
- Then there is boot time for your server. A large, complex, server may take many minutes to launch.
- Finally, the Load Balancer needs to poll the server for a few cycles before it is comfortable that the server is healthy and accepting requests.

Therefore, it is advisable to consider the full response time when creating your Scaling Policies.

Creating an Elastic Load Balancer

Before you create an Auto Scaling Group, you first need an Amazon Elastic Load Balancer (ELB). This load balancer will send requests to your EC2 instances, dynamically distributing them across EC2 instances as the Auto Scaling Group increases and decreases in size.

1. When you are logged into the console, click **EC2**.

2. Click **Load Balancers** in the left panel.

3. Click **Create Load Balancer**.

4. In the **Define Load Balancer** panel:
    a. **Load Balancer name:** `auto-scaling-elb`
    b. Click **Continue**
5. In the **Configure Health Check** panel, set the Health Check parameters:
    a. **Ping Protocol:** `TCP`
    b. **Health Check Interval:** `10`
    c. **Healthy Threshold:** `2`
    d. Click **Continue**
6. In the **Assign Security Groups** panel, allow incoming web traffic:
    a. Click **Create a new security group** (at the top)
    b. **Security group name:** `Web`
    c. **Description:** `Web traffic`
    d. Traffic on port 80 is already permitted, so click **Continue**
7. In the **Add Instance to Load Balancer** panel, click **Continue**.
8. Click **Create**.
9. Click **Close** to return to the EC2 dashboard.

Creating a Launch Configuration

Your first step in creating an Auto Scaling group is to create a **Launch Configuration**. A Launch Configuration specifies details such as which AMI ("Amazon Machine Image") to use when launching new instances, which instance type to use, and what configuration scripts should be run.

1. Click **Launch Configurations** in the left panel (you may need to scroll down to see it). Since you have not yet created an Auto Scaling group, the console assumes you ultimately want to create an Auto Scaling group.
2. Click **Create Auto Scaling group**.
3. Click **Create launch configuration**.

The next few steps should appear familiar to you – they are very similar to the steps you follow when launching an individual EC2 instance.

4. On the **Choose AMI** panel, select the **Amazon Linux AMI**.
5. On the **Create Instance Type** panel, click **Next: Configure details**.

6. On the **Configure details** panel:

    a. **Name:** `Web launch configuration`

    b. **Monitoring:** Check the **Enable CloudWatch detailed monitoring** check box

    c. Expand the **Advanced Details** section.

    d. Enter the following text in the **User Data** field.
    Note: The text is available in a **command reference text file** attached to the *qwik*LABS page for this lab.

```
#!/bin/sh
yum -y install httpd php mysql php-mysql
chkconfig httpd on
/etc/init.d/httpd start
cd /tmp
wget https://us-west-2-aws-training.s3.amazonaws.com/architecting-lab-2-autoscaling-3.1/static/examplefiles-as.zip
unzip examplefiles-as.zip
mv examplefiles-as/* /var/www/html
```

This script will automatically run when the instance starts. The script loads a web server and database, then downloads content for the website. This is an example of 'bootstrapping' an EC2 instance to perform a particular function, without requiring a pre-configured AMI.

    e. Click **Next: Add Storage**.

7. There are no modifications needed in the **Add Storage** panel. Click **Next: Configure Security Group**.

8. At the **Configure Security Group** panel, permit inbound Web and SSH traffic:

    a. **Security group name:** `Web-SSH`

    b. **Description:** `Web and SSH`

    c. Click **Add Rule** to add another rule (there is already a rule for SSH)

    d. Set the **Protocol** drop-down to `HTTP`

    e. Click **Review**

9. Click **Create launch configuration**. You are presented with the **Select an existing key pair or create a new key pair** dialog.

10. Check the acknowledgement box and click **Create launch configuration**.

You are now prompted to create an Auto Scaling group that uses this Launch Configuration.

Creating an Auto Scaling Group

While the Launch Configuration controls **what** instances are launched and how they are configured, the Auto Scaling group controls **when** instances are launched or terminated, and what criteria triggers an auto scaling Action.

At the conclusion of the last section, you should have ended at a screen that starts the creation of an Auto Scaling group. If you do not see this screen:

- From the AWS Management Console, select **EC2** from the **Services** menu and then click **Auto Scaling Groups**.
- Click **Create auto scaling group**.
- Select the **Launch Configuration** you created in the last section and click **Next Step**.

1. In the **Configure Auto Scaling group details** panel:
   a. **Group Name:** `web-group`
   b. **Subnet:** Select at least one subnet by clicking in the Subnet box
   c. Expand the **Advanced Details** section.
   d. In the Load Balancing section, check the **Receive traffic from Elastic Load Balancer(s)** option.
   e. Click inside the box that appears and select the `auto-scaling-elb` load balancer that you created earlier.
   f. Click **Next: Configure Scaling Properties**.
2. Verify that **Keep this group at its initial size** option is selected and click **Next: Configure Notifications**.
3. Click **Review**.
4. Click **Create Auto Scaling Group**.
5. Click **Close**.

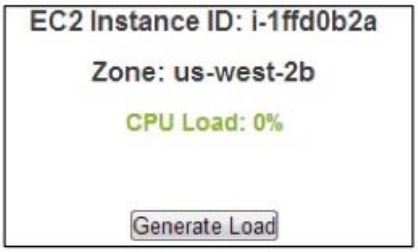
Verifying and Testing Auto Scaling

In this section of the lab, you will verify your lab environment and test your Auto Scaling group.

Verifying that the Servers Launched

You will use the AWS Management Console to inspect your instance count. You should see one instance in your fleet because you set the minimize size to 1 (you may need wait a few minutes before it appears).

1. Click **Instances** in the left panel. You should see one instance running.
2. If the Status Checks for this instance still says **Initializing**, wait a few minutes. (You can periodically click the refresh button to update the dashboard.
3. On the **Details** tab below, copy the **Public DNS** name of the instance into your clipboard.
4. **Open a new tab** in your web browser, **paste the DNS address** and hit Enter. You should see something similar to the following:



This confirms that your Auto Scaling group successfully launched the instance and used the script from the **User Data** to bootstrap and configure the instance as a web server.

Verifying That Auto Scaling Works

1. In the AWS Management Console, stop the instance by selecting **Stop** from the **Actions** menu.
2. When the confirmation box appears, click **Yes, Stop**.

Auto Scaling will detect that the instance has become non-responsive and will terminate it. A replacement instance will be launched automatically because the fleet size is below the minimum size. **Refresh the console periodically** to see the new instance.

Tagging Auto Scaling Resources

Notice that the Auto Scaling instances are launched without names. You can make these resources easier to identify by adding a new column to the Management Console.

1. Click the **Show/Hide** button ⚙ in the top-right, then check `aws:autoscaling:groupName` on the left.
2. Click **Close**. Auto scaling automatically displays a column with the name of the Auto Scaling group that launched the instance.

