Creating Amazon EC2 Instances for Linux

Self-Paced Lab Guide

Version 2.8

self-paced-lab-01-06202014

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

For all other questions, please contact us at:
https://aws.amazon.com/contact-us/aws-training/

Table of Contents

Lab Overview

Overview

This lab leads you through the steps to launch and configure your first virtual machine in the Amazon cloud. You will learn about using Amazon Machine Images to launch Amazon EC2 Instances, creating Key Pairs for SSH authentication, securing network access to Amazon EC2 Instances with Security Groups, automatically configuring Amazon EC2 Instances with bootstrapping scripts, and attaching Elastic IPs to Amazon EC2 Instances to provide static internet addresses. At the end of this lab you will have deployed a simple web server which includes an informational page to display details of your virtual web server instance.

Topics Covered

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

- Create a new Amazon EC2 server instance from an existing server template
- Create a security group to restrict access to the server's resources
- Launch the instance
- Access the instance's Linux command-line interface directly, using a Key Pair for authentication
- Associate an Elastic IP address with your Amazon EC2 instance

Technical Knowledge Prerequisites

To successfully complete this lab, you should be familiar with basic Linux server administration and comfortable using the Linux command-line tools.

Download PuTTY

If you do not already have the PuTTY client installed on your machine, you can download and then launch it from here:

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

Amazon Elastic Compute Cloud (Amazon EC2)

What is Amazon EC2?

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. It is designed to make web-scale computing easier for developers.

Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment. Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change. Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use.

Amazon EC2 enables you to increase or decrease capacity within minutes, not hours or days. You can commission one, hundreds or even thousands of server instances simultaneously. Of course, because this is all controlled with web service APIs, your application can automatically scale itself up and down depending on its needs.

You have complete control of your instances. You have root access to each one, and you can interact with them as you would any machine. You can stop your instance while retaining the data on your boot partition and then subsequently restart the same instance using web service APIs. Instances can be rebooted remotely using web service APIs. You also have access to console output of your instances.

You have the choice of multiple instance types, operating systems, and software packages. Amazon EC2 allows you to select a configuration of memory, CPU, instance storage, and the boot partition size that is optimal for your choice of operating system and application. For example, your choice of operating systems includes numerous Linux distributions, and Microsoft Windows Server.

Amazon EC2 works in conjunction with Amazon Simple Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), Amazon SimpleDB and Amazon Simple Queue Service (Amazon SQS) to provide a complete solution for computing, query processing and storage across a wide range of applications.

Amazon EC2 offers a highly reliable environment where replacement instances can be rapidly and predictably commissioned. The service runs within Amazon’s proven network infrastructure and datacenters. The Amazon EC2 Service Level Agreement commitment is 99.95% availability for each Amazon EC2 Region.

Amazon EC2 works in conjunction with Amazon Virtual Private Cloud (VPC) to provide security and robust networking functionality for your compute resources.

Amazon EC2 passes on to you the financial benefits of Amazon’s scale. You pay a very low rate for the compute capacity you actually consume.

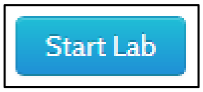
This lab guide explains basic concepts of AWS in a step by step fashion. However, it can only give a brief overview of Amazon EC2 concepts. For further information, please refer to the official Amazon Web Services Documentation for Amazon EC2 at: https://aws.amazon.com/documentation/ec2/

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 qwikLABS. If you are prompted for a token, use the one distributed to you (or the token you purchased).

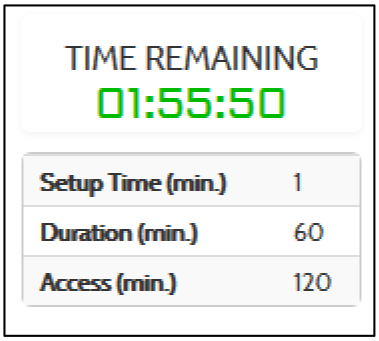


Note: A status bar shows the progress of the lab environment creation process. The AWS Management Console is accessible during lab resource creation, but your AWS resources may not be fully available until the process is complete.

