AWS Basics:

Creating EC2 Instances (with Linux)

Table of Contents

Introduction

Objectives

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
- 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.

Prerequisites

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

Background

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 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Amazon Machine Images (AMIs)

An Amazon Machine Image (AMI) is a special type of pre-configured operating system and virtual application software which is used to create a virtual machine within the Amazon Elastic Compute Cloud (EC2). It serves as the basic unit of deployment for services delivered using EC2.

An AMI contains all information necessary to boot an Amazon EC2 instance with your software. An AMI is like a template of a computer's root volume. For example, an AMI might contain the software to act as a web server (Linux, Apache, and your web site) or it might contain the software to act as a Hadoop node (Linux, Hadoop, and a custom application). You launch one or more instances from an AMI. An instance might be one web server within a web server cluster or one Hadoop node.

Instance Types and Families

You can launch different types of instances from a single AMI. An instance type essentially determines the hardware of the host computer used 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 a single AMI.

Amazon Elastic Compute Cloud (Amazon EC2) instance types are grouped into the general families described in the following table.

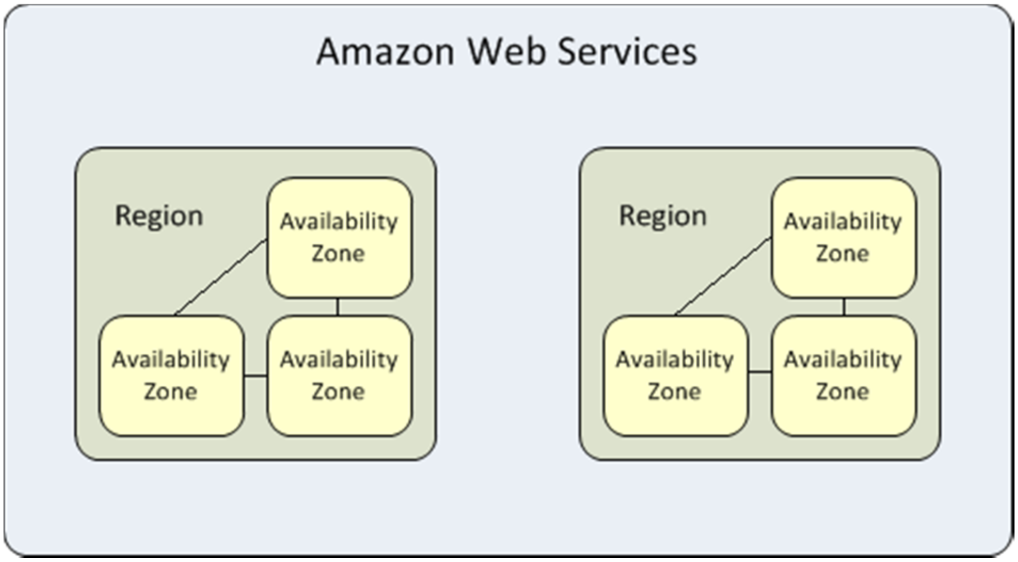
Family	Description
Micro	Provide a small amount of consistent CPU resources and enable you to burst CPU capacity when additional cycles are available. They're well-suited for lower throughput applications and websites that consume significant compute cycles periodically.
Standard	Have memory-to-CPU ratios suitable for most general-purpose applications.
Cluster Compute	Have a very large amount of CPU coupled with increased networking performance. They're well-suited for High Performance Compute (HPC) applications and other demanding network-bound applications.

Family	Description
High CPU	Have proportionally more CPU resources than memory (RAM). They're well-suited for compute-intensive applications.
High I/O	Provide tens of thousands of low-latency, random I/O operations per second (IOPS) to an application. They're well-suited for NoSQL databases, clustered databases, and OLTP (online transaction processing) systems.
High Memory	Have proportionally more memory resources. They're well suited for high-throughput applications, such as database and memory caching applications.
High Storage	Provide very high storage density and high sequential read and write performance per instance. They are well-suited for data warehousing, Hadoop/MapReduce, and parallel file systems.
Cluster GPU	Provide general-purpose graphics processing units (GPUs), with proportionally high CPU and increased network performance for applications that benefit from highly parallelized processing. They're well-suited for HPC applications as well as rendering and media processing applications.
High-Memory Cluster	Have large amounts of memory coupled with high CPU and network performance. These instances are well suited for in-memory analytics, graph analysis, and scientific computing applications.

Regions and Availability Zones (AZs)

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones. Amazon EC2 provides you the ability to place resources such as instances In multiple locations. Resources aren't replicated across regions unless you do so specifically.

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.



Amazon Simple Storage Service (S3)

Amazon S3 is storage for the Internet. It is designed to make web-scale computing easier for developers. Amazon S3 provides a simple web services interface that can be used to store and retrieve any amount of data, at any time, from anywhere on the web. It gives any developer access to the same highly scalable, reliable, secure, fast, inexpensive infrastructure that Amazon uses to run its own global network of web sites. The service aims to maximize benefits of scale and to pass those benefits on to developers.

Amazon Elastic Block Store (EBS)

Amazon Elastic Block Store (EBS) provides block level storage volumes for use with Amazon EC2 instances. Amazon EBS volumes are network-attached, and persist independently from the life of an instance. Amazon EBS provides highly available, highly reliable, predictable storage volumes that can be attached to a running Amazon EC2 instance and exposed as a device within the instance. Amazon EBS is particularly suited for applications that require a database, file system, or access to raw block level storage.

Instance Store and EBS-Backed Instances

When you launch an Amazon EC2 instance, the root device volume contains the image used to boot the instance. When we introduced Amazon EC2, all AMIs were backed by Amazon EC2 instance store, which means the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3. After we introduced Amazon EBS, we introduced AMIs that are backed by Amazon EBS. This means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot. You can choose between AMIs based by Amazon EC2 instance store and AMIs backed by Amazon EBS. We recommend that you use AMIs backed by Amazon EBS, because they launch faster and use persistent storage.

Instances that use instance store for the root device automatically have instance store volumes available, with one serving as the root device volume. When an instance is launched, the image that is used to boot the instance is copied to the root volume. Any data on the instance store volumes persists as long as the instance is running and is deleted when the instance fails or terminates.

Instances that use Amazon EBS for the root device automatically have an Amazon EBS volume attached. When you launch an Amazon EBS-backed instance, we create an Amazon EBS volume for each EBS snapshot referenced by the AMI you use. You can optionally use other Amazon EBS volumes or instance store volumes.

Amazon Virtual Private Cloud (VPC)

Amazon Virtual Private Cloud (Amazon VPC) enables you to launch Amazon Web Services (AWS) resources into a virtual network that you've defined. This virtual network closely resembles a traditional network that you'd operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC. You can configure your VPC; you can select its IP address range, create subnets, and configure route tables, network gateways, and security settings.

EC2-Classic and EC2-VPC

There are two supported platforms into which you can launch instances: EC2-Classic and EC2-VPC.

Your AWS account is capable of launching instances either into both platforms or only into EC2-VPC, on a region by region basis. If you can launch instances only into EC2-VPC, we create a default VPC for you. Then, when you launch an instance, we launch it into your default VPC, unless you create a non-default VPC and specify it when you launch the instance.

A default VPC combines the benefits of the advanced features provided by EC2-VPC with the ease of use of EC2-Classic. If you have a default VPC and don't specify a subnet when you launch an instance, the instance is launched into your default VPC. You can launch instances into your default VPC without needing to know anything about Amazon VPC.

Security Groups

A security group acts as a firewall that controls the traffic allowed to reach one or more instances. When you launch an instance, you assign it one or more security groups. You add rules to each security group that control traffic for the instance. You can modify the rules for a security group at any time; the new rules are automatically applied to all instances to which the security group is assigned.

Security groups for EC2-VPC have additional capabilities that aren't supported by security groups for EC2-Classic.

Public and Private IP Addresses

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 the private IP address within the Amazon EC2 network.

Amazon EC2 Instance IP Addressing

We provide your instances with IP addresses and DNS hostnames. These can vary depending on whether you launched the instance in the EC2-Classic platform or in a virtual private cloud (VPC).

When you launch an instance, we allocate a private IP address for the instance using DHCP. Private IP addresses are not reachable from the Internet. Each instance that you launch into a VPC has a default network interface. The network interface specifies the primary private IP address for the instance. If you don't select a primary private IP address, we select an available IP address in the subnet's range.

Each instance is provided an internal DNS hostname that resolves to the private IP address of the instance in EC2-Classic or your VPC. We can't resolve the DNS hostname outside the network that the instance is in.

