Immersion Day

Getting Started with Linux on Amazon EC2

Rev 2015-01-15

Table of Contents

Overview 3
Create a new Key Pair 4
Launch a Web Server Instance 6
Browse the Web Server 14
Appendix – Additional EC2 Concepts 14
  Change the Instance Type 14
  Black Belt Booting 18
Appendix B – SSH to EC2 instances using MindTerm 19
Appendix C – Using a 3rd Party SSH Client 25
  Windows (PuTTY) 25
  Mac OS X or Linux (OpenSSH) 32

Overview

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides resizable compute capacity in the cloud. Amazon EC2's simple web service interface allows you to obtain and configure capacity with minimal friction. 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.

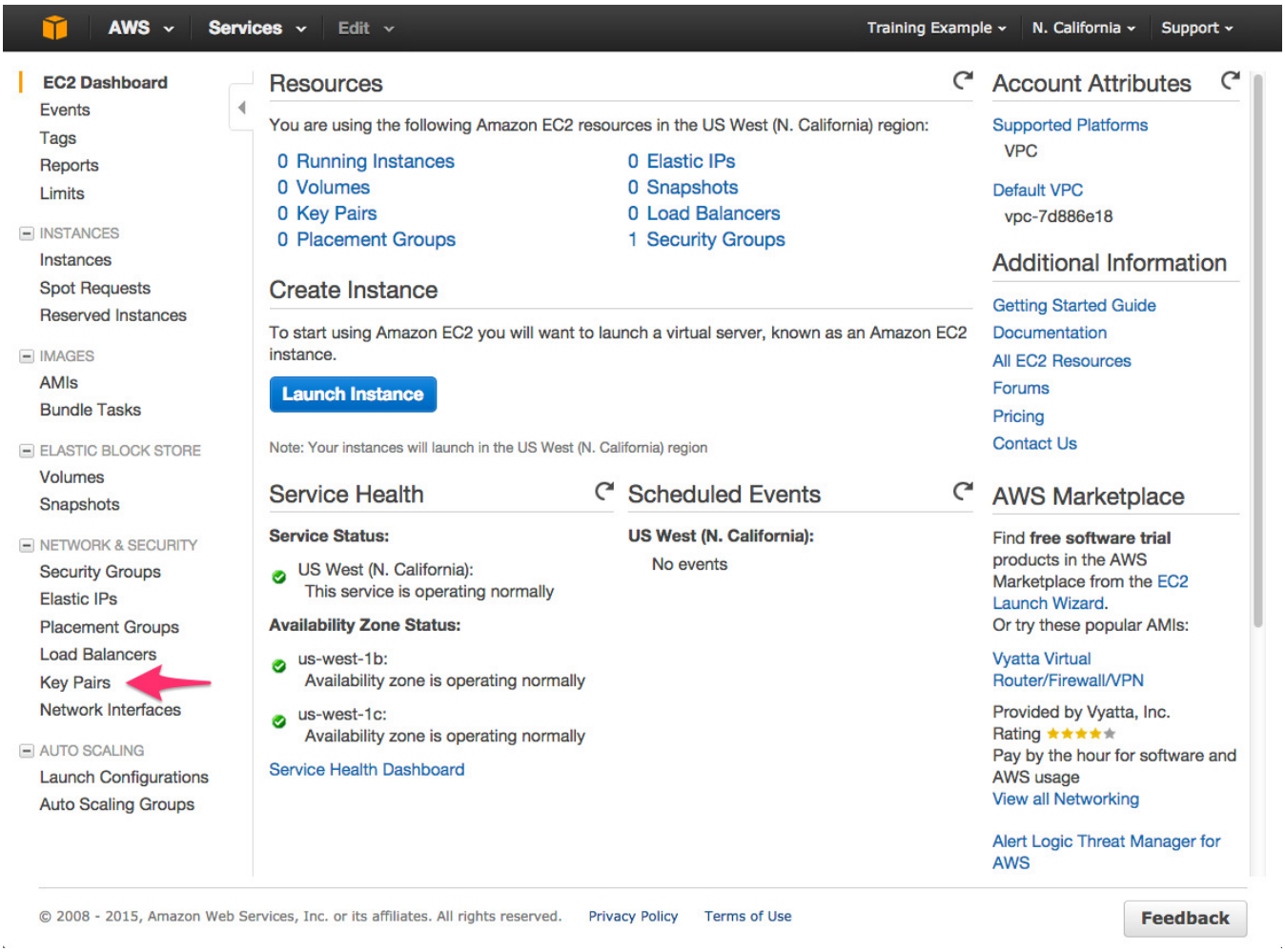
This lab will walk you through launching, configuring, and customizing an EC2 virtual machine to run a web server. It will walk you though successfully provisioning and starting an EC2 instance using the AWS Management Console.

Create a new Key Pair

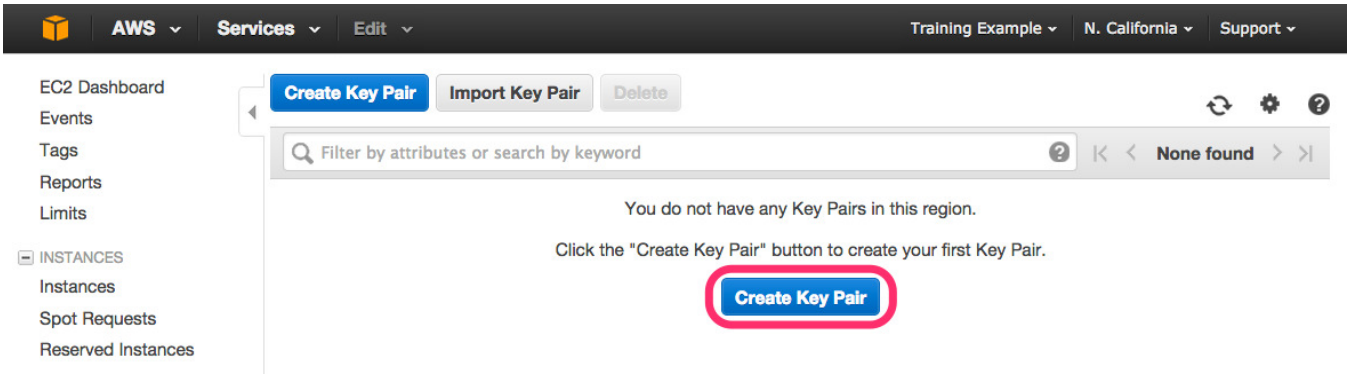
In this lab, you will need to create an EC2 instance using an SSH keypair. The following steps outline creating a unique SSH keypair for you to use in this lab.

1. Sign into the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2.

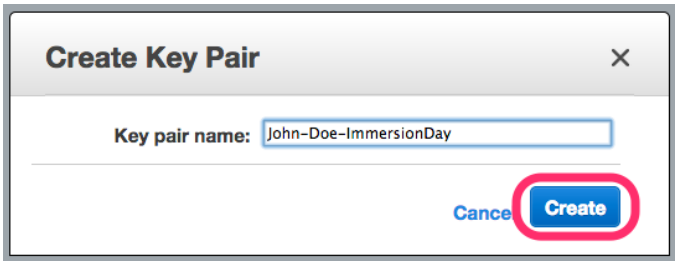
2. Click on **Key Pairs** in the NETWORK & SECURITY section near the bottom of the leftmost menu. This will display a page to manage your SSH key pairs.