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

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

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

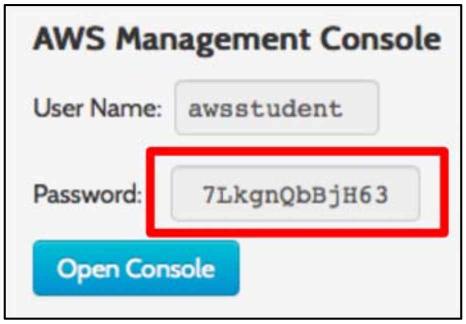


TIME REMAINING

01:55:50

Setup Time (min.)	1
Duration (min.)	60
Access (min.)	120

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



AWS Management Console

User Name: awsstudent

Password: 7LkgnQbBjH63

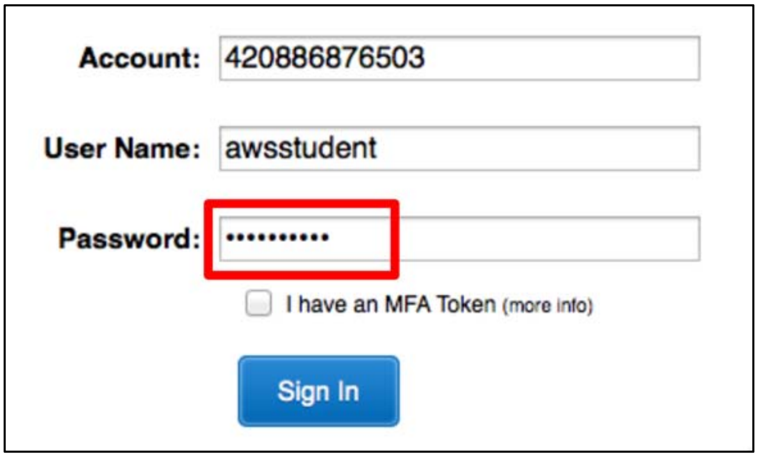
4. Click the **Open Console** button:

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**.



Account: 420886876503

User Name: awsstudent

Password: •••••••••••

□ I have an MFA Token (more info)

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

Verify Your Region in the AWS Management Console

You are now logged into the Management Console. Before proceeding, we need to verify the AWS region in which we are going to create our server.

Amazon EC2 provides the ability to place instances in multiple locations. Amazon EC2 locations are composed of Availability Zones and Regions. Regions are dispersed and located in separate geographic areas (US, EU, etc.). Availability Zones are distinct locations within a Region that are engineered to be isolated from failures in other Availability Zones and provide inexpensive, low latency network connectivity to other Availability Zones in the same Region.

By launching instances in separate Regions, you can design your application to be closer to specific customers or to meet legal or other requirements. By launching instances in separate Availability Zones, you can protect your applications from localized regional failures.

6. From the home page of Management Console, select **EC2** from the left hand navigation bar.

7. Look in the upper right corner of the Management Console and make a note of the AWS Region Name that your lab is configured for. The AWS Region was set for your lab on the qwikLAB launch page.

8. Use the chart below to determine the **Region Code**. You will normally use the code (ie. us-west-2) instead of the Region Name (i.e. “US West (Oregon) Region”) whenever your labs ask you to specify your region.

Region Name	Region Code
US East (Northern Virginia) Region	us-east-1
US West (Northern California) Region	us-west-1
US West (Oregon) Region	us-west-2
Asia Pacific (Tokyo) Region	ap-northeast-1
Asia Pacific (Singapore) Region	ap-southeast-1
Asia Pacific (Sydney) Region	ap-southeast-2
EU (Ireland) Region	eu-west-1
South America (Sao Paulo) Region	sa-east-1

More on Regions can be found here:

http://docs.aws.amazon.com/general/latest/gr/rande.html.

Using the Commands and Scripts in this Lab

IMPORTANT NOTE

If you want to copy and paste ANY code or scripts from this lab guide into an SSH session instead of manually typing them in, please make sure that you first copy and paste it to a text file and THEN copy and paste into the SSH session.

Copying text from Word documents or PDF file frequently introduces line breaks or extra (sometimes hidden) characters when you paste into SSH. We've seen more labs fail because student pasted directly from PDF into their SSH session and the commands didn't execute properly. Please use your text editor for all code and script copy and paste operations.

While we encourage students to type many of the commands themselves to help encourage learning of the core concepts, for many labs, you may find a .txt document containing the commands required for the labs on one of the “Instructions” tabs on the QwikLab lab overview page. This text file is intended to you by listed commands required for the lab so you can easily copy and paste the commands into the appropriate places during the lab. If you choose to use this text file, please download it to your computer, open it in a text editor, and read the instructions located at the beginning of the text file.