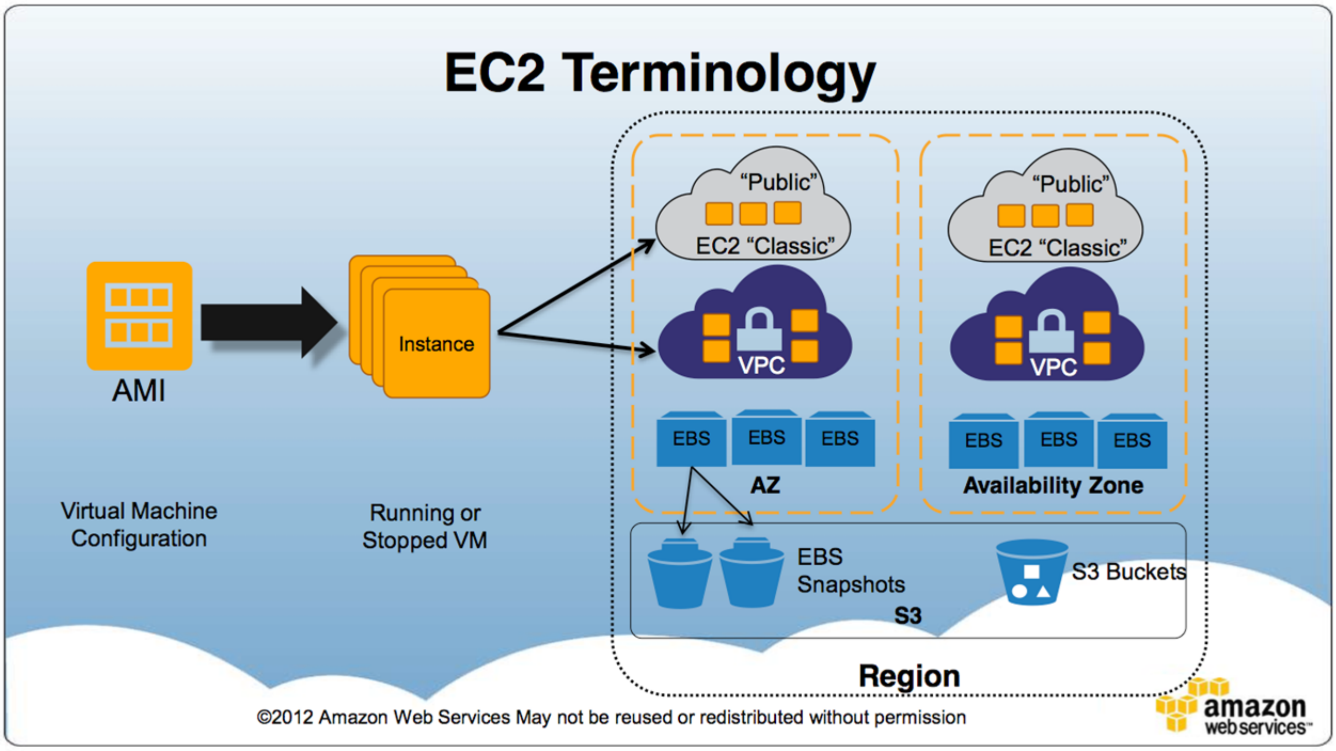
When you launch an instance in EC2-Classic or a default subnet in EC2-VPC, we allocate a public IP address for the instance. We provide each instance that has a public IP address with an external DNS hostname. We resolve an external DNS hostname to the public IP address of the instance outside the network of the instance, and to the private IP address of the instance from within the network of the instance.

Elastic IP Addresses (EIPs)

An Elastic IP address (EIP) is a static IP address designed for dynamic cloud computing. With an EIP, you can mask the failure of an instance by rapidly remapping the address to another instance. Your EIP is associated with your AWS account, not a particular instance, and it remains associated with your account until you choose to explicitly release it. There's one pool of EIPs for use with the EC2-Classic platform and another for use with your VPC. You can't associate an EIP that you allocated for use with a VPC with an instance in EC2-Classic, and vice-versa

EC2 Overview Diagram

The following diagram provides an overview of the concepts and terminology introduced above as well as their relationships:



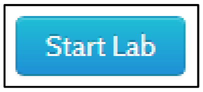
Additional Information

This lab guide gives a brief overview of Amazon EC2 concepts. For additional information, please refer to the official Amazon Web Services Documentation for EC2 at: https://aws.amazon.com/documentation/ec2/

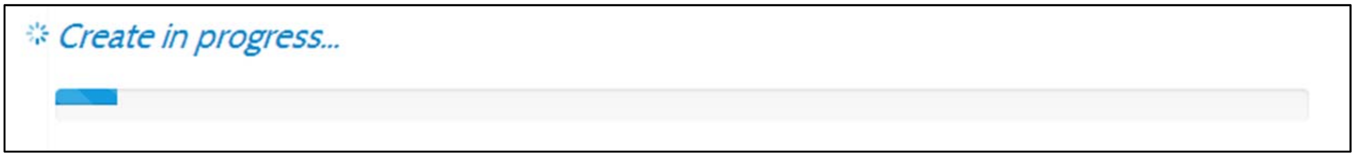
Hands-On Exercise

Start your *qwikLAB™*

1. Start your *qwikLAB™*
   Use the 'Start Lab' button to start your lab.
   (Hint: If you are prompted for a token, please use one you've been given or have purchased.)



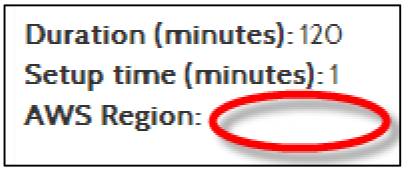
You will see the lab creation in progress.


Create in progress...

2. Note a few properties of the lab.

   a. **Duration -** The time the lab will run for before shutting itself down.
   b. **Setup Time -** The estimated lab creation time on starting the lab.
   c. **AWS Region** - The AWS Region the lab resources are being created in.

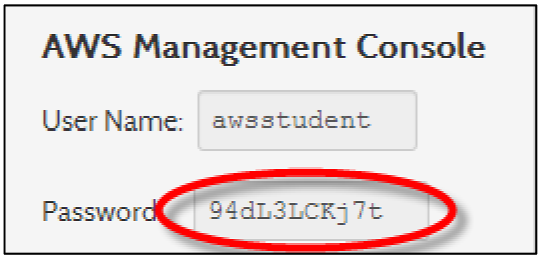
3. Note the AWS Region set for your lab in *qwikLAB™*


Duration (minutes): 120
Setup time (minutes): 1
AWS Region:

2)

4. Copy the Password provided.

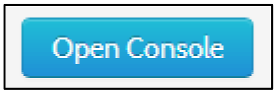
   a. Hint: selecting the value shown and using Ctrl+C works best


AWS Management Console

User Name: awsstudent

Password: 94dL3LCKj7t

5. Click the 'Open Console' button.



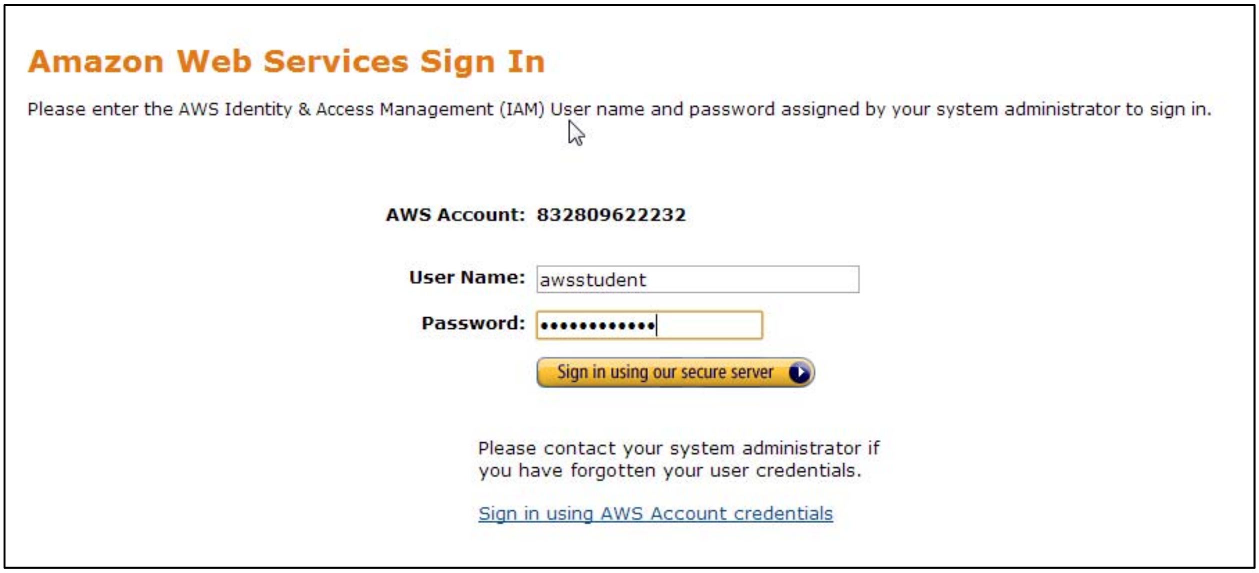
6. Make sure that you are not logged into any other instances of the AWS console (in a student account or your own account), as this may cause conflicts when you open the console and log in below for this lab.

7. Login to the AWS Management Console

   Enter the User Name ‘**awsstudent**’ and paste the password you copied from the lab details in *qwikLAB™* into the Password field.

   Click on the ‘Sign in using our secure server’ button.

   In this step you logged into the AWS Management Console using login credentials for a user provisioned via AWS Identity Access Management in an AWS account by *qwikLAB™*.



Amazon Web Services Sign In

Please enter the AWS Identity & Access Management (IAM) User name and password assigned by your system administrator to sign in.

AWS Account: 832809622232

User Name: awsstudent

Password: ••••••••••••

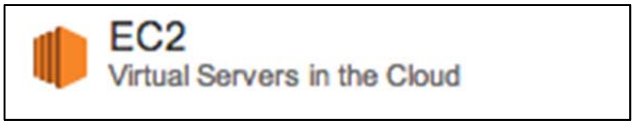
Sign in using our secure server

Please contact your system administrator if you have forgotten your user credentials.