Adding Auto Scaling Notifications

Typically, Auto Scaling activities are occurring transparently. You can configure Auto Scaling to notify you when it automatically creates or terminates instances. Auto Scaling sends notifications via the Amazon Simple Notification Service (SNS). SNS is a web service that makes it easy to send and subscribe to notifications from the AWS cloud. It provides developers with a highly scalable, flexible, and cost-effective capability to publish messages from an application and immediately deliver them to subscribers or other applications.

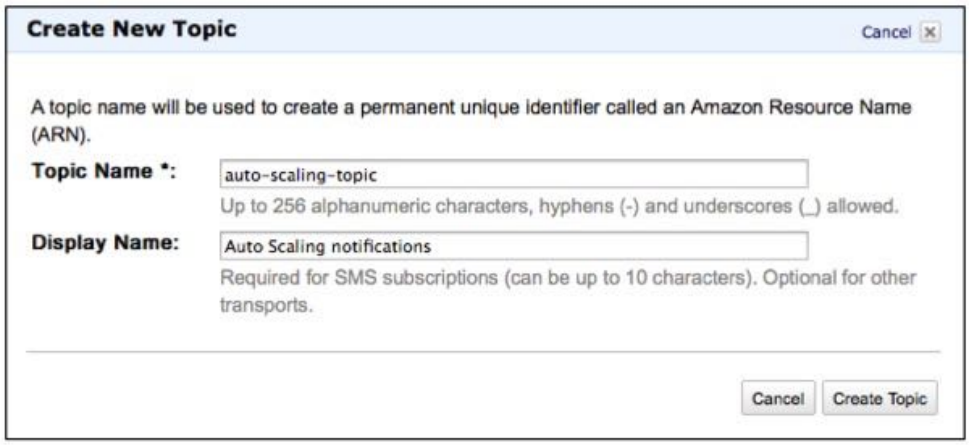
Creating an SNS Topic

First, create an SNS topic used to send SNS notifications.

1. From the AWS Management Console, select **SNS** from the **Services** menu.
2. Click **Create New Topic**, then:
    a. **Topic Name:** `auto-scaling-topic`

b. **Display Name:** `Auto Scaling notifications`

c. Click **Create Topic**



3. Click the **Create Subscription** button, then:

   a. **Protocol:** `Email`

   b. **Endpoint:** Type an email address that you can access from the classroom, so you can view email notifications.

   c. Click **Subscribe**.

4. Check your email and click the link in the message to confirm your subscription to the topic.

Creating an Auto Scaling Notification

You can now update the Auto Scaling group to use the new notification.

1. From the AWS Management Console, select **EC2** from the **Services** menu.
2. Click **Auto Scaling Groups** in the left panel (you may need to scroll down to see it).
3. Select your Auto Scaling group.
4. Click the **Notifications** tab in the lower half of the window.
5. Click **Create Notification**, then:

   a. Verify that your `auto-scaling-topic` is selected from the **Send a notification** to list. Notice that you can select to receive notifications when instance launch, fail to launch, terminate, or fail to terminate. Leave all the options enabled.

   b. Click **Save**.

6. Check your email. You should receive a test notification email confirming the configuration.

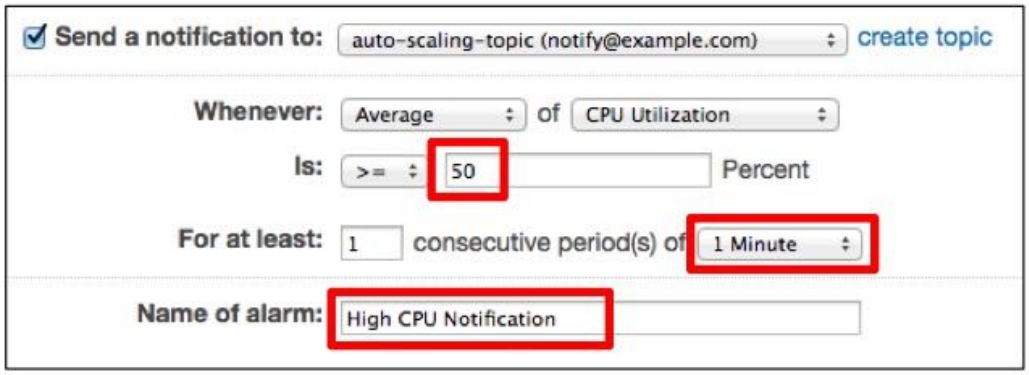
Creating Auto Scaling Policies

You currently have an Auto Scaling group that continually ensure that you have one running instance. You will now add an Auto Scaling policy that automatically adds and removes instances based on specific criteria. In this case, the policy will instruct the auto scaling group to automatically scale up whenever the average CPU of the web server fleet is >= 50%.

Creating a scale-out policy

First, you will create a scale-out policy that adds instances when CPU utilization reaches a certain threshold.

1. Select your Auto Scaling group.
2. Click the **Scaling Policies** tab (in the lower half of the window).
3. Click **Add Policy**, then:
   a. **Name:** `web-scale-out`
   b. Click **Create new alarm.**
   c. Enter these details:



   d. Click **Create Alarm**.

e. **Take the Action:** `Add 1 instances`

f. Click **Create**.

Creating a scale-in policy

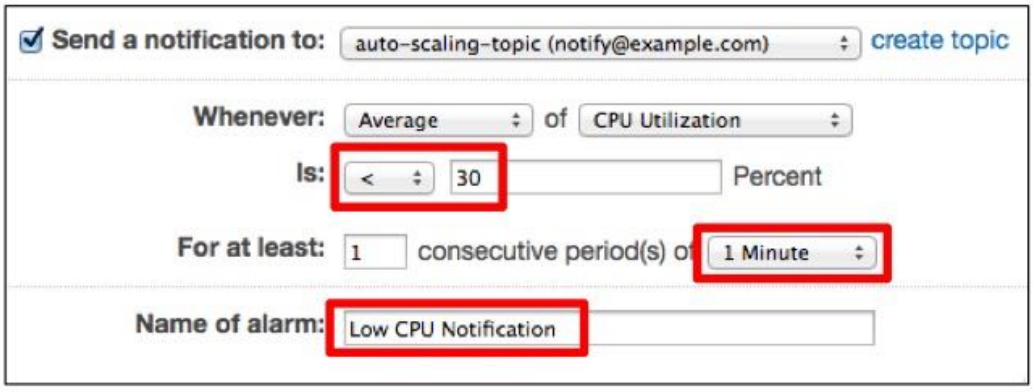
Now you can create a corresponding scale-in policy.