Create a New Amazon EC2 Server Instance

In this module, you will create a new Amazon EC2 server that will host a Web server visible to anyone over the Internet. In this pared-down example, our Web server will host a simple PHP script that provides some basic information about the server on which it is running.

Application Machine Images (AMIs) and Instances

Amazon EC2 provides templates known as *Amazon Machine Images (AMIs)* that contain a software configuration (for example, an operating system, an application server, and applications). You use these templates to launch an *instance*, which is a copy of the AMI running as a virtual server in the cloud.

You can launch different types of instances from a single AMI. An *instance type* essentially determines the hardware capabilities of the virtual host computer for your instance. Each instance type offers different compute and memory capabilities. Select an instance type based on the amount of memory and computing power that you need for the application or software that you plan to run on the instance. You can launch multiple instances from an AMI.

Your instance keeps running until you stop or terminate it, or until it fails. If an instance fails, you can launch a new one from the AMI.

Launch a Linux Instance

In this lab, we will launch a default Amazon Linux Instance with an Apache PHP web server installed on initialization.

9. If you aren't already on the EC Dashboard, click **EC2** on the AWS Management Console.
10. Click the **Launch Instance** button in the middle of the dashboard.
11. You first need to select an AMI. As we require a Linux instance, select the basic 64-bit **Amazon Linux AMI**, which will normally be the first option on the list.
12. Click **Select**.

When you create an instance, AWS will ask you which instance *family* you want to use. The family you choose determines how much throughput and processing cycles are available to your instance.

13. On the **Choose an Instance Type** screen, the Micro instance type, which is the smallest and lowest-cost option, should be automatically selected.
14. Click **Next: Configure Instance Details** to go to Step 3.
15. On the **Configure Instance Details** screen, scroll down and expand the **Advanced Details** section at the bottom of the screen.

Since we will be using our Amazon EC2 Instance as a Web server, we need to ensure that the Apache HTTPd server is up and running, and that the PHP programming language is installed. We can accomplish this with a simple Linux shell script. The script below installs HTTPd and PHP using the yum package manager, and then starts the HTTPd server.

16. Copy the following initialization script:

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

17. Open up a text editor and create a new text file.
18. Paste the script into a text file.
19. Now copy the data from the text file.
20. Paste the script into the **User Data** box with the **As Text** option selected. This will automatically install and start the Apache Web server when the instance is created and launches.

> **IMPORTANT:** Copy and pasting from Word documents or PDF files into SSH sessions sometimes introduces line breaks or extra characters that will cause your lab to fail. Either type all of the commands in by hand, use notepad to copy-paste all of the code in the lab, or use the commands text file (if one is included for your lab).

Note: If you are typing this instead of copy-pasting you will need to use SHIFT+RETURN for new lines in the browser form element.

21. Click **Next: Add Storage** to go to Step 4.

This screen displays which *Amazon Elastic Block Store (EBS)* volumes are attached to your image. When you launch an Amazon EC2 instance, the root device volume contains the image used to boot the instance. Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When an Amazon EBS-backed instance is launched, an EBS volume is created for each EBS snapshot referenced by the AMI. You must have at least one snapshot that denotes the root device; the others are optional and denote additional volumes to be created from other snapshots.

22. Accept the default Add Storage configuration.

23. Click **Next: Tag Instance** to accept the default storage device configuration and go to Step 5.

24. In **Step 5: Tag Instance**, choose a memorable name for your instance and type it into the **Value** field. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Use a name that you can easily recognize and remember.

25. Click, **Next: Configure Security Group** to continue on to Step 6.

Now we will create a security group. A *security group* acts as a firewall that controls the traffic allowed into a group of instances. When you launch an Amazon EC2 instance, you can assign it to one or more security groups. For each security group, you add rules that govern the allowed inbound traffic to instances in the group. All other inbound traffic is discarded. You can modify rules for a security group at any time. The new rules are automatically enforced for all existing and future instances in the group.

26. To create a security group, rename the security group with the same name you used for your instance name in the **Tag Instance** section above.

27. You can also add a Description for your security group.

By default, AWS creates a rule that allows Secure Shell (SSH) access from any IP address. It is **highly recommended** that you restrict terminal access to the ranges of IP addresses (e.g., IPs assigned to machines within your company) that have a legitimate business need to administrate your Amazon EC2 instance. Since the lab image you are using will be recycled within two hours, we will bypass this step for now.