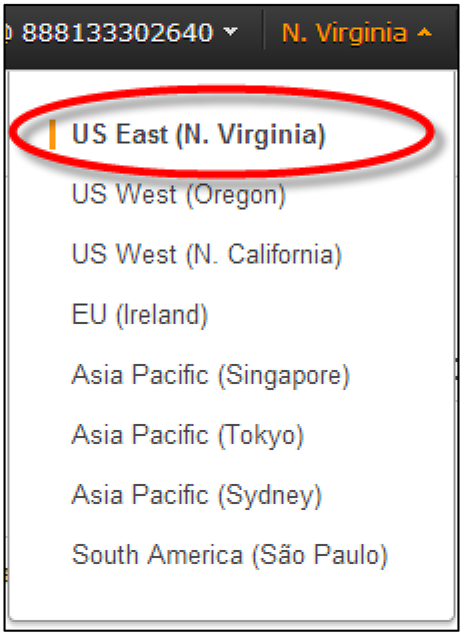
Sign in using AWS Account credentials

AWS Management Console

1. Select "EC2" from the Console Home



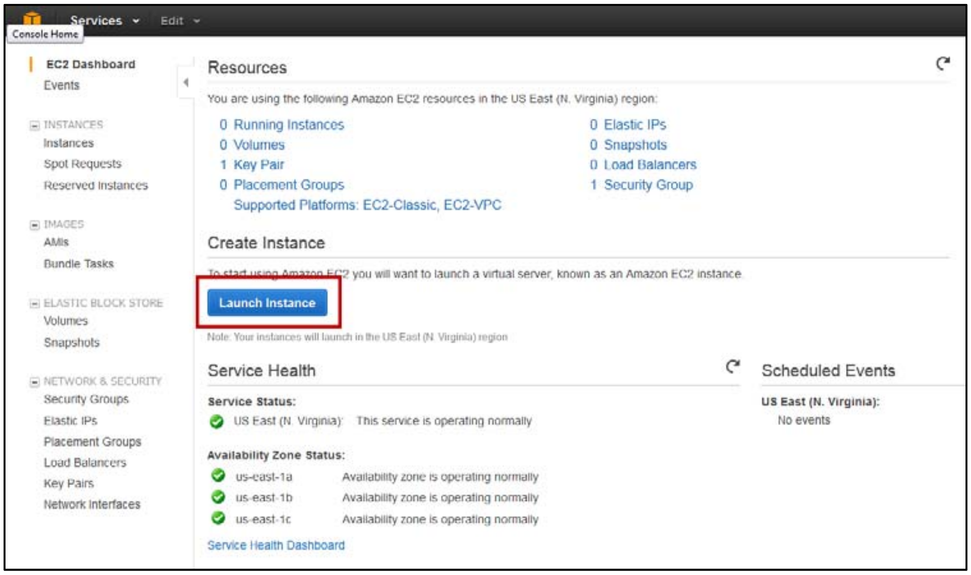
2. Select or confirm that the same AWS Region that you saw in your QwikLab lab screen is already set in the AWS Management Console



Launch a Linux Instance

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

1. Click on **Launch Instance**:



Resources

You are using the following Amazon EC2 resources in the US East (N. Virginia) region:

0 Running Instances
0 Volumes
1 Key Pair
0 Placement Groups
Supported Platforms: EC2-Classic, EC2-VPC

0 Elastic IPs
0 Snapshots
0 Load Balancers
1 Security Group

Create Instance

To start using Amazon EC2 you will want to launch a virtual server, known as an Amazon EC2 instance.

Launch Instance

Note: Your instances will launch in the US East (N. Virginia) region

Service Health

Service Status:
✓ US East (N. Virginia): This service is operating normally

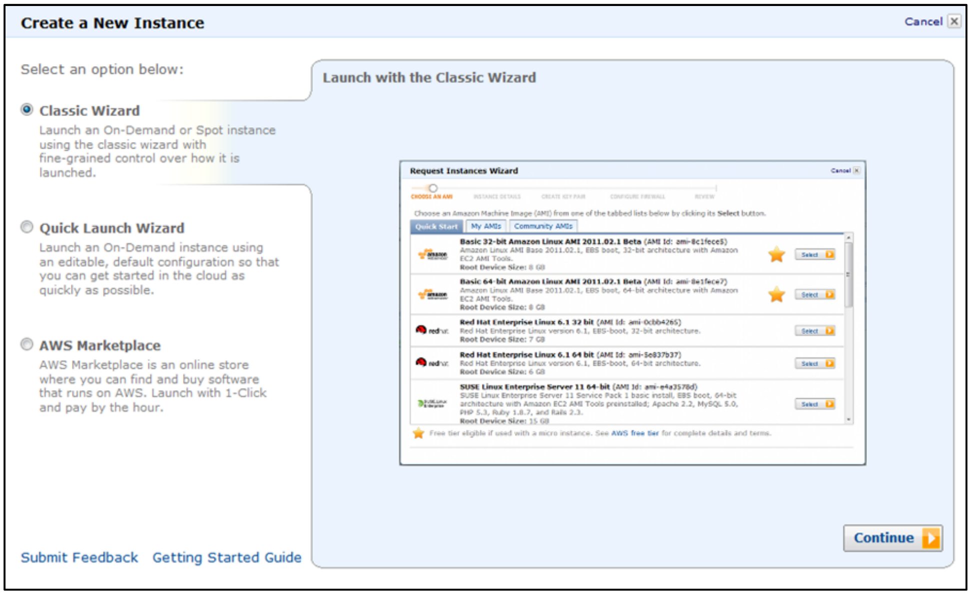
Availability Zone Status:
✓ us-east-1a Availability zone is operating normally
✓ us-east-1b Availability zone is operating normally
✓ us-east-1c Availability zone is operating normally

Service Health Dashboard

Scheduled Events

US East (N. Virginia):
No events

2. Select Classic Wizard and click Continue:



Create a New Instance

Select an option below:

● Classic Wizard
Launch an On-Demand or Spot instance using the classic wizard with fine-grained control over how it is launched.

○ Quick Launch Wizard
Launch an On-Demand instance using an editable, default configuration so that you can get started in the cloud as quickly as possible.

○ AWS Marketplace
AWS Marketplace is an online store where you can find and buy software that runs on AWS. Launch with 1-Click and pay by the hour.

Launch with the Classic Wizard

Request Instances Wizard

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Basic 32-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8c1fece5)
Amazon Linux AMI Base 2011.02.1, EBS boot, 32-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

Basic 64-bit Amazon Linux AMI 2011.02.1 Beta (AMI Id: ami-8e1fece7)
Amazon Linux AMI Base 2011.02.1, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools.
Root Device Size: 8 GB

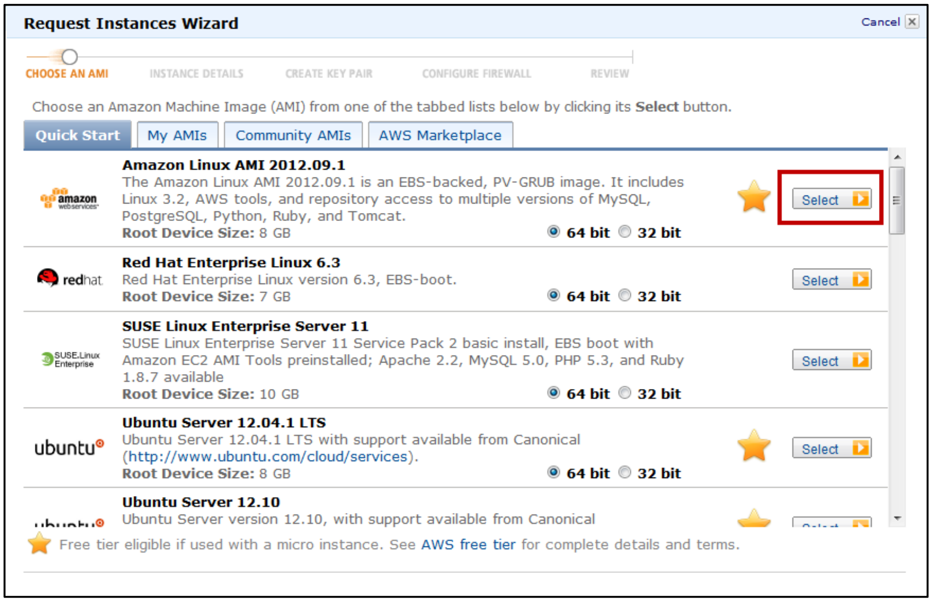
Red Hat Enterprise Linux 6.1 32 bit (AMI Id: ami-0cbb4265)
Red Hat Enterprise Linux version 6.1, EBS-boot, 32-bit architecture.
Root Device Size: 7 GB

Red Hat Enterprise Linux 6.1 64 bit (AMI Id: ami-5e837b37)
Red Hat Enterprise Linux version 6.1, EBS-boot, 64-bit architecture.
Root Device Size: 6 GB

SUSE Linux Enterprise Server 11 64-bit (AMI Id: ami-e4a3578d)
SUSE Linux Enterprise Server 11 Service Pack 1 basic install, EBS boot, 64-bit architecture with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, Ruby 1.8.7, and Rails 2.3.
Root Device Size: 15 GB

Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

3. Depending upon the resources or OS your instance requires, you may select another type. As we require a Linux instance, select the Basic 64-bit Amazon Linux AMI.



Request Instances Wizard

Choose an Amazon Machine Image (AMI) from one of the tabbed lists below by clicking its **Select** button.