3. To create a new SSH key pair, click the **Create Key Pair** button at the top of the browser window.



4. In the resulting pop up window, type *[First Name]-[Last Name]-ImmersionDay* into the **Key Pair Name:** text box and click **Create.**



5. The page will download the file "*[Your-Name]*-ImmersionDay.pem" to the local drive. Follow the browser instructions to save the file to the default download location

6. Remember the full path to the file .pem file you just downloaded.

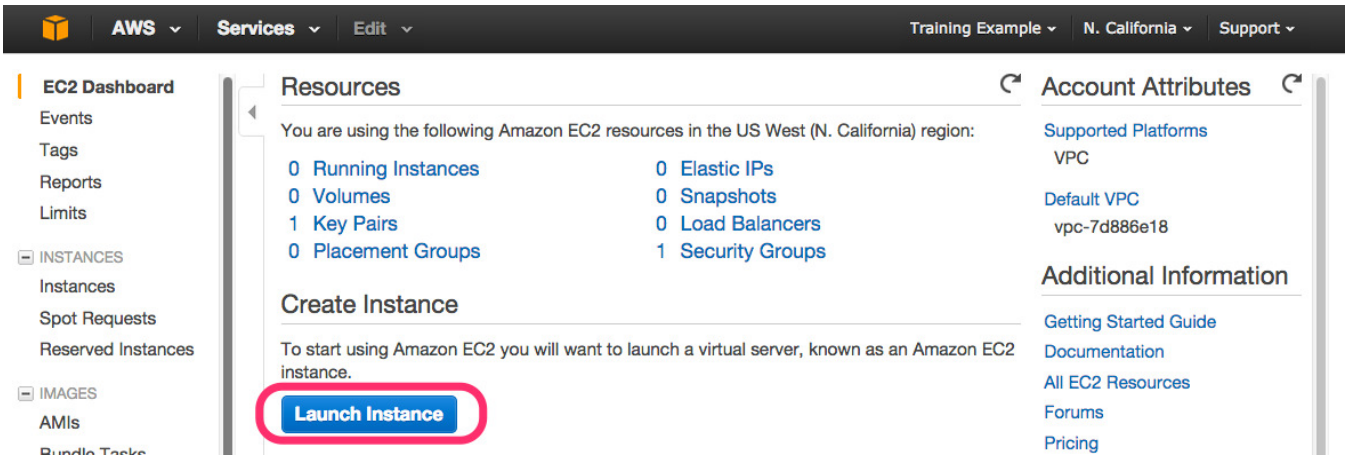
You will use the Key Pair you just created to manage your EC2 instances for the rest of the lab.

Launch a Web Server Instance

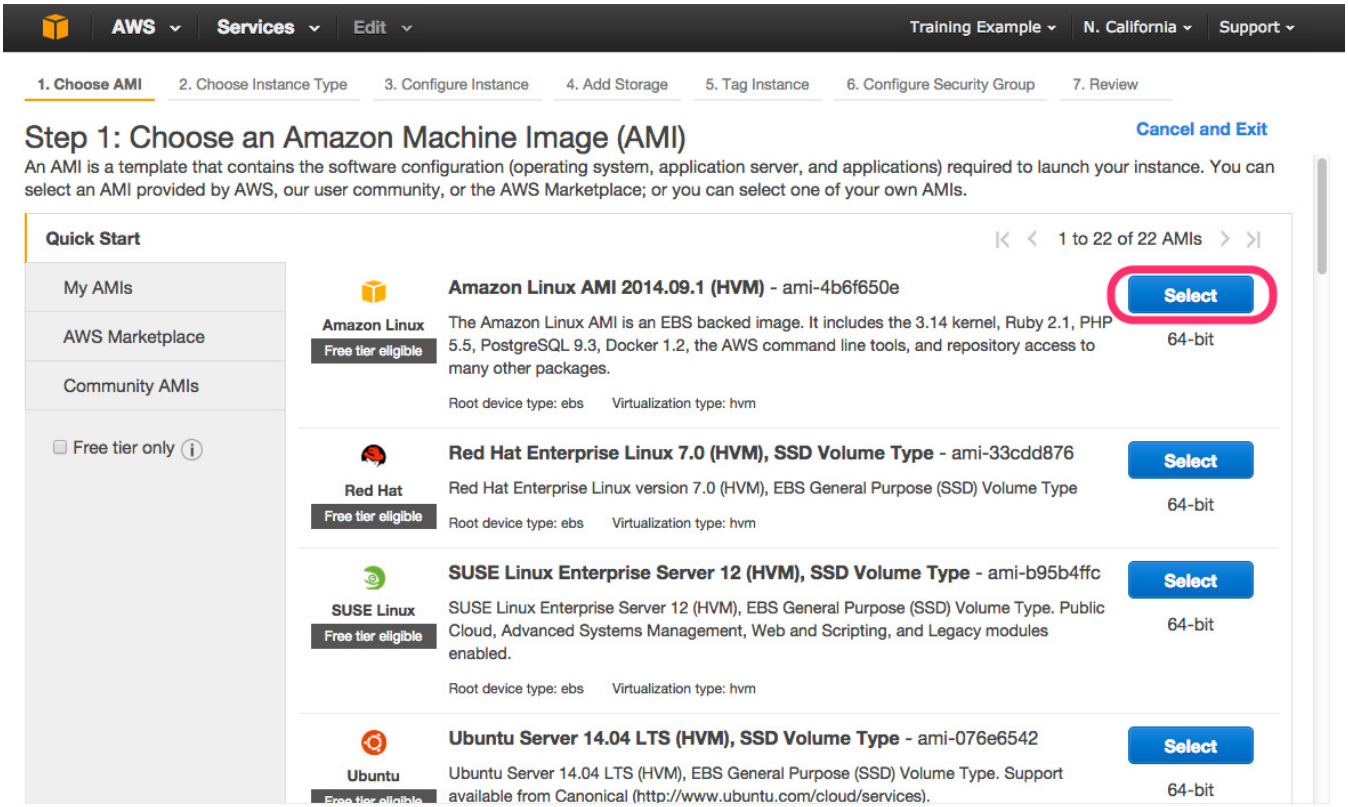
In this example we will launch a default Amazon Linux Instance with an Apache/PHP web server installed on initialization.