28. Click **Add Rule** to open a new port.

29. From the **Protocol** dropdown, select **HTTP**.

This will add a default handler for HTTP that will allow requests from anywhere on the Internet. Since we want our Web server to be accessible to the general public, we can leave this rule as is without any further configuration.

30. Click **Review and Launch** to continue on to Step 7.

    Note: You may see a warning on this screen that “Your security group … is open to the world.” This is a result of not restricting SSH access to our machine, as described above. For the purposes of this lab only, you may ignore this warning.

31. Review your choices, and then click **Launch**.

32. You will receive a popup window to select a key pair or create a new one. Check the acknowledgment box.

33. Click **Launch Instances**.

Next you will see a status page, notifying you that your instances are launching.

34. Click the **View Instances** button to continue.

You will now be taken to the Instances tab of the Amazon EC2 Dashboard, which displays the list of all running Amazon EC2 instances in the currently selected region. You can see the status of your instance here.

35. Select the small square to the left of your instance. You should see a list of details and status update for your instance in the bottom pane of the console.

36. Scroll through the various details about your running instance.

37. Your instance is displayed in the list of running Amazon EC2 instances. If the **Instance State** field does not say **running**, wait a few moments and press the double-arrow icon to refresh the list.

38. When it is running, continue on to the next section.

Instructions for Windows Users: Connect to your Amazon EC2 Instance via SSH

Note: If you are running OSX or Linux, skip to the next section. This section is for Windows users only.

In this next step, we will connect to our server using the PUTTY Secure Shell (SSH) client using our server’s public DNS address.

All Amazon EC2 instances are assigned two IP addresses at launch: a *private IP address* (RFC 1918) and a *public IP address* that are directly mapped to each other through Network Address

Translation (NAT). Private IP addresses are only reachable from within the Amazon EC2 network. Public addresses are reachable from the Internet.

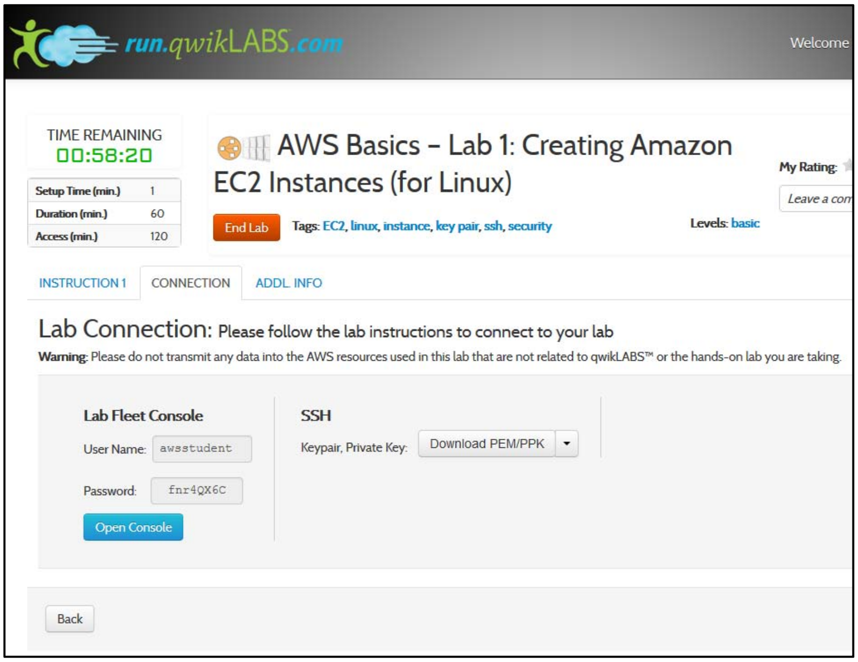
Amazon EC2 also provides an *internal DNS name* and a *public DNS name* that map to the private and public IP addresses, respectively. The internal DNS name can only be resolved within Amazon EC2. The public DNS name resolves to the public IP address outside the Amazon EC2 network, and to the private IP address within the Amazon EC2 network.