Amazon Linux AMI 2012.09.1
The Amazon Linux AMI 2012.09.1 is an EBS-backed, PV-GRUB image. It includes Linux 3.2, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.
Root Device Size: 8 GB ● **64 bit** ○ **32 bit**

Red Hat Enterprise Linux 6.3
Red Hat Enterprise Linux version 6.3, EBS-boot.
Root Device Size: 7 GB ● **64 bit** ○ **32 bit**

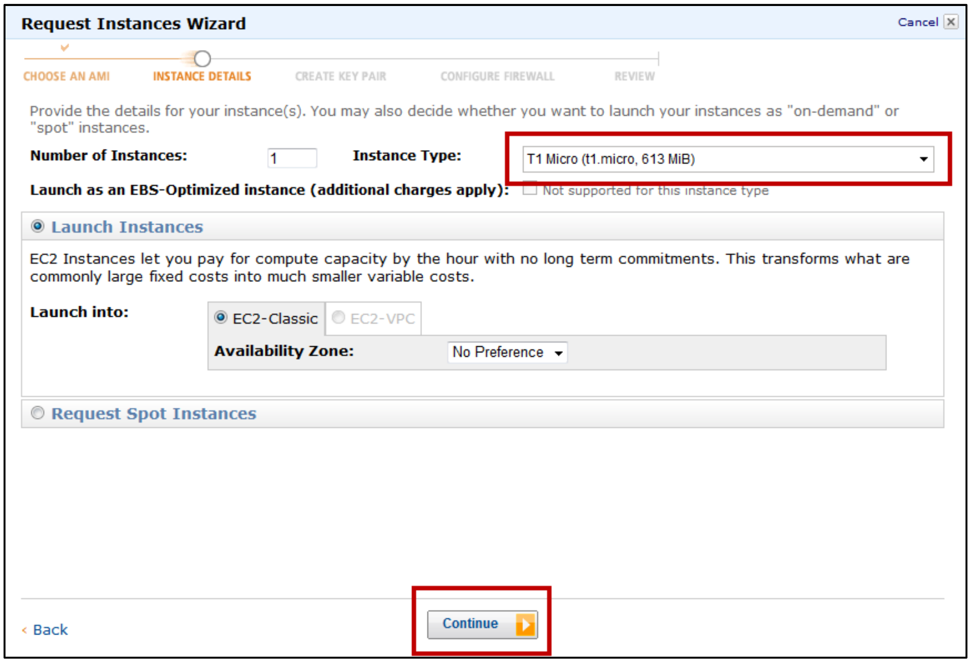
SUSE Linux Enterprise Server 11
SUSE Linux Enterprise Server 11 Service Pack 2 basic install, EBS boot with Amazon EC2 AMI Tools preinstalled; Apache 2.2, MySQL 5.0, PHP 5.3, and Ruby 1.8.7 available
Root Device Size: 10 GB ● **64 bit** ○ **32 bit**

Ubuntu Server 12.04.1 LTS
Ubuntu Server 12.04.1 LTS with support available from Canonical (http://www.ubuntu.com/cloud/services).
Root Device Size: 8 GB ● **64 bit** ○ **32 bit**

Ubuntu Server 12.10
Ubuntu Server version 12.10, with support available from Canonical

Free tier eligible if used with a micro instance. See AWS free tier for complete details and terms.

4. Select the T1 Micro (t1.micro) instance type and click Continue:



Provide the details for your instance(s). You may also decide whether you want to launch your instances as "on-demand" or "spot" instances.

Number of Instances: 1 **Instance Type:** T1 Micro (t1.micro, 613 MiB)

Launch as an EBS-Optimized instance (additional charges apply): □ Not supported for this instance type

● Launch Instances

EC2 Instances let you pay for compute capacity by the hour with no long term commitments. This transforms what are commonly large fixed costs into much smaller variable costs.

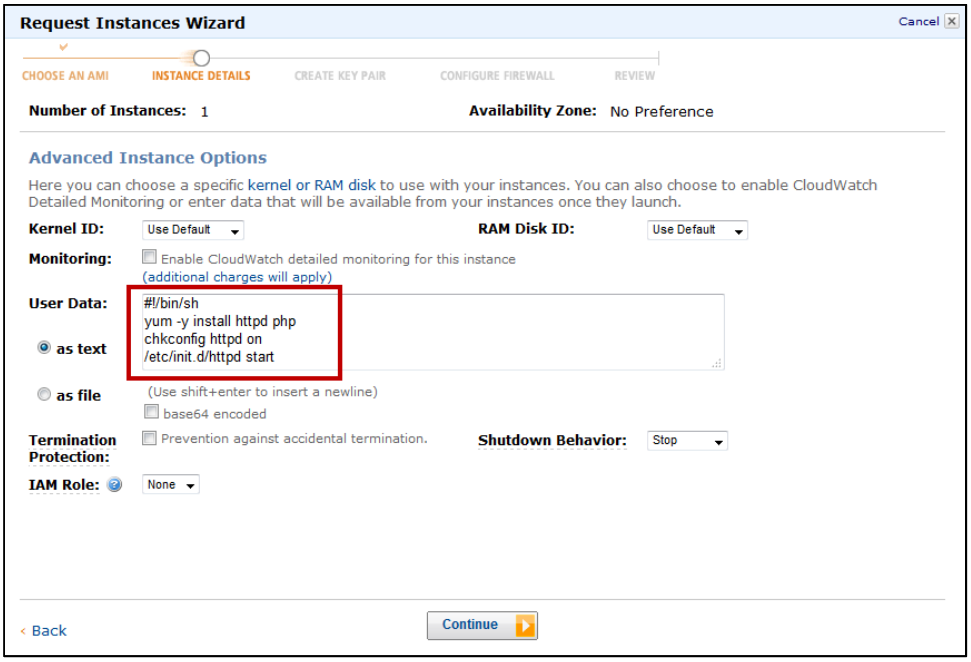
Launch into: ● EC2-Classic ○ EC2-VPC

Availability Zone: No Preference

○ Request Spot Instances

5. In the next screen, copy & paste the following initialization script (you may need to type this into a text editor and copy & paste the results) into the User Data field (this will automatically install and start Apache on launch). Then click Continue.

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



Number of Instances: 1 **Availability Zone:** No Preference

Advanced Instance Options

Here you can choose a specific kernel or RAM disk to use with your instances. You can also choose to enable CloudWatch Detailed Monitoring or enter data that will be available from your instances once they launch.

Kernel ID: Use Default **RAM Disk ID:** Use Default

Monitoring: □ Enable CloudWatch detailed monitoring for this instance (additional charges will apply)

User Data:

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

● as text

○ as file

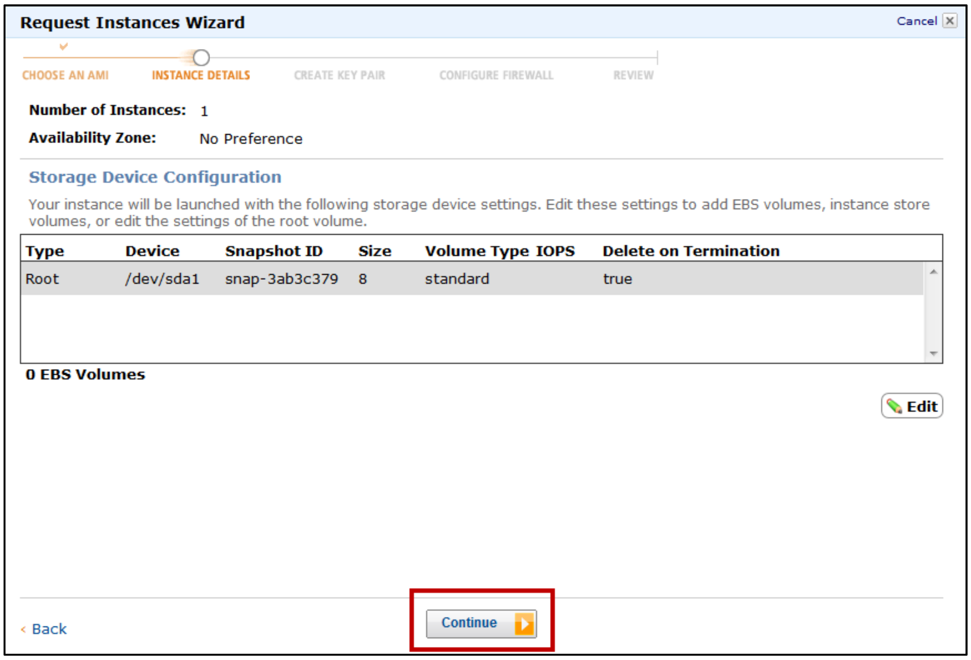
(Use shift+enter to insert a newline)

□ base64 encoded

Termination Protection: □ Prevention against accidental termination. **Shutdown Behavior:** Stop