1. Click **Add Policy**, then:

   a. **Name:** `web-scale-in`

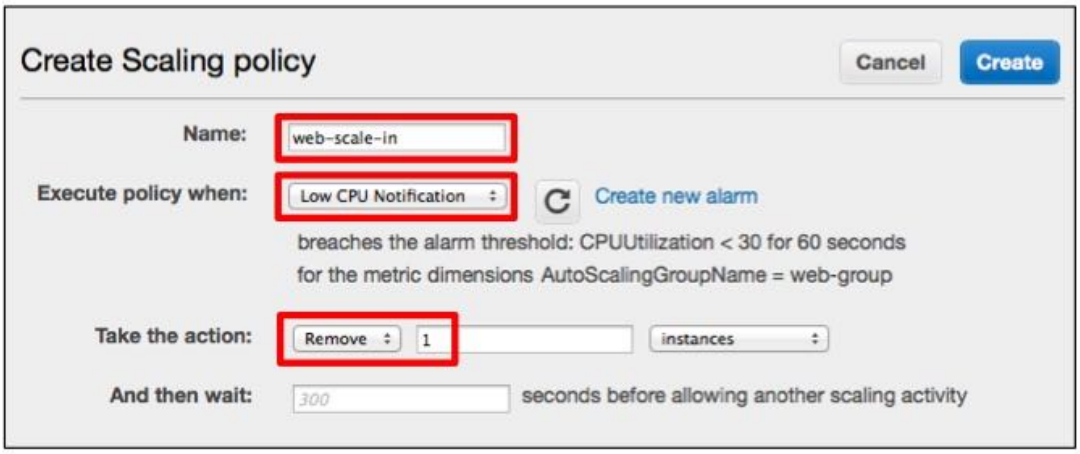
   b. Click **Create new alarm.**

   c. Enter these details:



   d. Click **Create Alarm**.

e. **Take the Action:** `Remove 1 instances`



f. Click **Create**.

Note: We strongly recommend that you configure your Auto Scaling policies to scale-out quickly and scale-in slowly. This allows the application to better respond to increased traffic loads after a scale-up event and to make more economical use of the AWS hourly billing cycle. This example in the lab is intentionally short and simplistic. From a billing perspective, it costs no more if the instance is scaled down after 59 minutes than if it is runs for 3 minutes.

Adjusting the maximum size of the Auto Scaling group

When you created your Auto Scaling group, you did not set the maximum number of instance you wanted to launch. As a result, the system left the value at its default setting of 1 instance. To see how these scaling policies work, you need to change the maximum number of instances to a larger number.

1. Click the **Details** tab. You will see that **Min** and **Max** are both set to 1.
2. Click **Edit** (on the right side of the Details tab), then:
   a. Set **Max** to: `5`
   b. Click **Save** (on the right).

Testing Auto Scaling

All pieces are in place to demonstrate Auto Scaling based on application usage:

- You have an Auto Scaling group with a minimum of 1 instance and a maximum of 5.
- You also have Auto Scaling Policies to increase and decrease the group by 1 instance when aggregate average CPU of the group is >= 50% and < 30%.

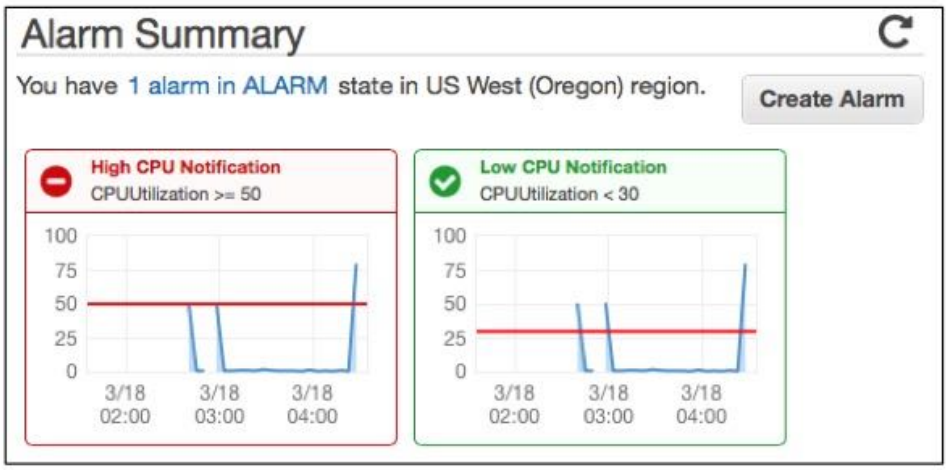
Currently, 1 instance is running because the minimum size is 1 and the group is not currently under any load. Even though CPU Utilization is < 30%, auto scaling is not removing instances because the group size is currently at its minimum of 1.

You will now access the web server to generate load and trigger a scale-out policy.

1. Click **Load Balancers** in the left panel.
2. Select the load balancer you created earlier.
3. In the **Description** tab, copy the **DNS Name (A Record)** to your clipboard. Do not copy the "(A Record)" text at the end.
4. **Open a new tab** in your web browser, **paste the DNS Name** and hit Enter. You should see something similar to the following:

5. Click the **Generate Load** button and you will see the CPU Load jump up to 100% (you may have to refresh your browser to see the CPU Load increase). This button triggers a simple background process to copy, zip, and unzip ~1GB of nothing (`/dev/zero`) for 10-20 minutes.
6. Return to the AWS Management Console and select **CloudWatch** from the **Services** menu.

You will see an **Alarm Summary** of your alarms. They might be temporarily indicating `Insufficient Data`, but this will change once the instance is generating load for a few minutes. You can always check the web server again to make sure it is under load. If not, click the **Generate Load** button again.



7. Check your email. You should receive an email notification from auto scaling informing you that a scale-up action was triggered.

8. In the AWS Management Console and select **EC2** from the **Services** menu.

9. Click **Instances** in the left panel. You will see a new instance has been added to your group.

10. Click **Auto Scaling Groups** in the left panel (at the bottom). You will notice that the **Desired** number of instances has now increased.

Name	Launch Configuration	Instances	Desired	Min	Max
web-group	Web launch configuration	2	2	1	5

A default **Cooling Down Period** of 300 seconds (5 minutes) will prevent any further scale-up or scale-down during that period. This gives Auto Scaling a chance to start/stop instances before another Scaling Policy applies.

After a few minutes, you should see your Auto Scaling group scale up, and then eventually scale back down to 1 instance. Also note that the instances were terminated in launch order, meaning that the "oldest" instances are terminated first. This allows you to roll in new changes to your application by updating your launch configuration to a newer AMI, then triggering Auto Scaling events to increase the Min size.

Viewing Auto Scaling Activities

Lastly, you can use the AWS Management Console to view a history of your auto scaling group's actions.

1. Select your Auto Scaling group.

2. Click the **Scaling History** tab.

You should see a list of events in which the Auto Scaling group added and removed EC2 instances.