Retrieve Your Host's Public DNS Address

39. In your list of running Amazon EC2 instances, select the instance to display the instance details. Identify the **Public DNS** value in the detail pane at the bottom of the screen, and copy it to the clipboard. It will look something like: ec2-54-84-236-205.compute-1.amazonaws.com

Download your Amazon EC2 Key Pair private key file

40. Return to your open browser that has the qwikLABS lab information:



AWS Basics – Lab 1: Creating Amazon EC2 Instances (for Linux)

TIME REMAINING 00:58:20

Setup Time (min.)	1
Duration (min.)	60
Access (min.)	120

Tags: EC2, linux, instance, key pair, ssh, security

Levels: basic

Lab Connection: Please follow the lab instructions to connect to your lab

Warning: Please do not transmit any data into the AWS resources used in this lab that are not related to qwikLABS™ or the hands-on lab you are taking.

Lab Fleet Console

User Name: awsstudent

Password: fnr4QX6C

SSH

Keypair, Private Key: Download PEM/PPK

Download the qwikLABS provided Amazon EC2 Key Pair private key file in the PuTTY compatible PPK format.

41. Click on the down arrow next to the **Download PEM/PPK** drop-down.

42. Click on the **Download PPK** option.

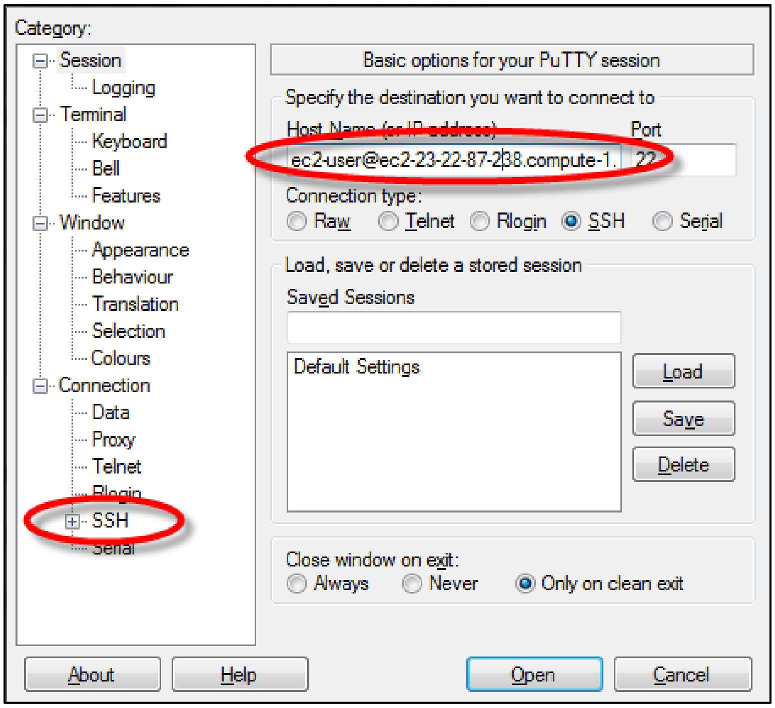
43. Save the file to your Downloads directory, or another directory of your choice.

Connect to the Amazon EC2 Instance using SSH and PuTTY.

44. Open PuTTY.exe that you downloaded at the beginning of this lab.

45. Enter *ec2-user@<your public DNS>* into the **Host Name** field (Ctrl+v).

46. Expand the **SSH** category by clicking on it.



Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): ec2-user@ec2-23-22-87-238.compute-1.

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

Saved Sessions: Default Settings

Close window on exit: ○ Always ○ Never ● Only on clean exit

47. Select the **Auth** category by clicking on it (not the + symbol next to it).

48. Click **Browse** and locate the PPK file (ending in .ppk) in your Downloads directory or whatever other location you chose.

49. Click **Open.**

50. Click **Yes** when prompted to allow a first connection to this remote SSH server. Since you are using a key pair for authentication, you will not be prompted for a password.

Common Issues

If PuTTY fails to connect to your Amazon EC2 instance, check that:

a. You have entered ‘ec2-user@<hostname>’ in Putty.
b. You have downloaded the .PPK file for this lab from qwikLAB™
c. You are using the downloaded .PPK for ‘Private key file for authentication’
d. The network you are on allows for outbound TCP connections to destination port 22

Instructions for OSX and Linux: Connect to your Amazon EC2 Instance via SSH

Note: This section is for OSX or Linux only. If you are a Windows user and just completed the section above, you can go to the Create a PHP Web Page section below.