7. Sign into the AWS Management Console and open the Amazon EC2 console at https://console.aws.amazon.com/ec2.

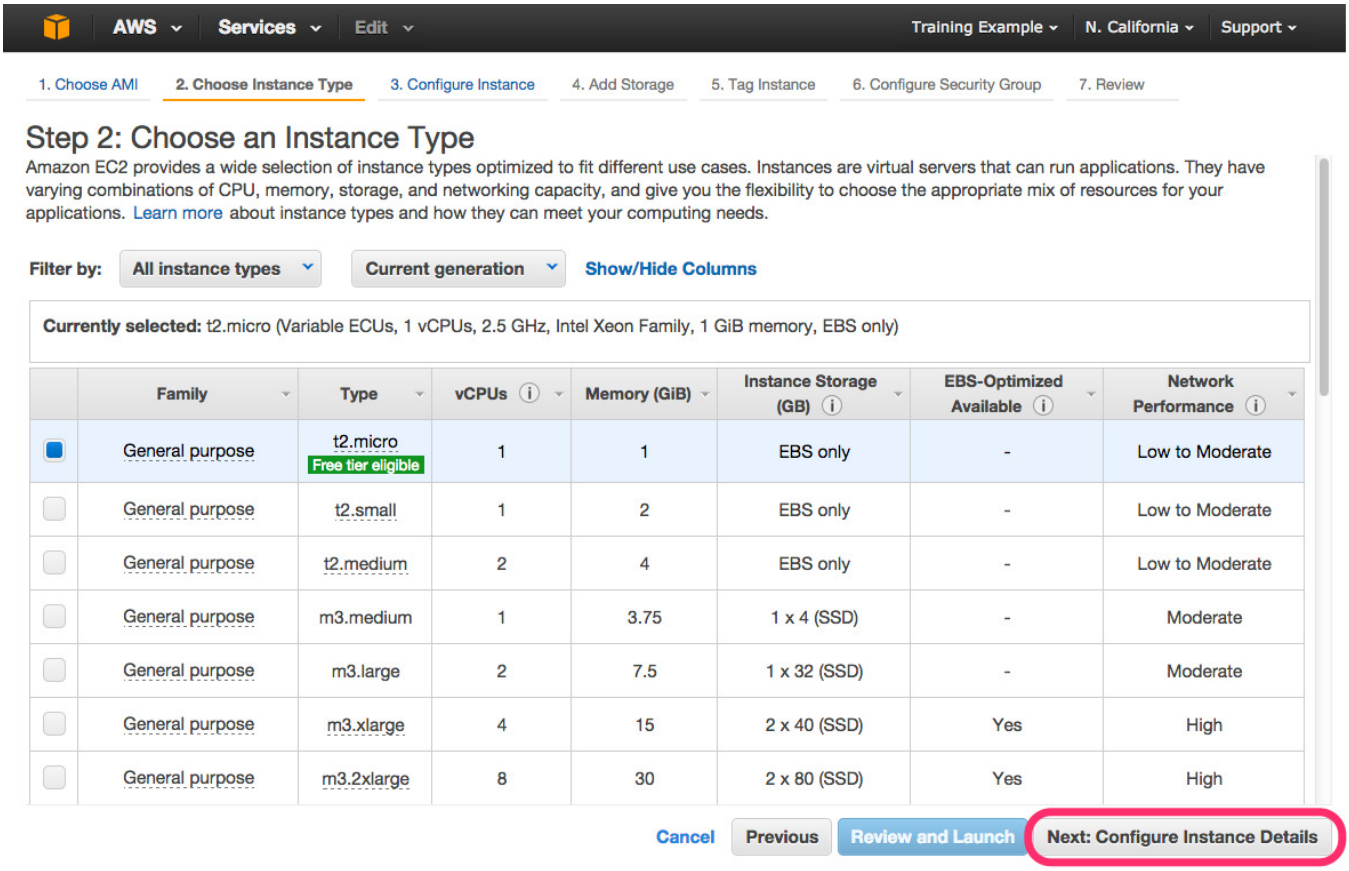
8. Click on **Launch Instance**



9. Select the Amazon Linux AMI and click **Select**

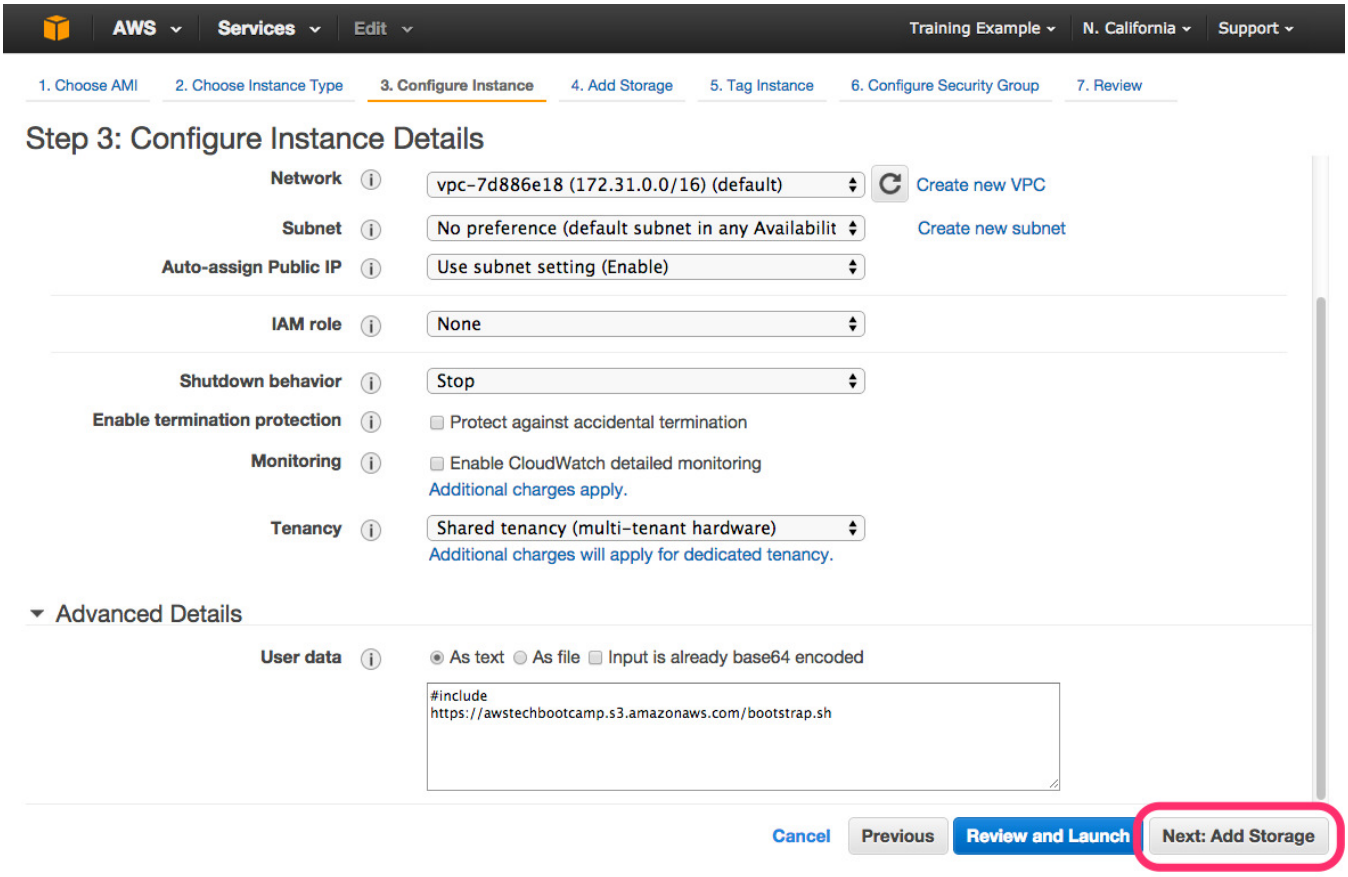


10. Select the **Micro (t2.micro)** instance size and click **Next: Configure Instance Details**

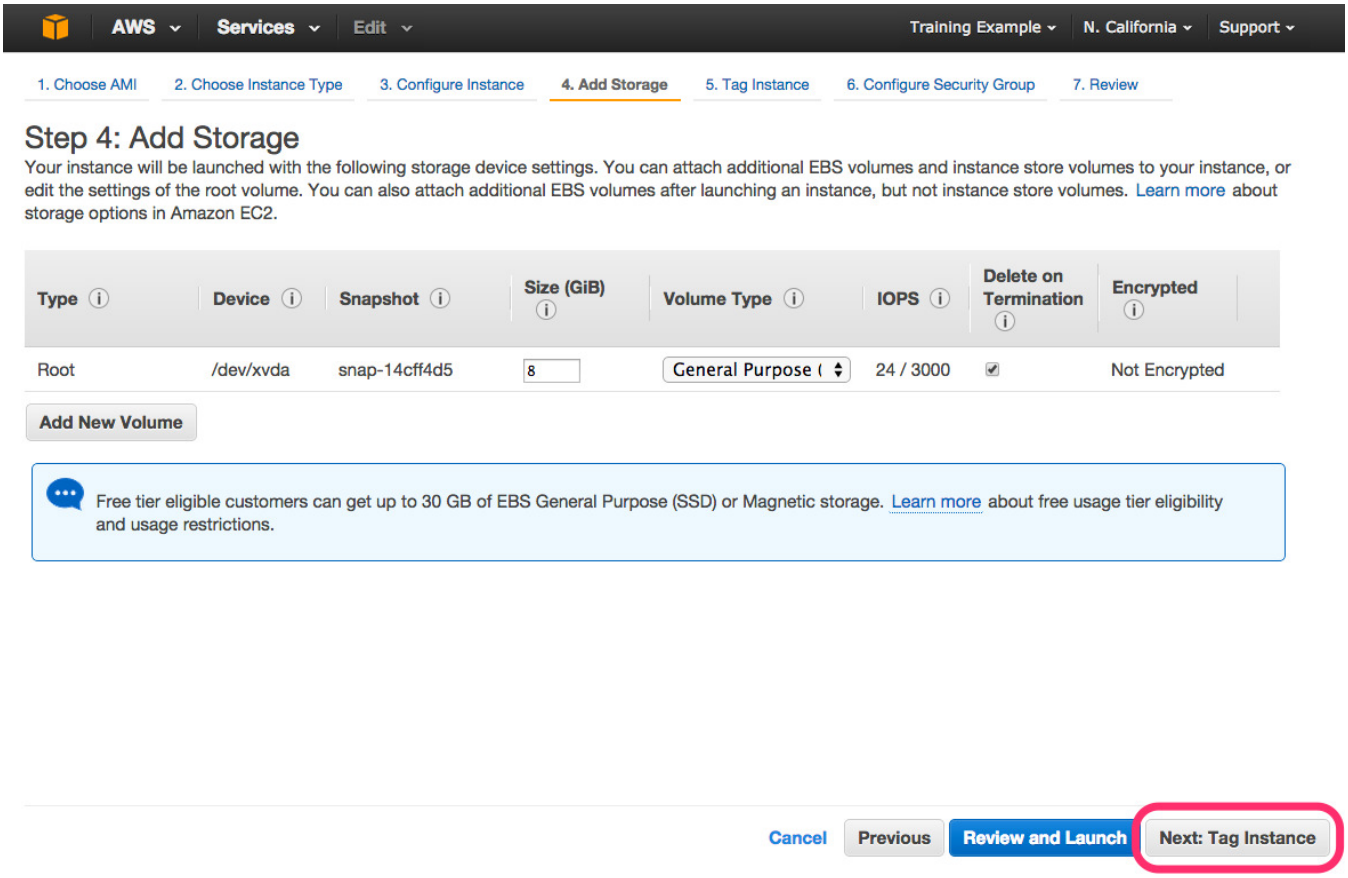


11. On the **Configure Instance Details** page, expand the **Advanced Details** section and type the following initialization script information *(you can use Shift-Enter to create the necessary line break, or alternatively you could type this into Notepad and copy & paste the results)* into the User Data field (this will automatically install and start the Apache web server on launch) and click **Next: Add Storage**:

#include
https://awstechbootcamp.s3.amazonaws.com/bootstrap.sh



12. Click **Next: Tag Instance** to accept the default Storage Device Configuration.

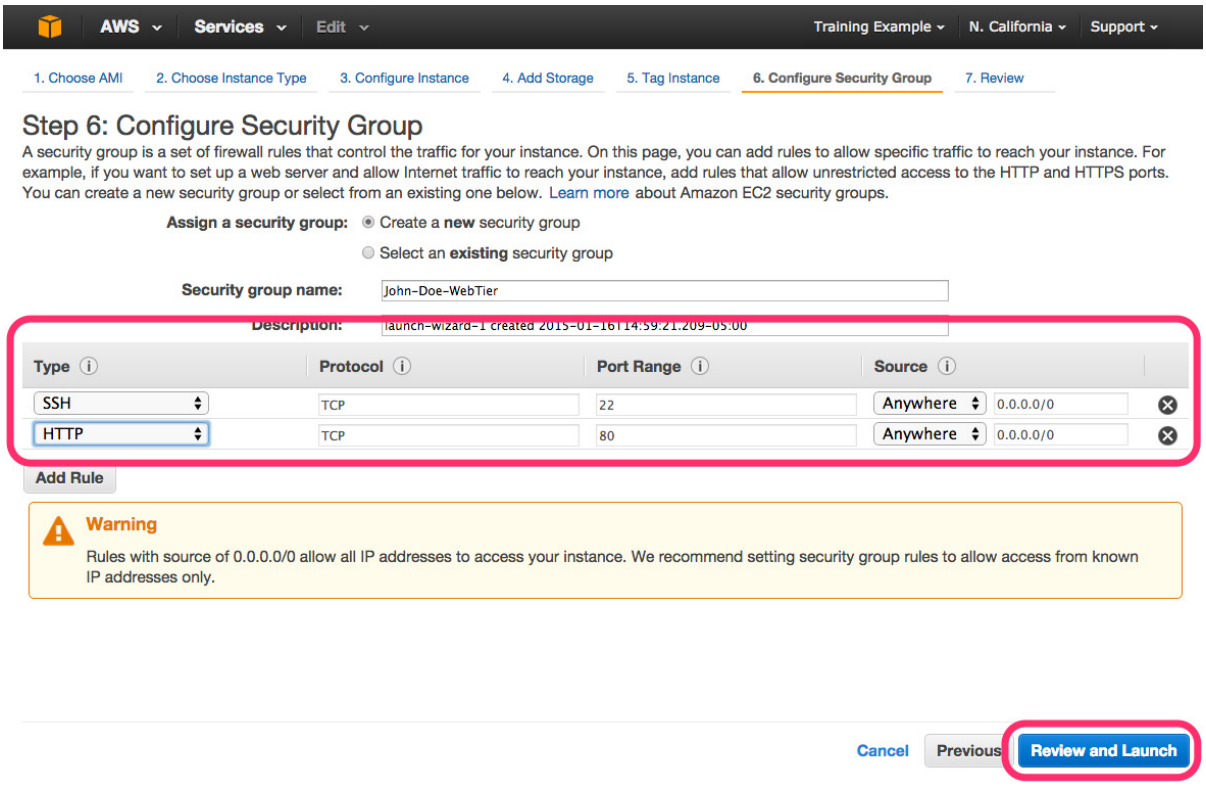


Next, choose a “friendly name” for your instance. This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Named yours according to this format: “[Your Name] Web Server.

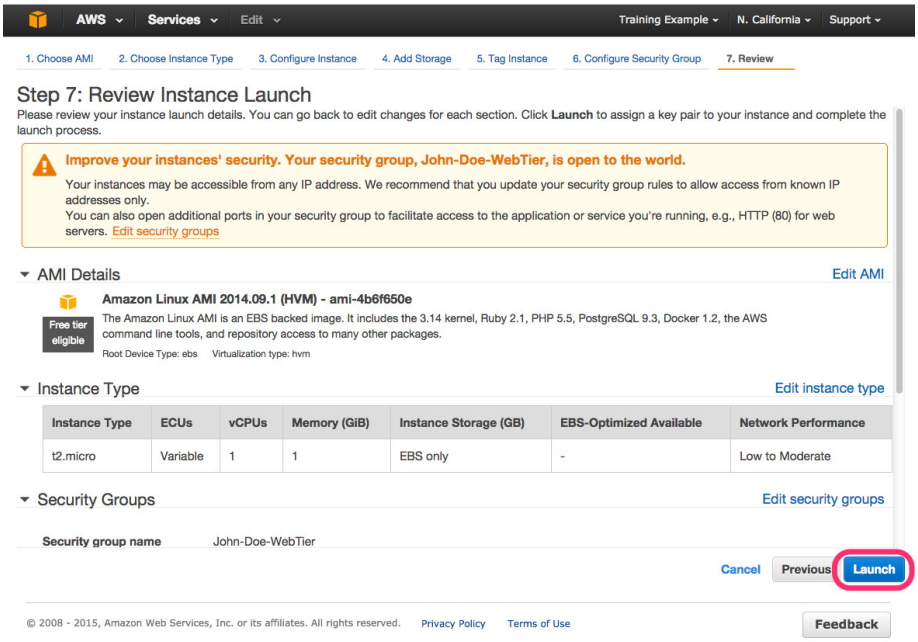
Then click **Next: Configure Security Group**.

13. You will be prompted to create a new security group, which will be your firewall rules. On the assumption that we are building out a Web server, name your group “[Your Name] Web

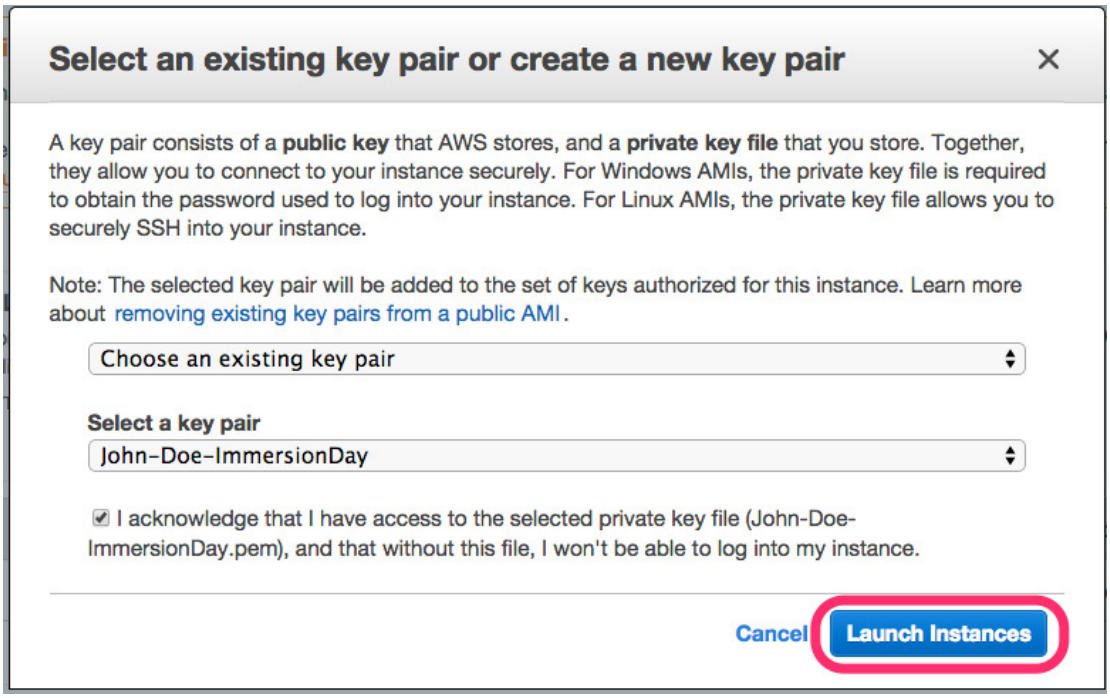
Tier", and open ports 22 and 80. Click the **Review and Launch** button after configuring the security group.