Conclusion

Congratulations! You now have successfully completed the following activities:

- Creating a Launch Configuration
- Creating an Auto Scaling Group
- Activating Auto Scaling Notifications
- Creating Scaling Policies
- Testing Auto Scaling by triggering Scaling Policies

End Your Lab

1. To log out of the AWS Management Console, from the menu, click `awsstudent @ [YourAccountNumber]` and choose **Sign out** (where `[YourAccountNumber]` is the AWS account generated by *qwik*LABS):

2. Close any active SSH client sessions or remote desktop sessions.

3. Click the **End Lab** button on the *qwik*LABS lab details page.

4. When prompted for confirmation, click **OK**.

Lab 4: Creating a Batch Processing Cluster

Overview

In this lab, you use the AWS Management Console to build a basic **batch processing cluster**. You will:

- Launch and configure an **EC2 instance** that will serve as the template for future worker nodes in your batch processing cluster
- Create an **Amazon Machine Image (AMI)** from that instance
- Use **SQS** to create task queues for passing messages to your instances
- Launch an **Auto Scaling Group** of instances based on your AMI
- **Schedule work** via your task queue
- Observe the **output** in your output queue

About the batch processing cluster

The worker nodes in your cluster have a simple job: to convert some number of individual images into a single montage image. A worker node will download images from a list that you provide and will then stitch them into a composite montage using the ImageMagick tool. While this is not the most CPU-intensive job, it does require some cycles, and the larger the size and number of images you provide for each job, the more work each node will have to do.

For this lab, you will provide a newline-delimited list of image URLs. An EC2 worker node will download each image and produce an output, such as:

Topics Covered

By the end of this lab, you will be able to:

- Bootstrap an EC2 instance using User Data
- Create an AMI from a running instance
- Use the AWS Management Console to create an SQS queue
- Create an Auto Scaling Group with Scaling Policies based on an SQS queue
- Use the AWS Management Console to pass messages to, and read messages from, an SQS queue

Login to the AWS Management Console

Using qwikLABS to login to the AWS Management Console

Welcome to this self-paced lab! The first step is for you to login to Amazon Web Services.

5. To the right of the lab title, click the **Start Lab** button to launch your *qwik*LABS. If you are prompted for a token, use the one distributed to you (or the token you purchased).

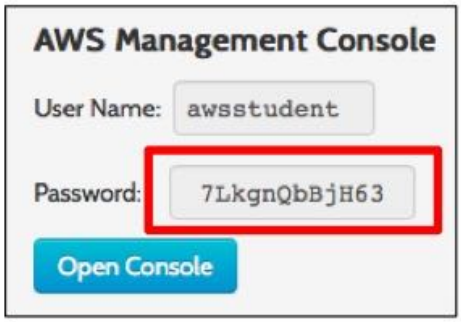
6. On the lab details page, notice the lab properties.
    a. **Duration -** The time the lab will run before automatically shutting down.
    b. **Setup Time -** The estimated time to set up the lab environment.
    c. **AWS Region** - The AWS Region in which the lab resources are created.

The AWS Region for your lab will differ depending on your location and the lab setup.

7. Click the **Download PEM/PPK** button and then:
    a. Windows users: Select **Download PPK**
    b. Mac & Linux users: Select **Download PEM**

A file will download to your computer for use later in the lab, so remember where it was saved (such as the Downloads folder):

8. In the AWS Management Console section of the *qwik*LABS page, copy the **Password** to the clipboard.



9. Click the **Open Console** button:

10. Log into the AWS Management Console using the following steps.

    a. In the **User Name** field type `awsstudent`.

    b. In the **Password** field, paste the password copied from the lab details page.

    c. Click **Sign in**.

Note: The AWS account is automatically generated by *qwik*LABS. Also, the login credentials for the `awsstudent` account are provisioned by *qwik*LABS using AWS Identity Access Management.

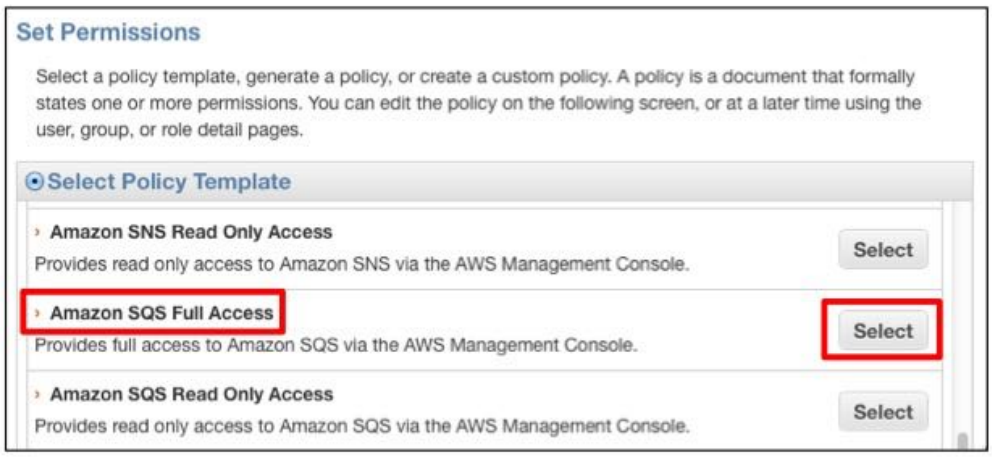
11. Return to the *qwik*LABS tab and open the **command reference text file**, located under the **Instruction** tab. You will require this file later in the lab.

Creating an IAM role

Your batch processing nodes will communicate with **Amazon SQS** to receive processing instructions and will then store results in **Amazon S3**. Start by creating an **IAM Role** that grants access to both SQS and S3. This role when then be assigned to your EC2 instances.

1. Return to the AWS console and click **IAM**.

2. Click **Roles** in the left panel.
3. Click **Create New Role**, then:
    a. **Role Name:** `BatchProcessing`
    b. Click **Continue**
4. In **Select Role Type**, select **Amazon EC2**.
5. Locate the **Amazon SQS Full Access** policy and click **Select** (you will need to scroll down to find it, or use the browser's Find command):



A policy document will be displayed.

6. Click **Continue** and then click **Create Role**.

You now need to add additional permissions for S3.

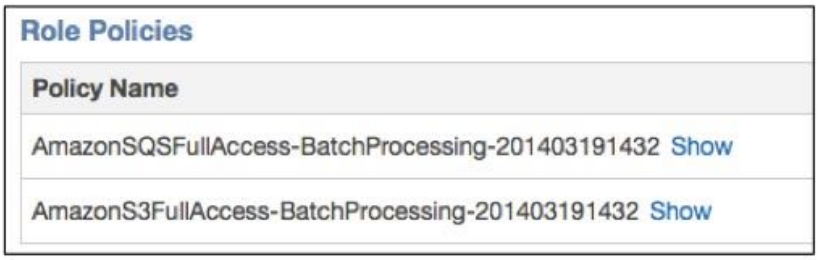
7. Select the `BatchProcessing` role that you just created.

8. Click the **Attach Role Policy** button at the bottom of the screen to attach another policy:

9. Locate the **Amazon S3 Full Access** policy and click **Select**.

10. Click **Apply Policy**.

Your Role Policies should now look similar to this:



Creating a 'Master' EC2 Instance

You can now launch an EC2 instance with a configuration script that loads ImageMagick and the batch processing software. This 'master' instance will then be used to create an Amazon Machine Image.