IAM Role: None

6. Click Continue to accept the default Storage Device Configuration.



Number of Instances: 1

Availability Zone: No Preference

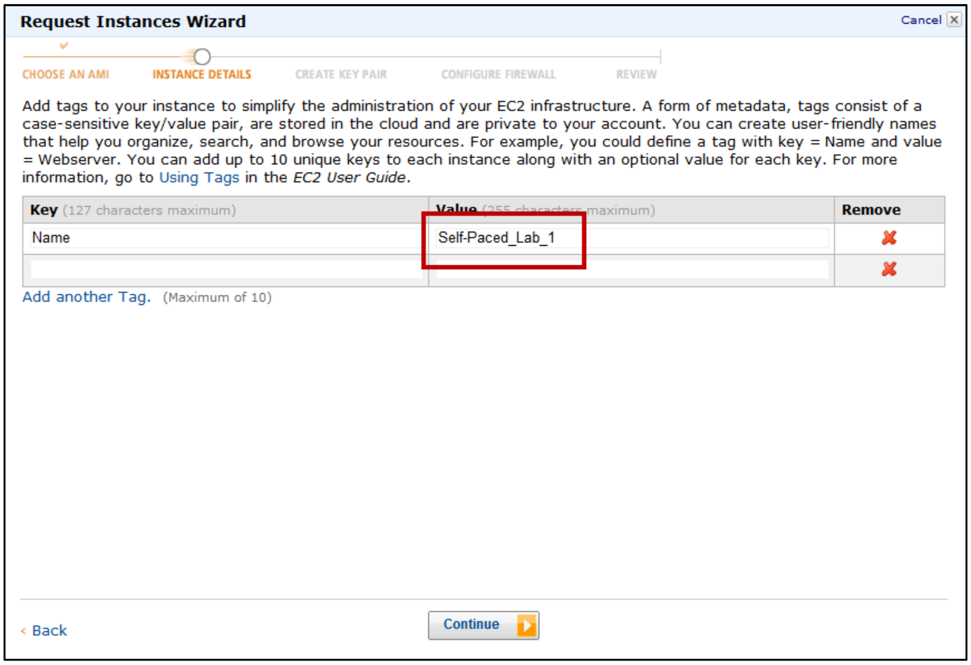
Storage Device Configuration

Your instance will be launched with the following storage device settings. Edit these settings to add EBS volumes, instance store volumes, or edit the settings of the root volume.

Type	Device	Snapshot ID	Size	Volume Type	IOPS	Delete on Termination
Root	/dev/sda1	snap-3ab3c379	8	standard		true

0 EBS Volumes

7. Next, choose a “friendly name” for your instance. 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. We named ours “Self-Paced_Lab_1”; however the only thing that matters is whether the name is meaningful to you. Put the name you choose in the Value field (see below). Then click Continue.

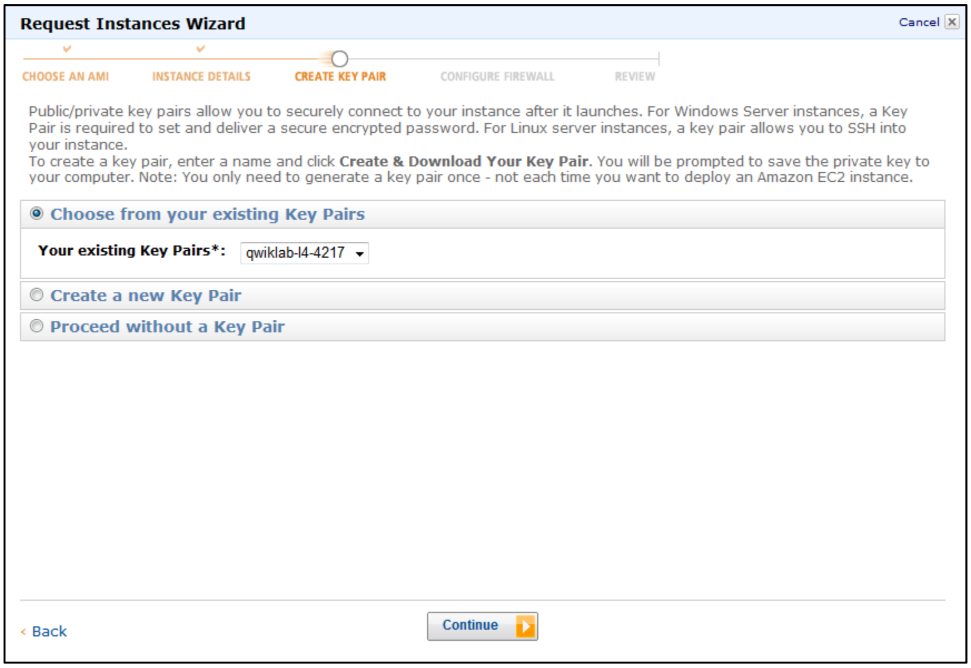


Add tags to your instance to simplify the administration of your EC2 infrastructure. A form of metadata, tags consist of a case-sensitive key/value pair, are stored in the cloud and are private to your account. You can create user-friendly names that help you organize, search, and browse your resources. For example, you could define a tag with key = Name and value = Webserver. You can add up to 10 unique keys to each instance along with an optional value for each key. For more information, go to Using Tags in the *EC2 User Guide*.

Key (127 characters maximum)	Value (255 characters maximum)	Remove
Name	Self-Paced_Lab_1	✖
		✖

Add another Tag. (Maximum of 10)

8. The Wizard will select the only EC2 Instance Key Pair in the account, the one created by qwikLAB™.
9. Click Continue.



Public/private key pairs allow you to securely connect to your instance after it launches. For Windows Server instances, a Key Pair is required to set and deliver a secure encrypted password. For Linux server instances, a key pair allows you to SSH into your instance.
To create a key pair, enter a name and click **Create & Download Your Key Pair**. You will be prompted to save the private key to your computer. Note: You only need to generate a key pair once - not each time you want to deploy an Amazon EC2 instance.

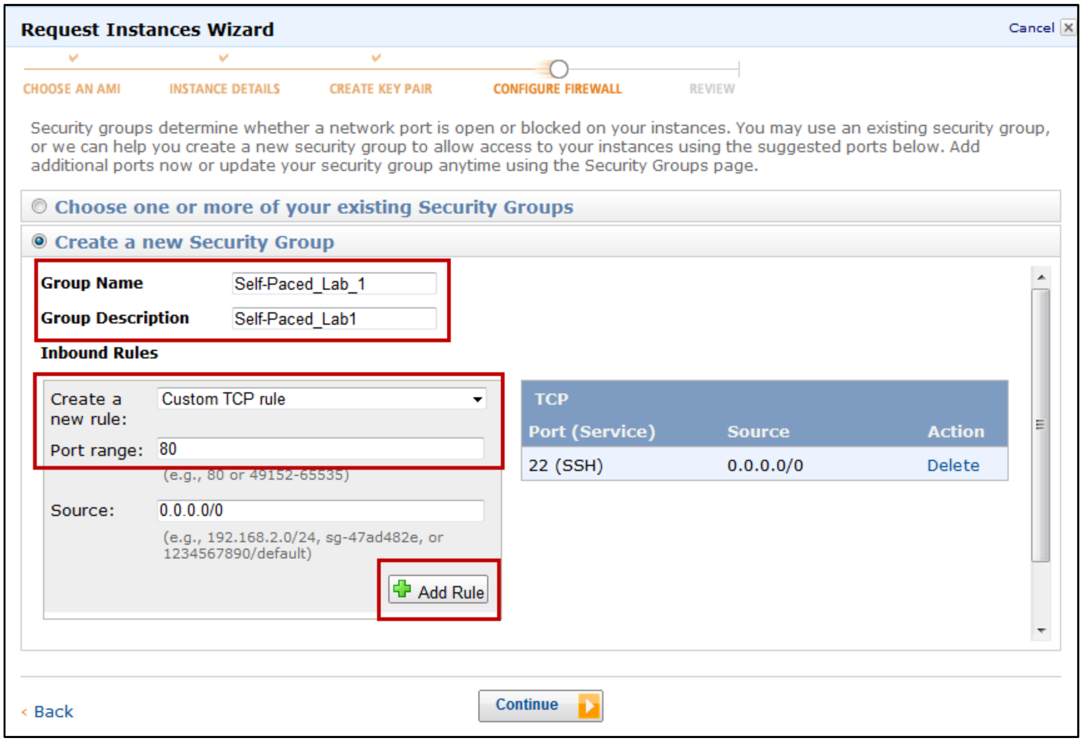
● Choose from your existing Key Pairs

Your existing Key Pairs*: qwiklab-l4-4217

○ Create a new Key Pair

○ Proceed without a Key Pair

10. Create a Security Group, which will be your firewall rules. We named this one “Self-Paced_Lab_1”. Again, the name is up to you.
    a. Make sure to open two ports: 22 (SSH) and 80 (HTTP).
    b. Port 22 is there by default.
    c. Select Custom TCP Rule, Enter 80 in port range/click add rule.



Request Instances Wizard

Security groups determine whether a network port is open or blocked on your instances. You may use an existing security group, or we can help you create a new security group to allow access to your instances using the suggested ports below. Add additional ports now or update your security group anytime using the Security Groups page.