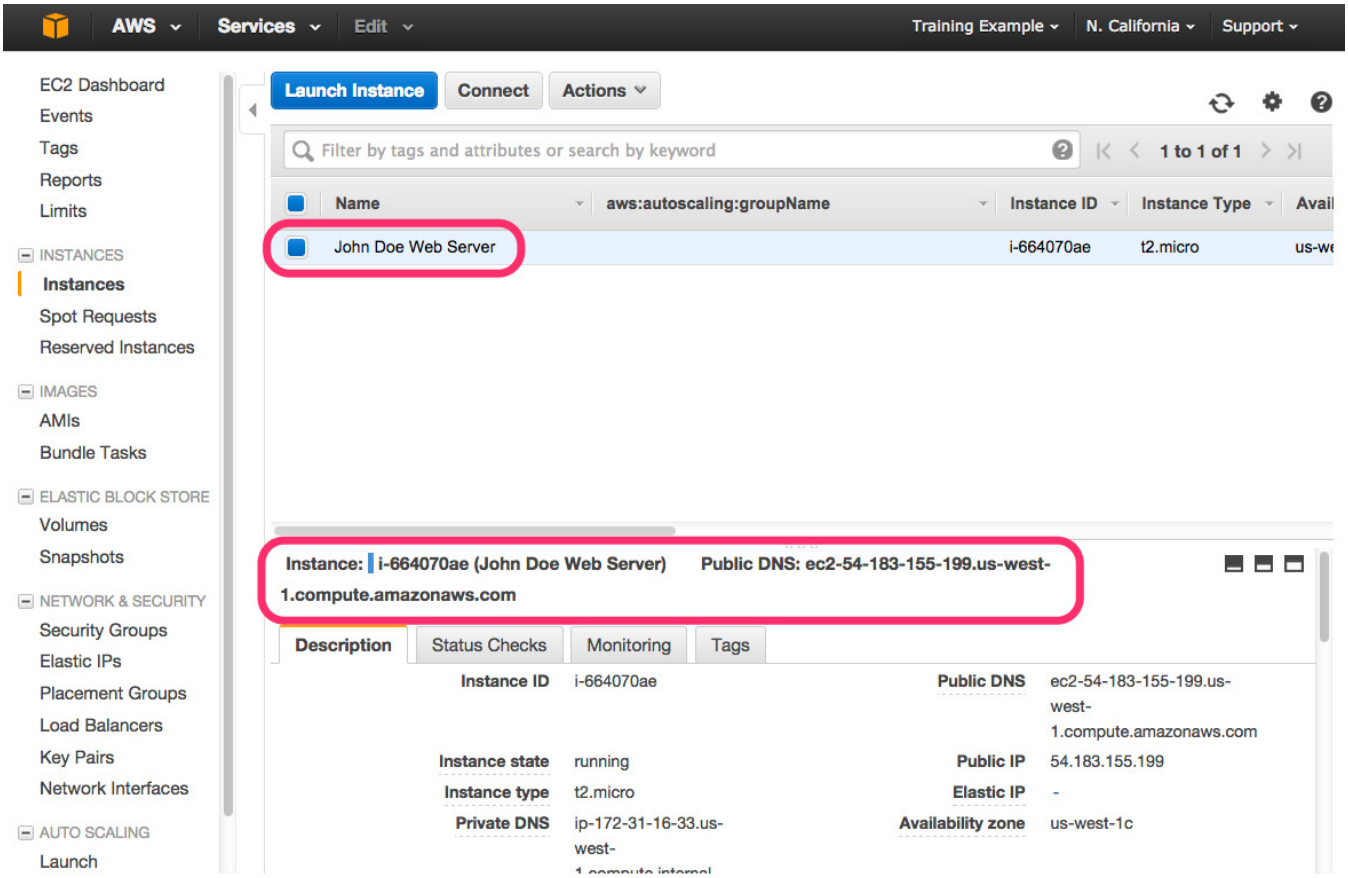
14. Review your choices, and then click **Launch**.



15. Select the *[YourName]-ImmersionDay* key pair that you created in the beginning of this lab from the drop and check the "I acknowledge" checkbox. Then click the **Launch Instances** button.



16. Click the **View Instances** button to view the list of EC2 instances. Once your instance has launched, you will see your “Manual Web Server,” as well as the Availability Zone the instance is in and the publicly routable DNS name.



Browse the Web Server

1. Wait for the instance to pass the Status Checks to finish loading.

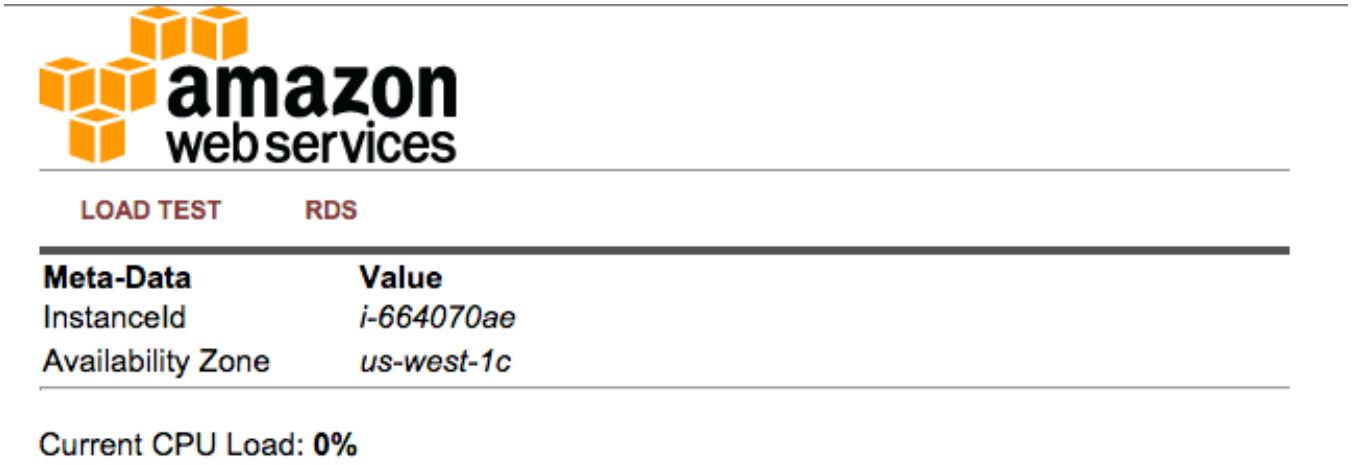
Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
John Doe Web Server	i-664070ae	t2.micro	us-west-1c	running	Initializing

Finished initializing

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks
John Doe Web Server	i-664070ae	t2.micro	us-west-1c	running	2/2 checks passed

2. Open a new browser tab and browse the Web Server by entering the EC2 instance's Public DNS name into the browser. The EC2 instance's Public DNS name can be found in the console by reviewing the "Public DNS" name line highlighted above.