1. From the AWS Management Console, select **EC2** from the **Services** menu.

2. Click **Launch Instance**.

3. Locate the **Amazon Linux AMI** and click **Select**.

4. At the **Choose and Instance Type** panel, click **Next: Configure Instance Details**.

5. At the **Configure Instance Details** panel:
    a. **IAM Role:** `BatchProcessing`
    b. Expand the **Advanced Details** section. (You may need to scroll down to locate it.)
    c. **User Data:** Paste in the text from the **command reference text file**, from the section titled `Launch an EC2 Instance`. It should look similar to this:

```
#!/bin/bash

# Install ImageMagick, a Python library, and create a directory
yum install -y ImageMagick
easy_install argparse
mkdir /home/ec2-user/jobs

# Download and install the batch processing script
# The following command must be on a single line:
wget -O /home/ec2-user/image_processor.py https://us-west-2-aws-training.s3.amazonaws.com
/architecting-lab-3-creating-a-batch-processing-cluster-3.1/static/image_processor.py
```

    d. Click **Next: Add Storage**.
6. There are no modifications needed in the **Add Storage** panel. Click **Next: Tag Instance**.
7. At the **Tag Instance** panel, enter the **Value:** `Master`
8. Click **Next: Configure Security Group**.
9. At the Configure Security Group panel:
    a. **Security group name:** `BatchProcessing`
    b. **Description:** `Batch Processing`
    c. Verify there are is an existing rule for port **22** (SSH).

This will only allow access to port 22, which permits SSH connections. For this lab it is allowing access from any IP address on the Internet. Normally you will want to restrict access to the address ranges specifically required for administration.

10. Click **Review and Launch**.
11. Click **Launch**. You are presented with the **Select an existing key pair or create a new key pair** dialog.
12. Check the acknowledgement box and click **Launch Instances**.
13. Click **View Instances** (you might need to scroll down to see it).

Connecting to your instance

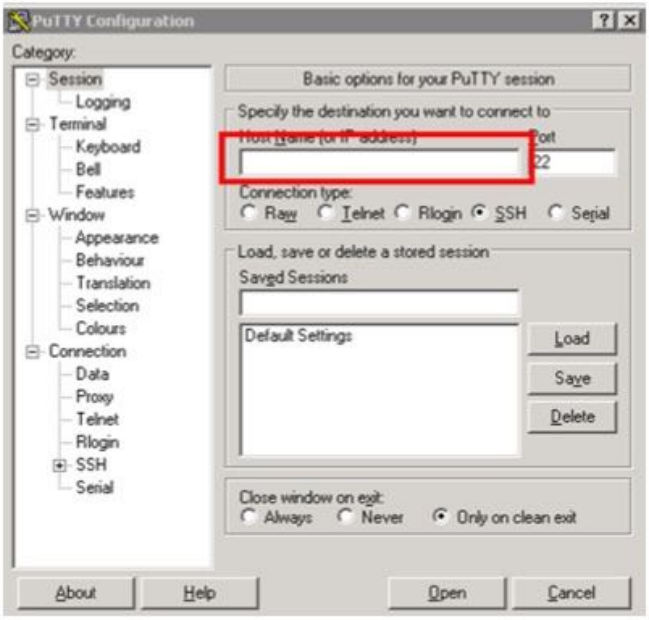
You are going to use your 'Master' instance as the basis for an Amazon Machine Image (AMI). Before doing so, you will connect to the instance via SSH to ensure that the necessary files were loaded via the User Data script.

Connecting to the EC2 instance (Windows)

Important: This section is for Windows users only. If you are using Mac or Linux, please skip ahead to the next section.

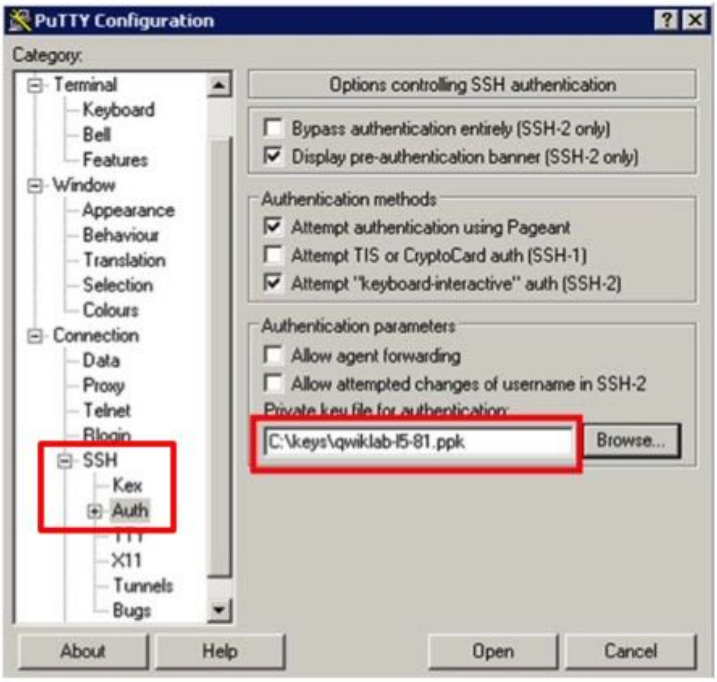
You will be connecting to the EC2 instance by using **PuTTY**, a free SSH utility. You are welcome to use an alternate SSH utility if you have one.

1. Download PuTTY from: http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe
2. Run PuTTY.
3. Copy the **Public DNS** name of your EC2 instance from the EC2 Management Console and paste it into the **Host Name** field in PuTTY:



4. Expand the SSH setting in the left panel and click **Auth**.

5. Click the **Browse** button and locate the **PPK file** that you downloaded at the start of this lab. (If desired, you can download it again from the *qwik*LABS tab in your browser.)



6. Click the **Open** button to initiate the connection. Accept any warning messages that appear.

7. **Login as:** `ec2-user`

```
login as: ec2-user
Authenticating with public key "imported-openssh-key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2013.09-release-notes/
[ec2-user@ip-172-31-13-171 ~]$ ls
image_processor.py  jobs
[ec2-user@ip-172-31-13-171 ~]$
```

8. Once connected, run the `ls` command to view the contents of your user directory.

If you see entries named `image_processor.py` and `jobs` then your instance is correctly configured. If these entries are not present, please ask your instructor for assistance.

Connecting to the EC2 instance (Mac and Linux)

Important: This section is for Mac and Linux users only. If you are using Windows, please skip ahead to the next section.

You will be connecting to the EC2 instance by using SSH.