Download your Amazon EC2 Key Pair private key file

51. Go back to your lab in qwikLABS.

Download the qwikLABS provided Amazon EC2 Key Pair private key file in the PEM format.

52. Click on the down arrow next to the **Download PEM/PPK** drop-down.

53. Click on the **Download PEM** option.

54. Save the file to your Downloads directory, or another directory of your choice.

Connect to the Amazon EC2 Instance using the OpenSSH CLI client

55. Open the Terminal application.

56. Enter the below commands substituting the path/filename for the .pem file you downloaded from qwikLABS and pasting ec2-user@<your EC2 hostname> to substitute the example below.

```
chmod 600 ~/Downloads/qwiklab-l33-5018.pem
ssh -i ~/Downloads/qwiklab-l33-5018.pem ec2-user@ec2-23-22-87-238.compute-1.amazonaws.com
```

Create a PHP Web Page on Your Linux Web Server

The AMI has already been customized with the installation of Apache and PHP from the script you entered as User Data when the instance was launched. Modify the web server by adding the following index.php file.

57. Type the following into PuTTY in order to create an index.php file at the root of your HTTP Web server's HTML document directory.

    ```
    cd /var/www/html
    sudo nano index.php
    ```

Nano is a popular, easy to use text editor for Unix systems. If you haven't used it before, don't worry – we'll walk you through the required commands.

In the following example, we will paste in some PHP code that displays information about your Amazon EC2 server instance. This information is obtained from a set of HTTP-based API calls supported by a service that is only available from your Amazon EC2 instance. For more information on this API, see the topic "Instance Metadata and User Data" (http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AESDG-chapter-instancedata.html) in the AWS Documentation.

58. Paste the code below into Nano:

    > Note: If you use copy/paste to transfer the code above from the PDF document to Nano directly, some characters may fail to copy correctly, especially some whitespace characters.
    >
    > Note: A known workaround is to copy/paste the code above into a text editor, then copy/paste the code again from notepad into Nano. This is known to get rid of any inconsistencies.

```
<?php
  $url = "http://169.254.169.254/latest/meta-data/instance-id";
  $instance_id = file_get_contents($url);
  echo "Instance ID: <b>" . $instance_id . "</b><br/>";
  $url = "http://169.254.169.254/latest/meta-data/placement/availability-zone";
  $zone = file_get_contents($url);
  echo "Zone: <b>" . $zone . "</b><br/>";
?>
```

59. Press **CTRL+O** and then **Enter** to save your document as index.php.

60. Press **CTRL+X** to exit the Nano editor.

61. Close your PuTTY or terminal window.

View Your Web Site

In this step, we will navigate to our new Web site and see the content of the page that we just created.

62. Return to your list of Amazon EC2 instances in the AWS Management Console.

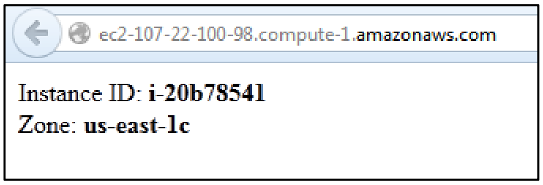
63. Select the small square to the left of your instance. You should see a list of details and status update for your instance in the bottom pane of the console.

64. Identify the **Public DNS** value in the detail pane at the bottom of the screen, and copy it to the clipboard. It will look something like: ec2-54-84-236-205.compute-1.amazonaws.com

65. Open a new browser window.

66. Paste the DNS name of your instance into your browser and connect to the server.

If successful, you will see your instance ID and Zone appear, similar to the following:



Instance ID: **i-20b78541**
Zone: **us-east-1c**

Assign a Fixed IP Address

AWS offers *Elastic IP addresses (EIPs)*, which are static IP addresses designed for dynamic cloud computing. An EIP is associated with your account, not a particular instance. You control addresses associated with your account until you choose to explicitly release them. Since EIPs are implemented as Network Address Translation (NAT) addresses that operate at a regional level, they work across Availability Zones within a single region.

Let’s assign an EIP to your instance.

67. Return to the AWS Management Console.

68. Select the **Elastic IPs** section in the navigation column on the left hand side of the **EC2** Dashboard.

69. On the Elastic IPs page, click **Allocate New Address.**

70. You will see a confirmation prompt. Click **Yes, Allocate.**

71. You will see a confirmation that a new address request has succeeded. Click **Close.**

72. Right-click on the allocated IP Address and select **Associate Address**.

73. Click in the **Instance** field, and a drop down window should appear that will list your running instance.

74. Click on the name of your instance in the drop down window (it should be the only instance listed).

75. Click **Associate**.

The Elastic IP address has been associated with your instance.