○ Choose one or more of your existing Security Groups

● Create a new Security Group

Group Name	Self-Paced_Lab_1
Group Description	Self-Paced_Lab1

Inbound Rules

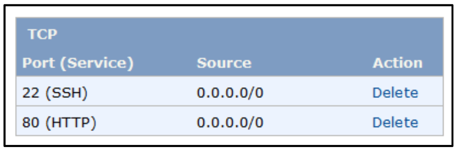
Create a new rule: Custom TCP rule

Port range: 80
(e.g., 80 or 49152-65535)

Source: 0.0.0.0/0
(e.g., 192.168.2.0/24, sg-47ad482e, or 1234567890/default)

TCP		
Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete

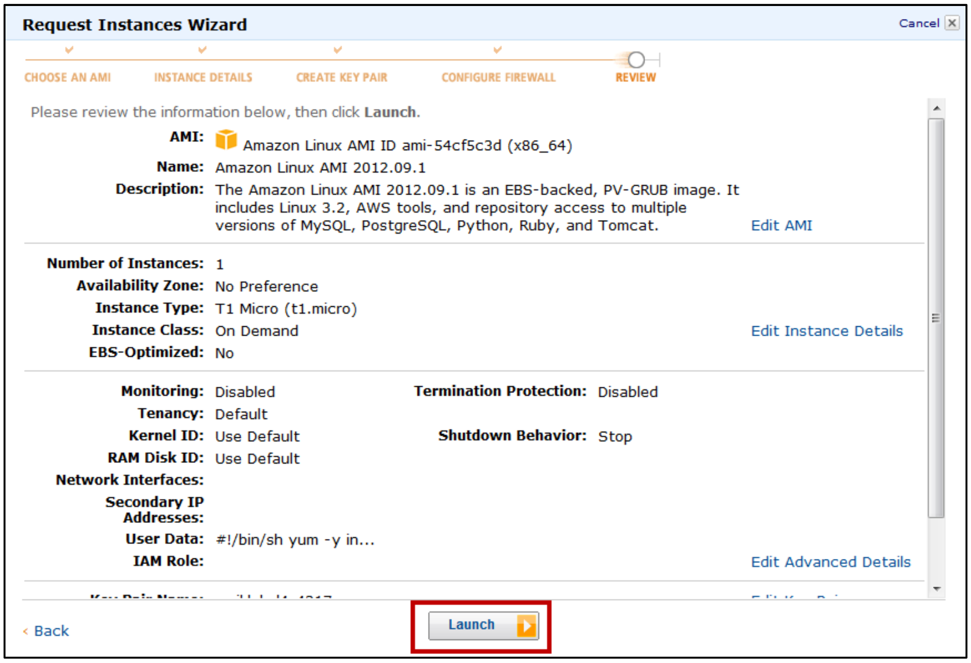
    d. Your TCP Port (Services) list should look like this when complete.



TCP		
Port (Service)	Source	Action
22 (SSH)	0.0.0.0/0	Delete
80 (HTTP)	0.0.0.0/0	Delete

    e. Click Continue when finished.

11. Review your choices, and then click Launch.



Request Instances Wizard

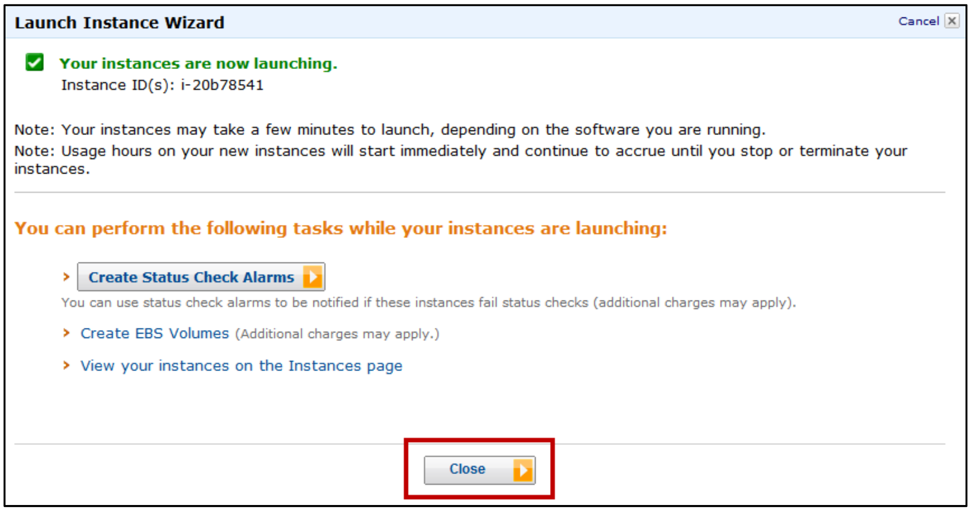
Please review the information below, then click **Launch**.

AMI: Amazon Linux AMI ID ami-54cf5c3d (x86_64)
Name: Amazon Linux AMI 2012.09.1
Description: The Amazon Linux AMI 2012.09.1 is an EBS-backed, PV-GRUB image. It includes Linux 3.2, AWS tools, and repository access to multiple versions of MySQL, PostgreSQL, Python, Ruby, and Tomcat.

Number of Instances: 1
Availability Zone: No Preference
Instance Type: T1 Micro (t1.micro)
Instance Class: On Demand
EBS-Optimized: No

Monitoring: Disabled
Termination Protection: Disabled
Tenancy: Default
Kernel ID: Use Default
Shutdown Behavior: Stop
RAM Disk ID: Use Default
Network Interfaces:
Secondary IP Addresses:
User Data: #!/bin/sh yum -y in...
IAM Role:

12. You will receive a popup window notifying you your instances are launching. Click “Close” to continue.



Launch Instance Wizard

☑ **Your instances are now launching.**
Instance ID(s): i-20b78541

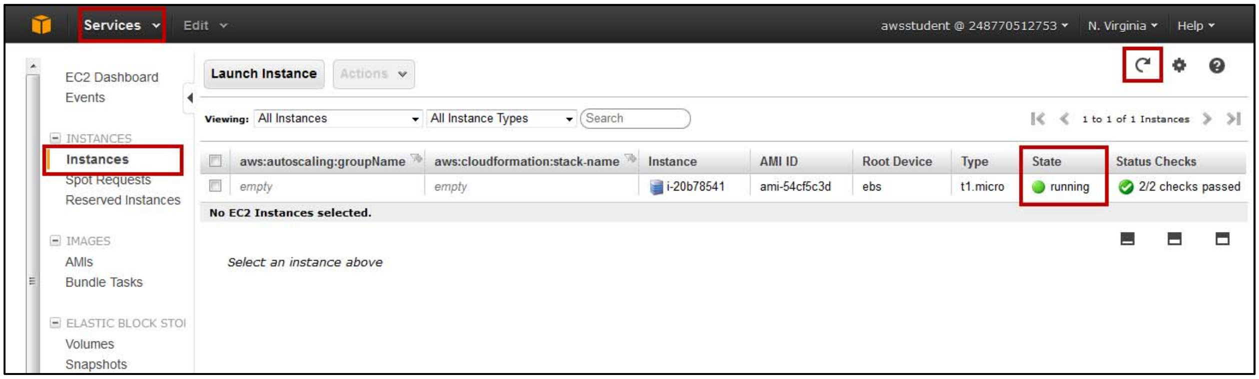
Note: Your instances may take a few minutes to launch, depending on the software you are running.
Note: Usage hours on your new instances will start immediately and continue to accrue until you stop or terminate your instances.

You can perform the following tasks while your instances are launching:

You can use status check alarms to be notified if these instances fail status checks (additional charges may apply).

(Additional charges may apply.)

13. The instance will begin launching.
    a. Monitor it to make certain it’s running by navigating to Services/EC2 and the Instances.
    b. Click refresh in your browser if the state of the instance appears not to be updating.



□	aws:autoscaling:groupName	aws:cloudformation:stack-name	Instance	AMI ID	Root Device	Type	State	Status Checks
□	*empty*	*empty*	i-20b78541	ami-54cf5c3d	ebs	t1.micro	● running	✓ 2/2 checks passed

No EC2 Instances selected.

Select an instance above

Configure the Linux Instance

The instance 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.

1. Log in to your instance using SSH. For instructions, see Appendix A - Connecting to your EC2 Instance.
2. The following must be typed at the $ prompt of your SSH session

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

3. If you are an experienced Linux user, you should know the basics of vi, the default text editor. Otherwise you may want to check out a vi tutorial. Here’s a tutorial that has everything you’ll need in a single HTML page: http://www.tjhsst.edu/~dhyatt/superap/vi.html. More documentation is available from then VIM project, the most popular implementation of vi: http://vimdoc.sourceforge.net/.