1. Copy the **Public IP** address of your EC2 instance from the EC2 Management Console.
2. Locate the **PEM file** that you downloaded at the start of this lab. (If desired, you can download it again from the *qwik*LABS tab in your browser.)
3. Run the following commands to connect to the instance. Remember to substitute for your own PEM file (which may be in a different directory) and Public IP address:

```
$ chmod 600 Downloads/qwikLABS-L12-3456.pem
$ ssh -i Downloads/qwikLABS-L12-3456.pem ec2-user@11.22.33.44

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

https://aws.amazon.com/amazon-linux-ami/2013.09-release-notes/

[ec2-user@ip-172-31-13-171 ~]$ ls
image_processor.py  jobs
[ec2-user@ip-172-31-13-171 ~]$
```

4. Do an `ls` command to view the contents of the user directory.

If you see entries named `image_processor.py` and `jobs` then your instance is correctly configured. If these entries are not present, try again in a few minutes to allow the User Data script to execute. If they do not appear after a few minutes, please ask your instructor for assistance.

At this point, you have customized a running EC2 instance that you can use as the basis for nodes in your batch processing cluster.

Creating an AMI from your batch processing instance

In this section you will use the AWS Management Console to create an Amazon Machine Image (AMI) from the running 'Master' instance. By creating an AMI from the instance you configured, you can launch many more identically-configured instances quickly and easily.

1. Select your instance in the **EC2** Management Console.
2. From the **Actions** menu, select **Create Image**, then:
    a. **Image Name:** `Worker Image`
    b. **Image Description:** `Batch Processing worker`

c. Click **Create Image**, then click **Close**.

Your instance will automatically reboot during this process, causing your SSH session to disconnect – this is normal.

3. Click **AMIs** in the left panel to view your AMI.

Your AMI will initially show as **pending** and will eventually change to **available**.

Please continue onto the next section – there is no need to wait for your AMI to become available.

Creating two SQS task queues

In this section, you will use the AWS Management Console to create two **Simple Queuing Service (SQS)** queues to hold input and output tasks. You will eventually dispatch work via the `input` queue, and view the results provided by your worker nodes in the `output` queue.

1. In the AWS Management Console, select **SQS** from the **Services** menu.
2. Click **Create New Queue**, then:
   a. **Queue Name:** `input`
   b. **Default Visibility Timeout:** `90 seconds`
   c. Click **Create Queue**
3. Repeat the previous step to **create another queue** named: `output`

You can now put some work into the input queue. It will remain in the queue until your worker nodes are launch later in the lab. This also gives time for the queue metrics to be received in CloudWatch.

4. Select your **input** queue.
5. From the **Queue Actions** menu, select **Send a Message**.
6. Paste a list of image URLs into the message. A sample set of image URLs is available in the **command reference text file** in the section titled Dispatch Work and View Results**.** Your list will look something like this:

```
https://us-east-1-aws-training.s3.amazonaws.com/arch-static-assets/DSC01265-L.jpg
https://us-east-1-aws-training.s3.amazonaws.com/arch-static-assets/DSC01267-L.jpg
https://us-east-1-aws-training.s3.amazonaws.com/arch-static-assets/DSC01292-L.jpg
https://us-east-1-aws-training.s3.amazonaws.com/arch-static-assets/DSC01315-L.jpg
https://us-east-1-aws-training.s3.amazonaws.com/arch-static-assets/DSC01337-L.jpg
```

7. Click **Send Message** then click **Close**.

Creating an S3 bucket

In this section you will create an Amazon S3 bucket to hold the output from your worker nodes.

Your bucket name needs to be globally unique, so change the numbers (shown below) to generate a unique bucket name.

1. In the AWS Management Console, select **S3** from the **Services** menu.
2. Click **Create Bucket**, then:
   a. **Bucket Name:** `image-bucket-12345`
   (Change the numbers to create a unique bucket name)
   b. Click **Create**

You are now ready to launch worker nodes within an Auto Scaling group.

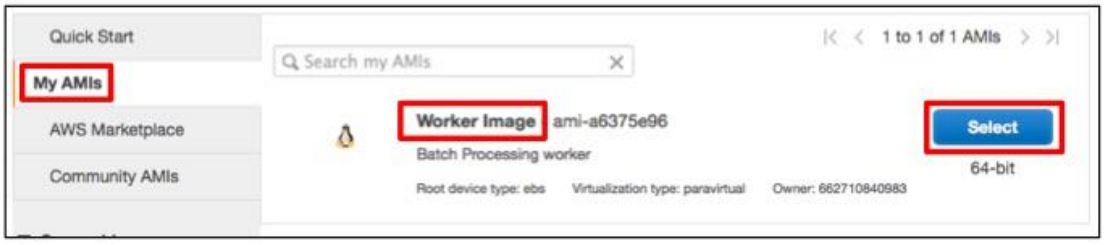
Launching worker nodes

In this section, you will create an Auto Scaling group of worker nodes to process your work. After you have successfully tested the initial node, you will add Scaling Policies to automatically expand the size of the Auto Scaling group.

Creating an auto scaling launch configuration

Your first step in creating an Auto Scaling group is to create a **Launch Configuration**. A Launch Configuration specifies details such as which AMI to use when launching new instances, which instance type to use, and what configuration scripts should be run.

1. In the AWS Management Console, select **EC2** from the **Services** menu.
2. Click **Launch Configurations** in the left panel (you may need to scroll down to see it). Since you have not yet created an Auto Scaling group, the console assumes you ultimately want to create an Auto Scaling group.
3. Click **Create Auto Scaling group**.
4. Click **Create launch configuration**.
5. Click **My AMIs** in the left panel.
6. Select your **Worker Image** AMI:



7. On the **Create Instance Type** panel, click **Next: Configure details**.
8. On the **Configure details** panel:
    a. **Name:** `Workers`
    b. **IAM Role:** `BatchProcessing`
    c. Expand the **Advanced Details** section.
    d. **User Data:** Paste in the text from the **command reference text file**, from the section titled `Launch Worker Nodes`. It will look similar to this:

```
#!/bin/sh
/usr/bin/python /home/ec2-user/image_processor.py &
```

    This User Data script will launch the two processes for the application that does image conversion. All the software to run these processes has already been loaded onto your AMI.

    e. Click **Next: Add Storage**.
9. There are no modifications needed in the **Add Storage** panel. Click **Next: Configure Security Group**.
10. At the **Configure Security Group** panel:
    a. Click **Select an existing security group**
    b. Select the `BatchProcessing` security group you created earlier
    c. Click **Review** (in the bottom-right)
11. Click **Create launch configuration**. You are presented with the **Select an existing key pair or create a new key pair** dialog.
12. Check the acknowledgement box and click **Create launch configuration**.

You are now prompted to create an Auto Scaling group that uses this Launch Configuration.

Creating an auto scaling group

Now that your launch configuration is created, create an Auto Scaling group to automatically launch your worker nodes.