   You should see a website that looks like the following:



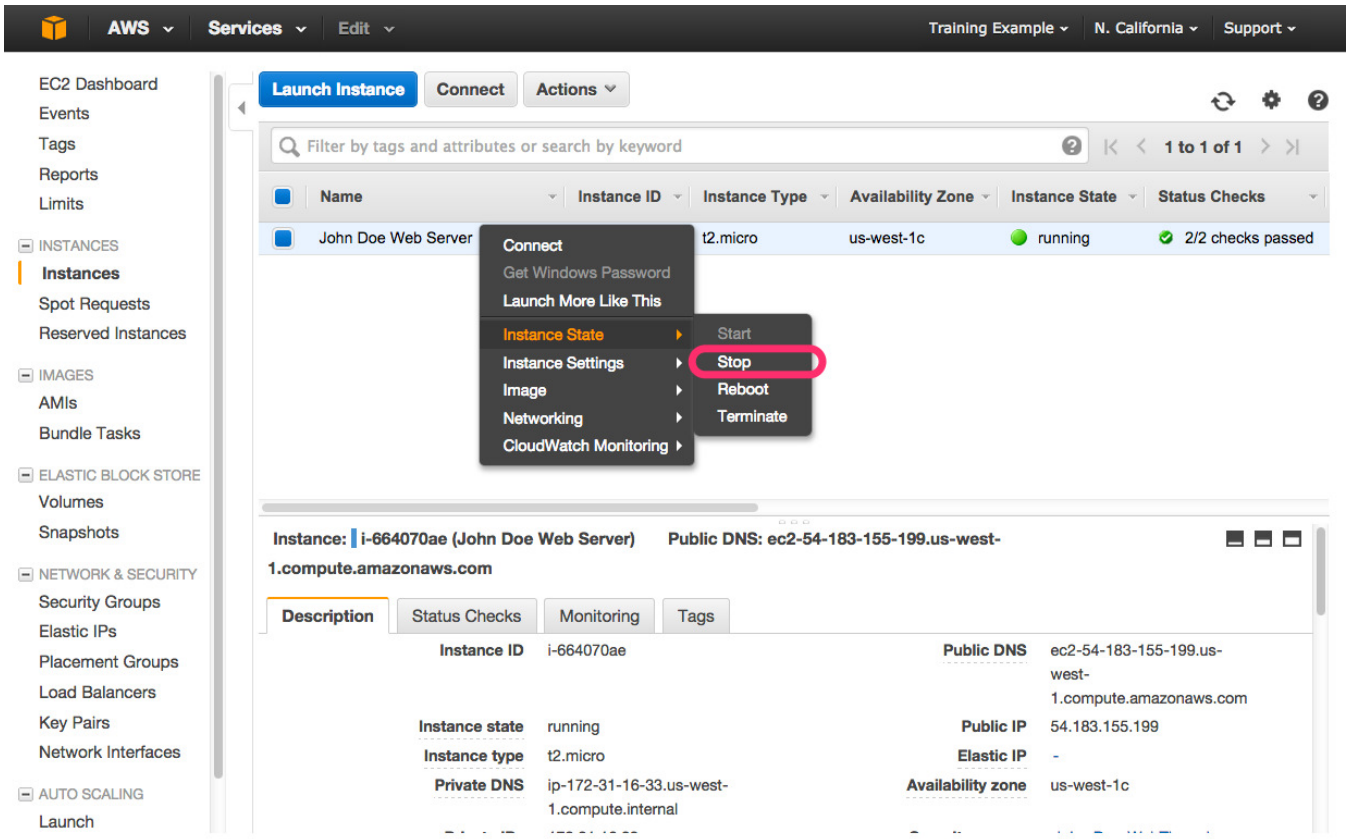
Great Job: You have manually built your first web server!

Appendix – Additional EC2 Concepts

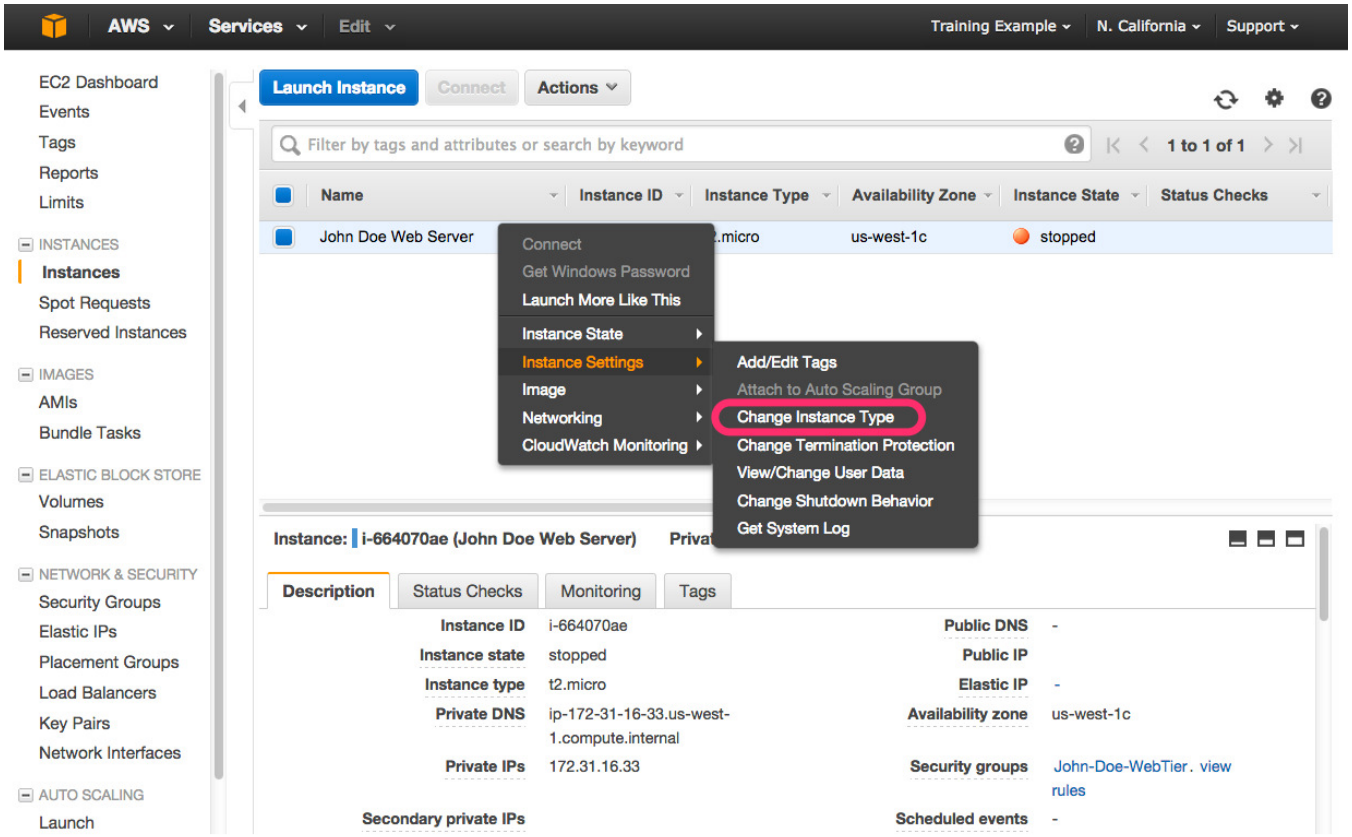
Change the Instance Type

Did you know that you can change the instance type that an AMI is running on? This only works with EBS-backed instances (what we're running here). There is no particular reason to change the instance type in this lab, but the following steps outline how easy it is to do in AWS.

In the AWS Console, select your lab instance, then right-click on it and hover over “Instance State” and select “Stop” (NOT “Terminate”!). Confirm you would like to stop the instance by selecting “Yes, Stop”.



After it has stopped, right-click on it again and hover over “Instance Settings” and select “Change Instance Type”.



After going through the options and selecting your new instance type, right-click your lab instance and start it again.

Elastic IPs

How do you set up practical DNS names for your web server? Using an address such as http://ec2-75-101-197-112.compute-1.amazonaws.com/ is not likely to win the day with your customers. Setting up a DNS record that points to http://www.yourdomain.com is easy enough – until you reboot the server and the underlying DNS name and IP address both change.

AWS offers Elastic IP Addresses, 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. Assign one to your application as follows:

- Click on the Elastic IPs link in the AWS Console
- Allocate a new address
- Associate the address with a running instance. If you change instances, it's as simple as allocating the address to the new instance.
- Create a DNS "A" record in your own DNS server that points tt.mydomain.com to 75.101.162.40.