76. Return to the **Instance** detail page by clicking on **Instances** on the left side of the screen.

77. Select the small square to the left of your instance. You should see a list of details and status update for your instance in the bottom pane of the console.

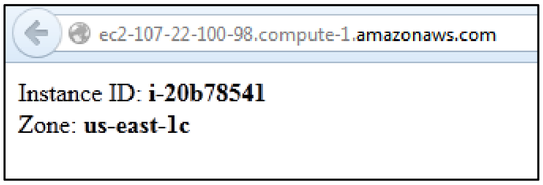
78. On the Description Tab at the bottom of the screen, find the **Elastic IP** address for your instance.

79. Copy the **Elastic IP** address for your instance.

80. Open a new browser window.

81. Paste the Elastic IP address for your instance into your browser and connect to the server.

82. If successful, you will see your instance ID and Zone appear, <u>similar</u> to the following:



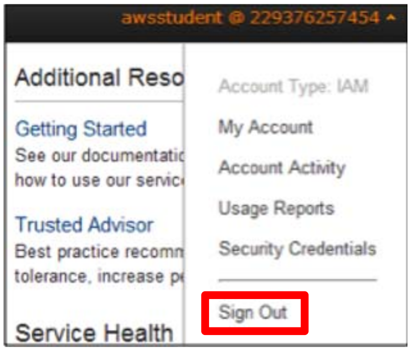
ec2-107-22-100-98.compute-1.amazonaws.com

Instance ID: **i-20b78541**
Zone: **us-east-1c**

End Your Lab

83. Return to the AWS Management Console.

84. From the menu on the upper right of the screen, click **`awsstudent @ [YourAccountNumber]`** and choose **Sign out** (where **`[YourAccountNumber]`** is the AWS account generated by qwikLABS):

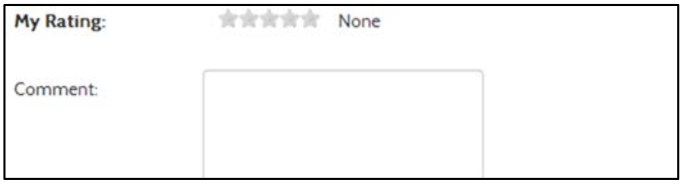


85. Close any active SSH client sessions or remote desktop sessions.

86. Click the **End Lab** button on the qwikLABS lab details page.

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

88. For **My Rating**, rate the lab (using the applicable number of stars), optionally type a **Comment**, and click **Submit**.



Note: The number of stars indicates the following:

- 1 star = Very dissatisfied
- 2 stars = Dissatisfied
- 3 stars = Neutral
- 4 stars = Satisfied
- 5 stars = Very satisfied.

You may close the dialog if you do not wish to provide feedback.

Conclusion

Congratulations! You now have successfully:

- Learned about the basic concepts and terminology of the Amazon Elastic Compute Cloud (EC2) service;
- Created your own Amazon EC2 server instance running Linux in the AWS cloud;
- Modified it to run a web server with a page that displays machine-specific information; and
- Assigned a fixed public IP address (Elastic IP) to your instance.

What Should I Do Next?

The following labs will help you understand how to leverage other features of AWS that add additional functionality to your AWS implementation. You can find these labs at http://run.qwiklabs.com.

To learn how to:

- Create a Windows server instance:
  - Creating Amazon EC2 Instances for Windows
- Create and attach additional storage to your Amazon EC2 instance:
  - Amazon Elastic Block Store (EBS)
- Use AWS Elastic Load Balancer (ELB) to balance Web traffic between two or more instances:
  - Elastic Load Balancing (ELB)
- Enable automatic creation of new Amazon EC2 instances during periods of heavy load:
  - Auto Scaling for Linux
- Create a new server instance by bidding on instance pricing:
  - Launching Amazon EC2 Spot Instances
- Add a relational database to your Virtual Private Cloud:
  - Using Amazon RDS for Applications

Additional Resources

- For more information about Amazon EC2, and Amazon EC2 pricing, go to the http://aws.amazon.com/ec2/
- http://aws.amazon.com/training/

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