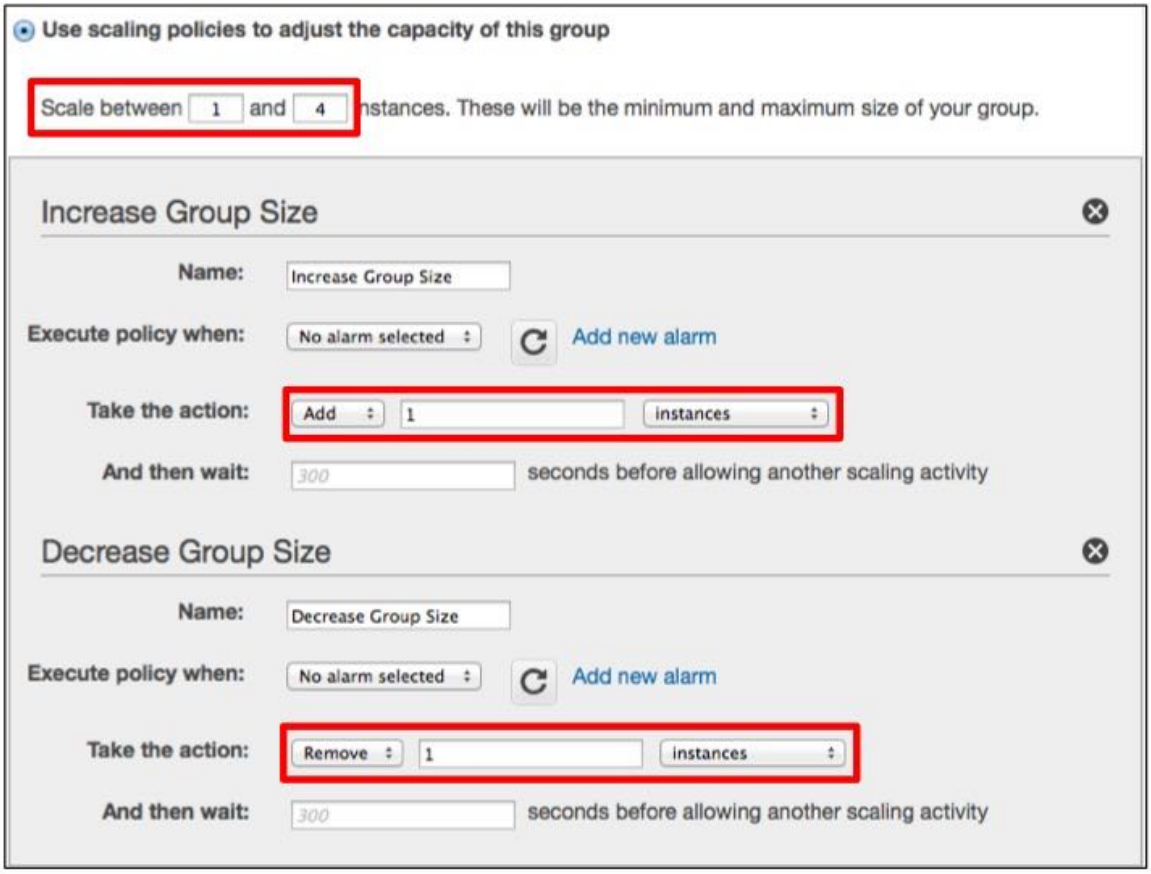
1. In the **Configure Auto Scaling group details** panel:
    a. **Group Name:** `worker-group`
    b. **Subnet:** Select at least one subnet by clicking in the Subnet box
    c. Click **Next: Configure Scaling Properties**.
2. On the **Configure scaling policies** panel (see picture below):

a. Click **Use scaling policies to adjust the capacity of this group**

b. **Scale between:** 1 and 4 instances

c. Under **Increase Group Size: Take the Action:** Add 1 instances

d. Under **Decrease Group Size: Take the Action:** Remove 1 instances



3. Click **Review**.

4. Click **Create Auto Scaling Group**.

5. Click **Close**.

6. Click **Instances** in the left panel to view your worker instances.

Your Auto Scaling group has only been configured to run a single instance at the moment, and no alarms have been attached to the Scaling Policies. Once you have verified that the worker node is functioning correctly, you will create alarms to automatically adjust the number of worker nodes.

Dispatching work and viewing results

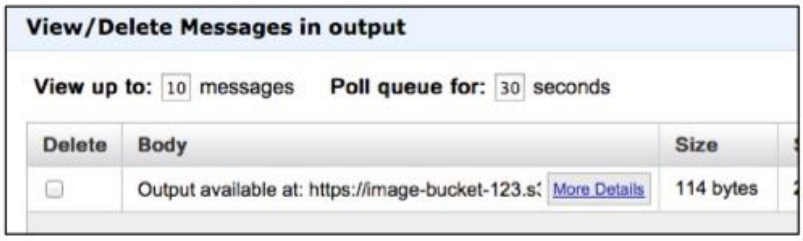
In this section you will use the SQS Management Console to put more messages in your SQS `input` queue. Your worker nodes expect a newline-delimited list of image URLs.

1. In the AWS Management Console, select **SQS** from the **Services** menu.
2. Select your **input** queue.
3. Confirm that there is 1 Message Available in your `output` queue.

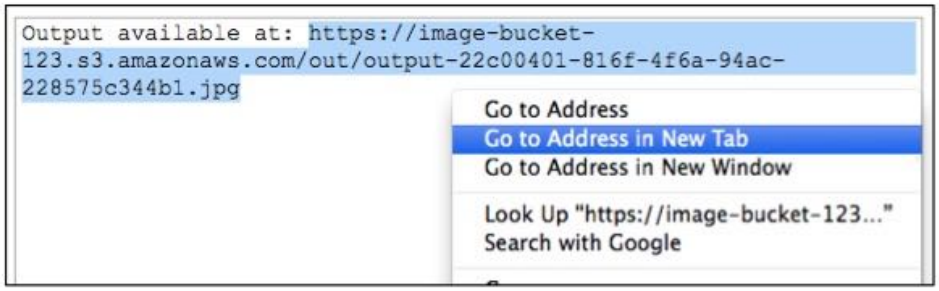
> **If there is no message in your output queue:**
> - Ensure that your queues are named `input` and `output`.
> - Ensure that your `BatchProcessing` role has granted full permissions for SQS and S3.
> - Ensure that your worker node is using the `BatchProcessing` IAM role (defined in the Launch Configuration).
> - Ensure that your worker node is running.
>
> Please consult your instructor for assistance in successfully running your worker node.

Follow these steps to view the output and open the resulting link in a browser:

4. Select your **output** queue.
5. From the **Queue Actions** menu, select **View/Delete Messages**.
6. Click **Start Polling for Messages**.
7. Find your message and click **More Details** to view the message body. The message will contain a link to the output:



8. To view the montage image, select the link, right-click, and choose **Go to Address in New Tab** (wording might vary by web browser). **Warning:** Using Ctrl-C to copy the URL does not work in this read-only text field. Use right-click to copy text or go to the link:



If the message says **Output available at: None**, then an error has occurred. Please request assistance from your instructor to debug your worker node configuration.