Two Important Notes:

1. As long as an Elastic IP address is associated with a running instance, there is no charge for it. However an address that is not associated with a running instance costs $0.01/hour. This prevents address hoarding; however it also means that you need to delete any addresses you create, or you will incur an ongoing charge.
2. Load balancing (covered in the next section) requires CNAME records instead of "A" records. So Elastic IP is not required for load-balanced applications.

Black Belt Booting

There are a number of advanced techniques that offer additional power and flexibility when booting Linux instances. For example, some organizations maintain a series of generic instances, and customize the images upon launch.

Common techniques include:

- Automatically check for updates upon each boot.
- Look in a well-known location, such as in a S3 bucket, for data or a script to tell the instance which packages to load.
- Pass **user data** to the instance to accomplish each of the above, or possibly instead of the other approaches.

How to Pass User Data

The general format looks like this from the command line:

```
aws ec2 run-instances --image-id [ami id] --user-data "user data up
to 2048 bytes" ...other params...
```

You can also paste user data into a text field via the AWS Console; however this is usually a form of automation – thus the command line example.

Security Concerns

All of the methods except the final one require that your AMIs have security keys embedded in the image (unless you are using IAM Roles for EC2 instances). That is a serious security concern, and we do not recommend storing the keys on your instance.

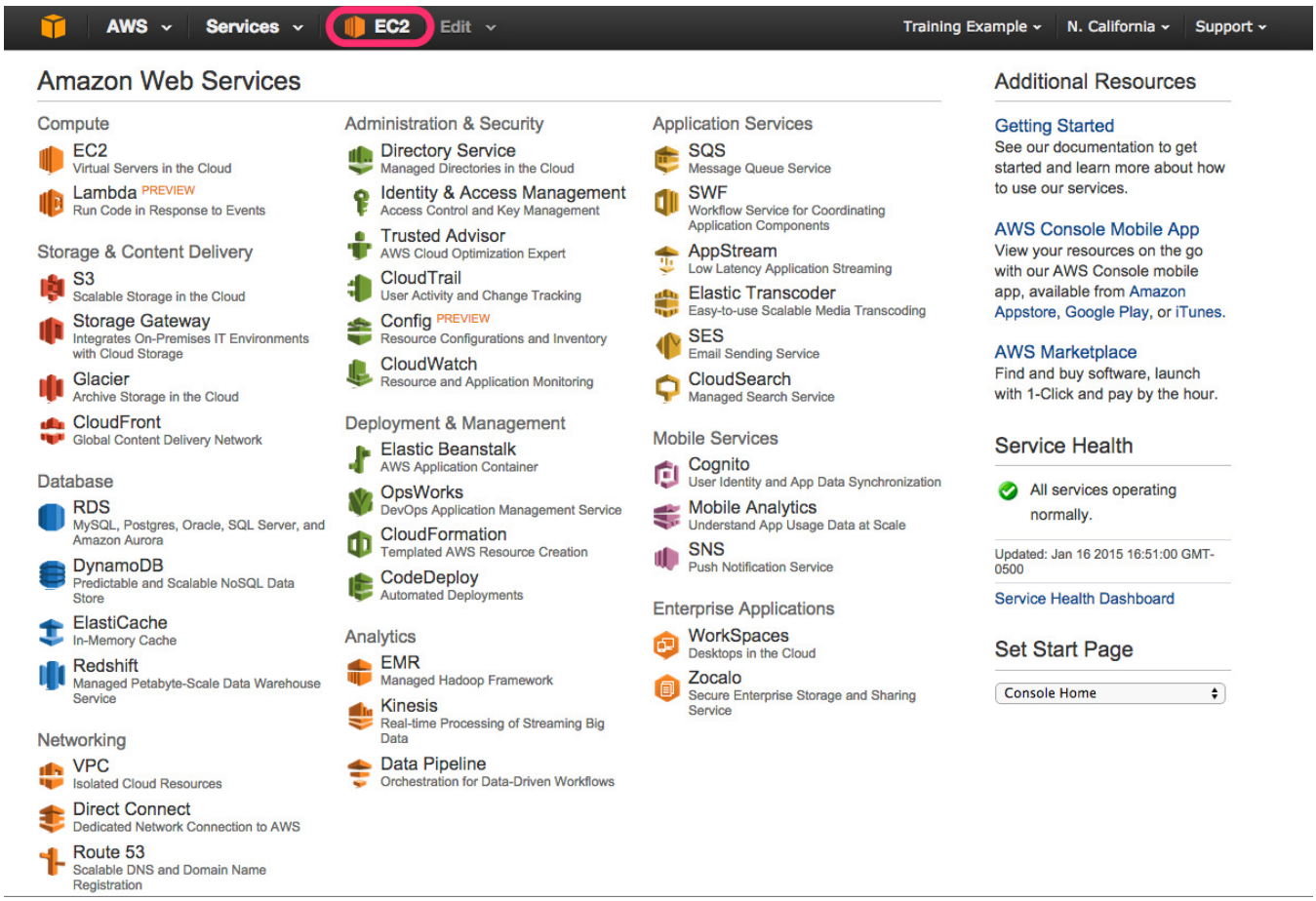
By passing user data, the keys can be stored locally on a master control server. There is some risk that the keys will be compromised; however it's a much lower risk than storing keys on the AMI. It is still a risk, though! That's because User Data can't be encrypted – although it does arrive at the control plane via https.

Even our Solutions Architects at AWS are not in complete agreement about risk vs. benefit of user data. We suggest that you start a class conversation on this subject, if there has not been one by the time that you read this section.

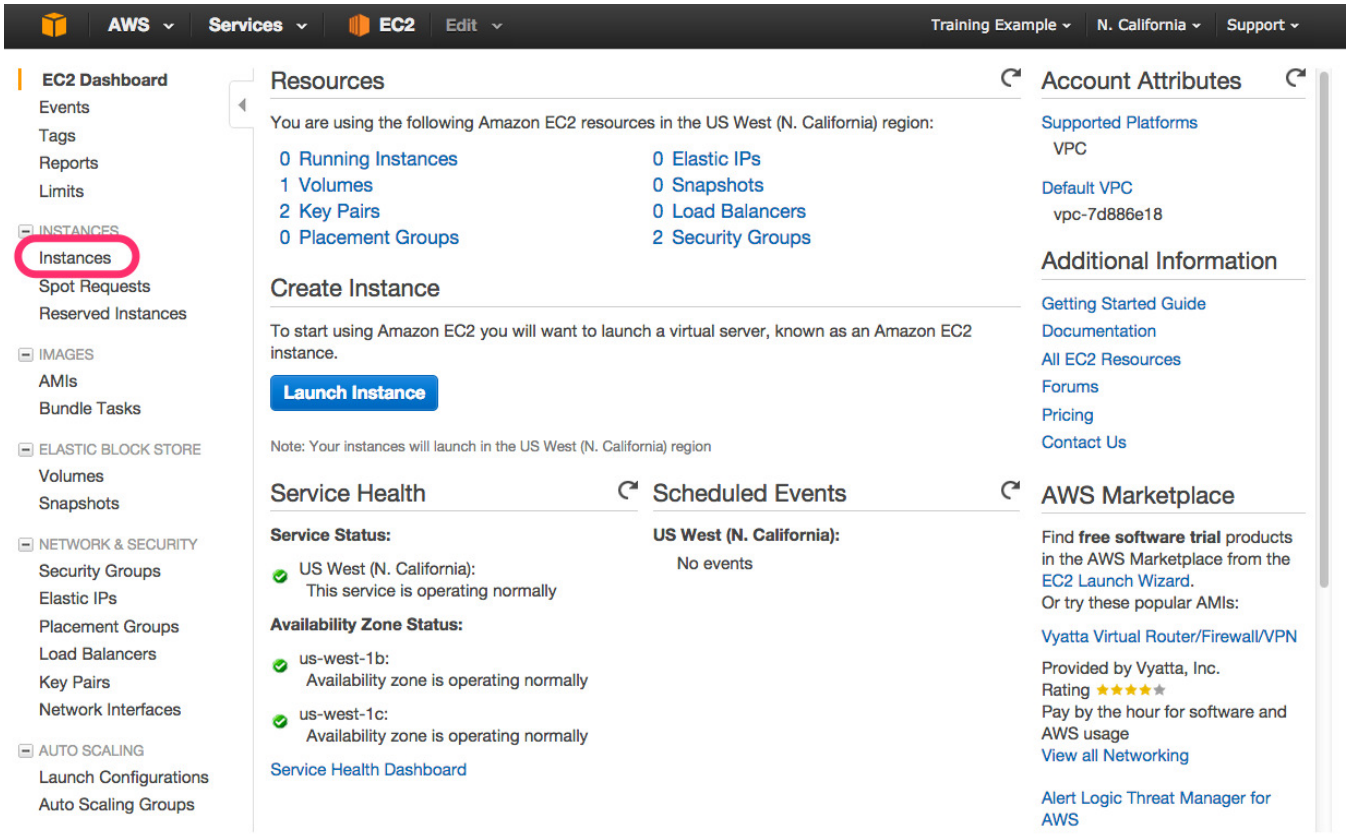
Appendix B – SSH to EC2 instances using MindTerm

In this example, we will connect to an EC2 instance that has been configured for the lab. These instructions require Java to launch the MindTerm SSH client through the console.