4. After starting vi, to start editing, press “i” to turn on insert mode.

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

6. In insert mode, you can place the cursor where you want to paste your copied text, and click the right-mouse button.

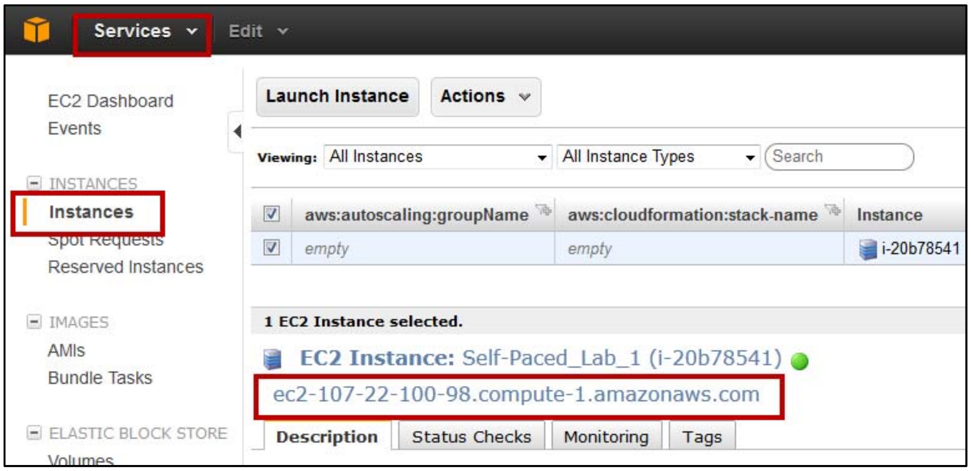
```
<?php
 $url = "http://169.254.169.254/latest/meta-data/instance-id";
 $instance_id = file_get_contents($url);
 echo "Instance ID: <b>" .  $instance_id .  "</b><br/>";
 $url = "http://169.254.169.254/latest/meta-data/placement/availability-zone";
 $zone = file_get_contents($url);
 echo "Zone: <b>"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

7. Once your text appears, press Escape
8. Type :wq to save and quit.

> Note: If for some reason vi is interrupted in the middle of editing (for example through a reboot), then it will salvage the currently edited file. After starting vi again, it will offer to recover the file for you. Just follow the instructions and proceed with the instructions above. You may or may not need to change or delete existing lines using the "i" or "dd" commands in vi.

Connect to the web server

1. Find the DNS name of your instance:



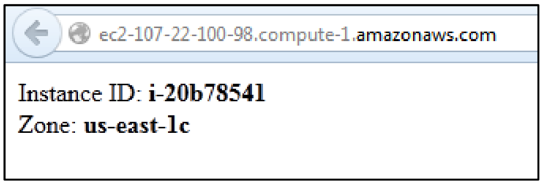
☑	aws:autoscaling:groupName	aws:cloudformation:stack-name	Instance
☑	empty	empty	i-20b78541

1 EC2 Instance selected.

EC2 Instance: Self-Paced_Lab_1 (i-20b78541)

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

2. Enter the DNS name of your instance into your browser and connect to the server.
    a. If successful, you will see your instance ID and Zone appear:



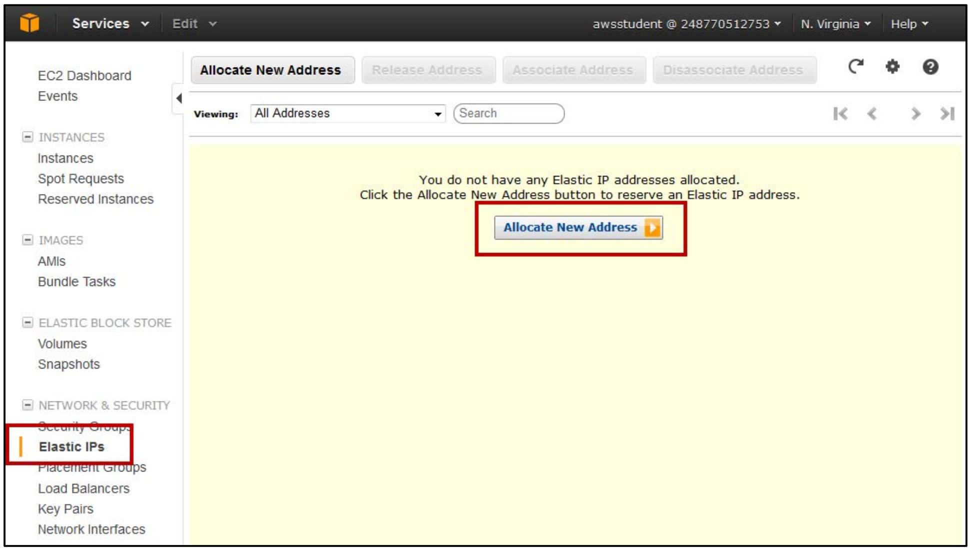
Instance ID: **i-20b78541**
Zone: **us-east-1c**

Assign a Static IP Address

AWS offers Elastic IP Addresses (EIPs), which are actually NAT addresses that operate at a regional level. That is, an Elastic IP Address works across Availability Zones, within a single region.

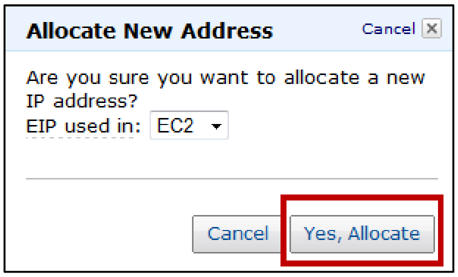
Let's assign an EIP to your instance.

1. Click on the Elastic IPs link in the AWS Console
2. Click on Allocate New Address



You do not have any Elastic IP addresses allocated.
Click the Allocate New Address button to reserve an Elastic IP address.

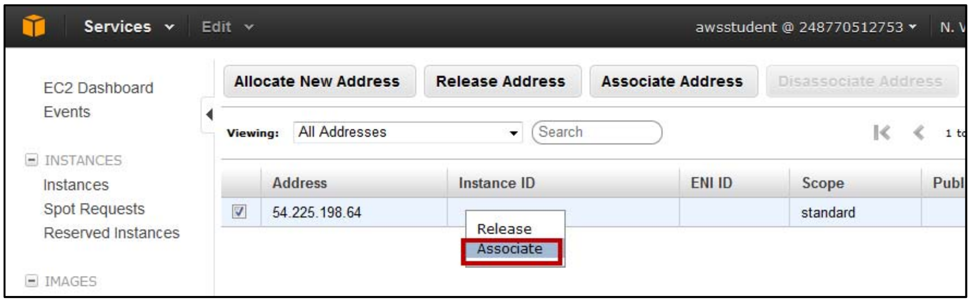
3. You'll see a confirmation prompt. Click Yes, Allocate



Are you sure you want to allocate a new IP address?

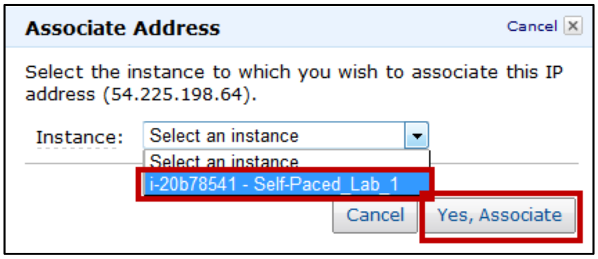
EIP used in: EC2

4. After confirmation, you'll see your newly allocated EIP. Right-click on it and choose Associate from the pop-up menu.



	Address	Instance ID	ENI ID	Scope	Publ
☑	54.225.198.64			standard	

5. A popup will allow you to associate the EIP with one of your running instances. Choose the instance that you just launched and click Yes, Associate:

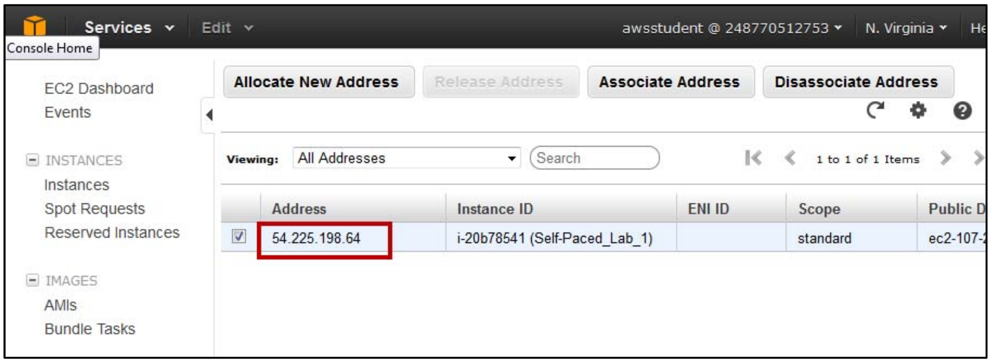


Associate Address

Select the instance to which you wish to associate this IP address (54.225.198.64).

Instance: Select an instance

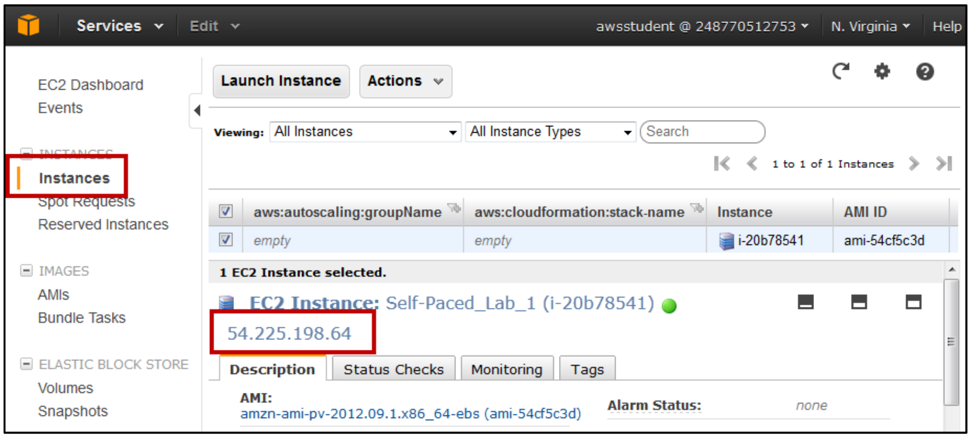
6. Now your Elastic IP address is associated to your instance:



Viewing: All Addresses 1 to 1 of 1 Items

	Address	Instance ID	ENI ID	Scope	Public D
☑	54.225.198.64	i-20b78541 (Self-Paced_Lab_1)		standard	ec2-107-

7. And your instance should now report its new IP address in the console:



Viewing: All Instances All Instance Types 1 to 1 of 1 Instances

☑	aws:autoscaling:groupName	aws:cloudformation:stack-name	Instance	AMI ID
☑	*empty*	*empty*	i-20b78541	ami-54cf5c3d

1 EC2 Instance selected.

EC2 Instance: Self-Paced_Lab_1 (i-20b78541)

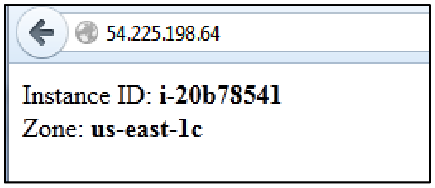
54.225.198.64

AMI: amzn-ami-pv-2012.09.1.x86_64-ebs (ami-54cf5c3d)

Alarm Status: *none*

8. Now, verify the new IP address of your web server in a browser:

If successful, you will see your instance ID and Zone appear:

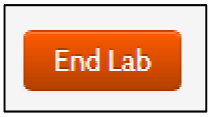


54.225.198.64

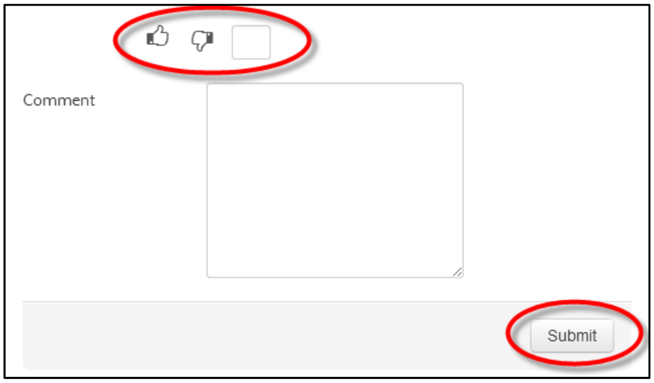
Instance ID: **i-20b78541**
Zone: **us-east-1c**

End Your *qwikLAB™* Session

1. Sign-out of the AWS Management Console.
2. Click the End Lab button in qwikLAB™.



Give the lab a thumbs-up/down, or enter a comment and click Submit



Errors in this lab guide can be reported to aws-course-feedback@amazon.com

Summary

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,
- Assigned a fixed public IP address (Elastic IP) to your instance.

We hope you enjoyed working through this tutorial and that you now have everything you need to start using EC2 for your own projects. Please feel free to explore our other self-paced labs to learn more about Amazon Web Services.

Appendix A - Connecting to your EC2 Instance via SSH

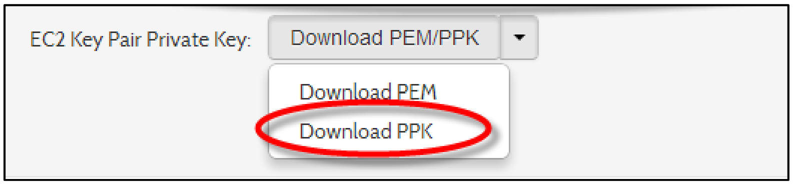
Windows

Download PuTTY

1. Download PuTTY to a location of your choice unless you already have PuTTY. http://the.earth.li/~sgtatham/putty/latest/x86/putty.exe

Download your EC2 Key Pair private key file

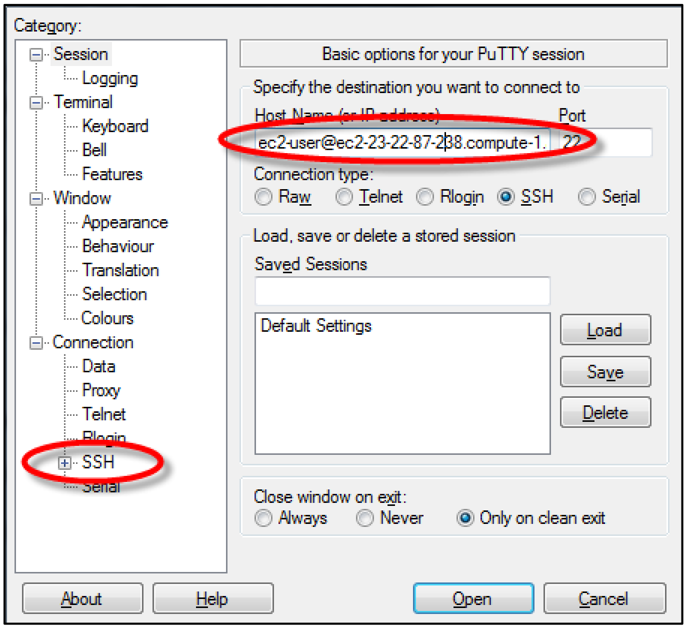
2. Go back to your lab in *qwikLAB*™.
3. Download the *qwikLAB*™ provided EC2 Key Pair private key file in the PuTTY compatible PPK format by clicking on the Download PPK option in the “Download PEM/PPK” drop-down.



4. Save the file to your Downloads directory (or some other directory of your choice.)

Connect to the EC2 Instance using SSH and PuTTY.

1. Open the putty.exe you downloaded or already had.
2. Enter ec2-user@<your EC2 hostname> into the Host Name input in Putty (Ctrl+v).
3. Expand the SSH category by clicking on it.



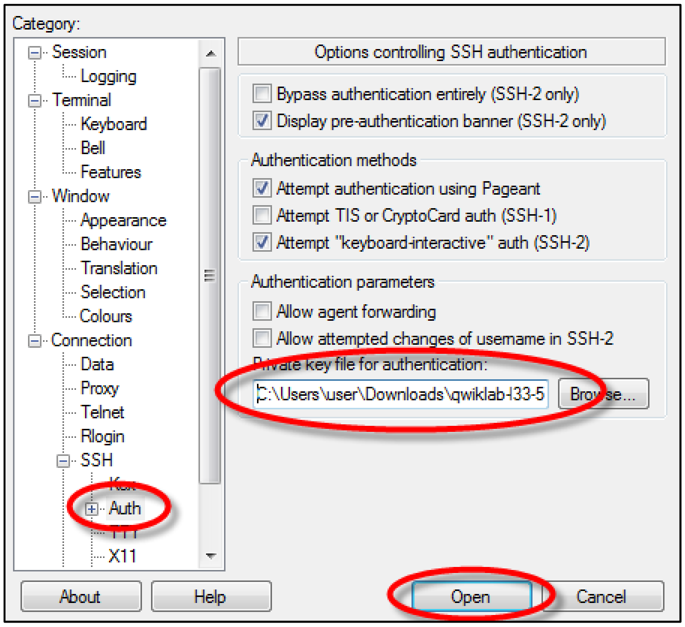
Host Name (or IP address): ec2-user@ec2-23-22-87-238.compute-1.
Port: 22

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

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

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

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



Options controlling SSH authentication

☐ Bypass authentication entirely (SSH-2 only)

☑ Display pre-authentication banner (SSH-2 only)

Authentication methods

☑ Attempt authentication using Pageant

☐ Attempt TIS or CryptoCard auth (SSH-1)

☑ Attempt "keyboard-interactive" auth (SSH-2)

Authentication parameters

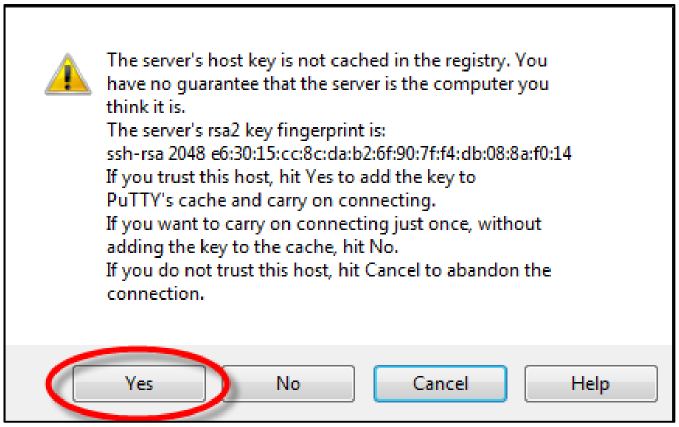
☐ Allow agent forwarding

☐ Allow attempted changes of username in SSH-2

Private key file for authentication:

C:\Users\user\Downloads\qwiklab-l33-5 Browse...

Click Yes when prompted to allow a first connection to this remote SSH server.

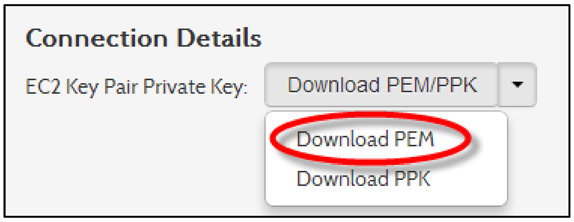


The server's host key is not cached in the registry. You have no guarantee that the server is the computer you think it is.
The server's rsa2 key fingerprint is:
ssh-rsa 2048 e6:30:15:cc:8c:da:b2:6f:90:7f:f4:db:08:8a:f0:14
If you trust this host, hit Yes to add the key to PuTTY's cache and carry on connecting.
If you want to carry on connecting just once, without adding the key to the cache, hit No.
If you do not trust this host, hit Cancel to abandon the connection.

OS X and Linux

Download your EC2 Key Pair private key file

1. Go back to your lab in *qwikLAB*™.
2. Download the *qwikLAB*™ provided EC2 Key Pair private key file in the PEM format by clicking on the Download PEM option in the “Download PEM/PPK” drop-down.



3. Save the file to your Downloads directory (or some other directory of your choice.)

Connect to the EC2 Instance using the OpenSSH CLI client

1. Open the Terminal application.
2. Enter the below commands substituting the path/filename for the .pem file you downloaded from *qiwk*LAB™ 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
```