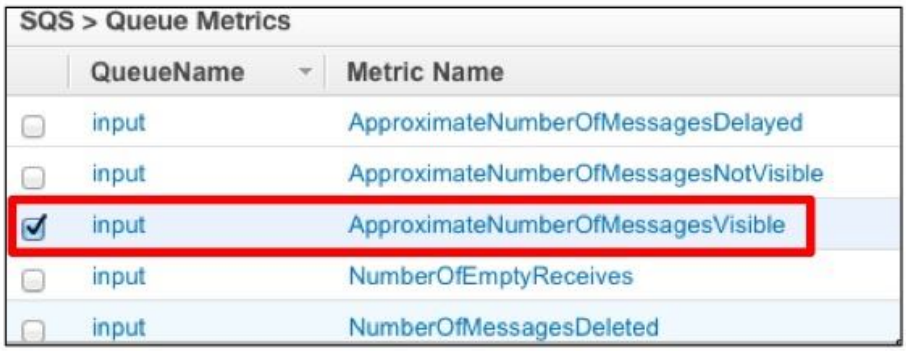
You are now ready to create CloudWatch Alarms based on the size of the input queue. To cause metrics to flow from SQS to CloudWatch, you will need to keep your queue size above zero for several minutes.

9. Send more messages to your `input` queue (10+) so that the queue is above zero for several minutes. You can use the **Send Another Message** button to send the same message multiple times.

Monitoring the cluster

You can now use CloudWatch to monitor your cluster. You will define a CloudWatch Alarm for use with Auto Scaling Policies.

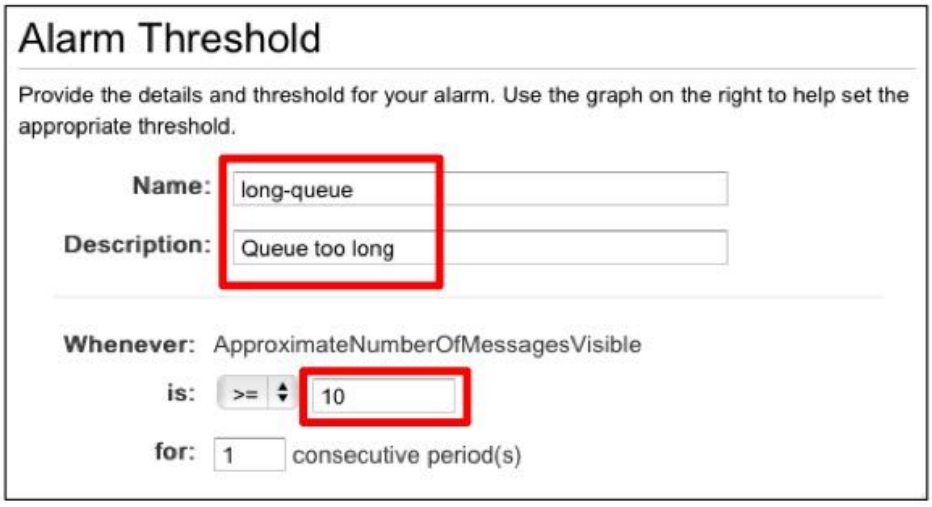
1. In the AWS Management Console, select **CloudWatch** from the **Services** menu.
2. Click **Browse Metrics**.
3. Click the **SQS Metrics** header. If the header is not visible, return to your `input` queue and ensure that there are messages queued for processing. This will trigger metrics to be sent to CloudWatch after a few minutes. The 'Visible' count may take a little bit longer to appear.
4. Select the line for:
   - **Queue Name:** `input`
   - **Metric Name:** `ApproximateNumberOfMessagesVisible`



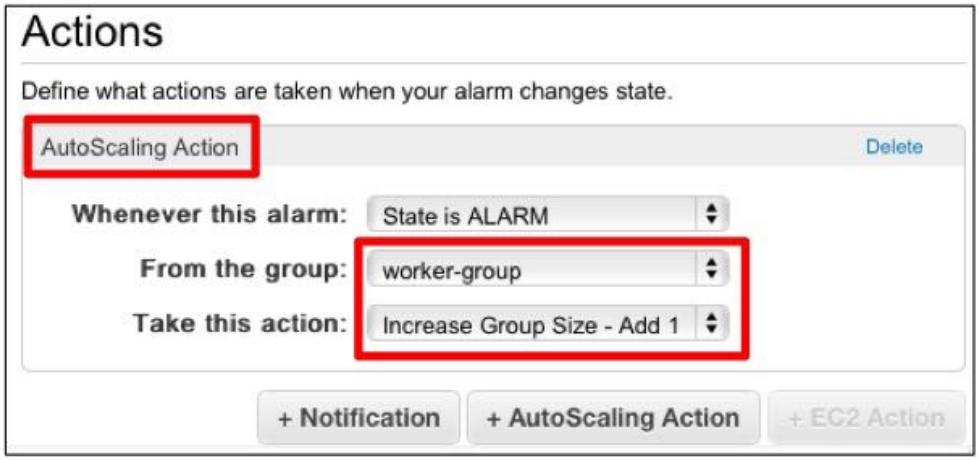
Note: This metric can take a few minutes before it becomes available. If you don't see it, wait a few minutes and browse your metrics again.

5. Click **Create Alarm** (in the bottom-right).

6. In **Alarm Threshold**, use these values:



7. In **Actions**:
   a. Delete the existing **Notification** action.
   b. Add an **AutoScaling Action**, then use these values:



8. Set **Period** to `1 minute` (bottom-right):

9. Click **Create Alarm**.

These configuration settings will automatically **add 1 instance** to your Auto Scaling group called `worker-group` whenever the `input` queue has **10 or more messages visible** within a **1 minute period**.

Testing your Auto Scaling group

You are now ready to test your Auto Scaling group!

1. Add enough messages (20+) to your `input` queue to trigger your Alarm.
2. Observe as your instances are scaled-out and the messages in your queue are processed more quickly.

It will take several minutes for your Alarm to trigger. If you see INSUFFICIENT_DATA, keep trying. It will eventually work if you keep above 10 messages in your input queue for enough minutes.

Additional Tasks (Optional)

Your Auto Scaling group is configured to scale-out, but it will not scale-in.

Try adding a **CloudWatch Alarm** to scale-in your worker nodes when the queue size is below 5.

Conclusion

Congratulations! You now have successfully:

- Bootstrapped an EC2 instance.
- Created an AMI from a running instance.
- Used the AWS Management Console to create an SQS queue.
- Used the AWS Management Console to pass messages to, and read messages from, an SQS queue.
- Created an Auto Scaling group to add EC2 instances when the number of SQS messages exceeds a certain threshold.

End Your Lab

3. To log out of the AWS Management Console, from the menu, click `awsstudent @ [YourAccountNumber]` and choose **Sign out**:

4. Close any active SSH client sessions or remote desktop sessions.

5. Click the **End Lab** button on the *qwik*LABS lab details page.

6. When prompted for confirmation, click **OK**.

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com

Additional Resources

- For more information about a specific AWS service and its pricing information, go to http://aws.amazon.com/<*service-name*>/
  e.g. http://aws.amazon.com/storagegateway/ to find out more about AWS Storage Gateway
- AWS Training and Certification

For feedback, suggestions, or corrections, please email: aws-course-feedback@amazon.com.