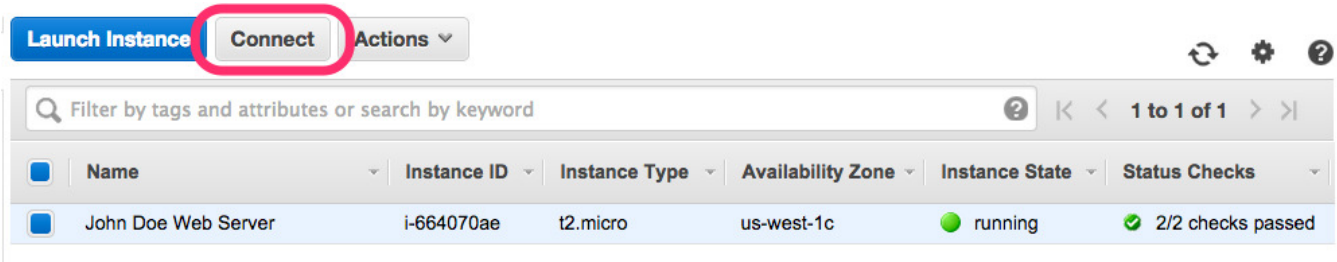
1. Navigate to the EC2 section of the AWS Console by clicking on the **EC2** shortcut.



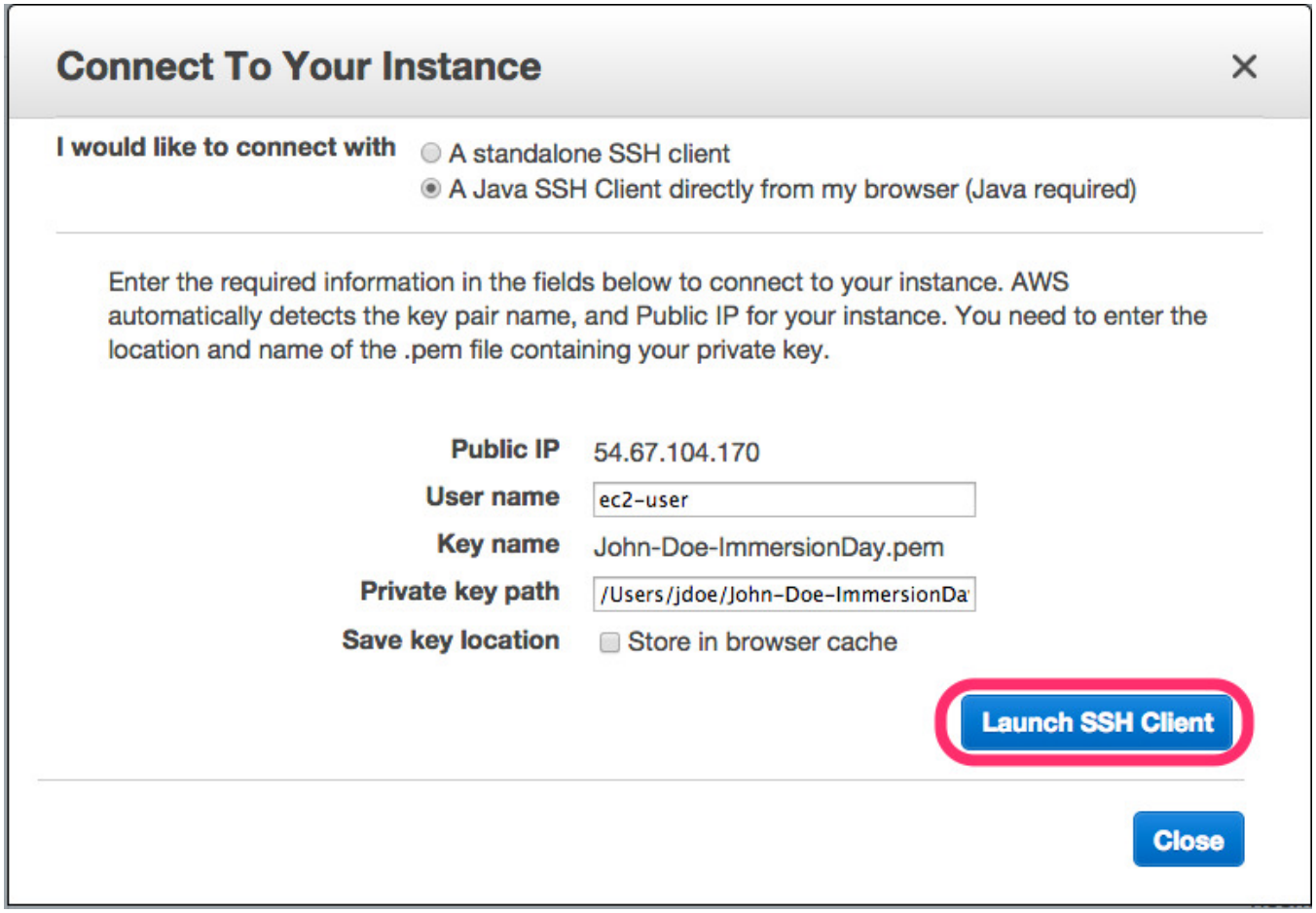
2. Click on **Instances** in the left menu**.**



3. Verify that the Instance is running and **click** on the instance and select **Connect** on the toolbar.

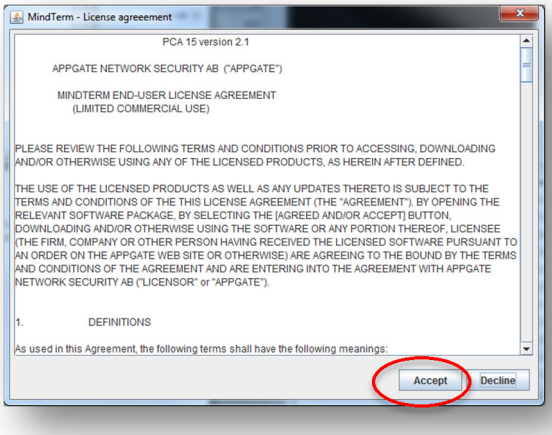


4. Make sure the User name is **ec2-user**, provide the location to your private key (confirm the Private Key Path value in your Workshop Configuration Details sheet), and **check** the option to **save the key location** (not the key itself) in browser cache so you will not have to retype this location in every time you connect to EC2 instances. Then click on **Launch SSH Client**.

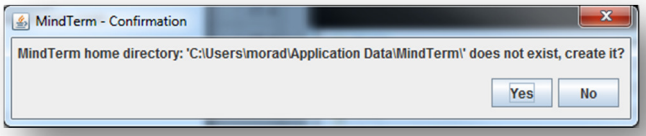


It can take some time for the MindTerm applet to download and run. If you do not have Java installed on your computer, or cannot launch the Java applet from your browser, please see **Appendix B** for instructions on using a 3rd party SSH client to connect to your EC2 instance.

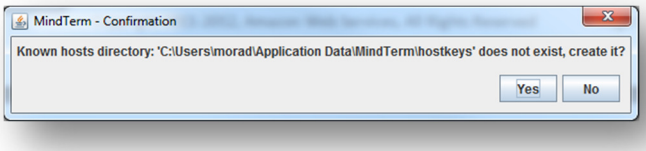
If this is the first time you have used MindTerm, you will be prompted to accept the MindTerm EULA. **Click Accept**:



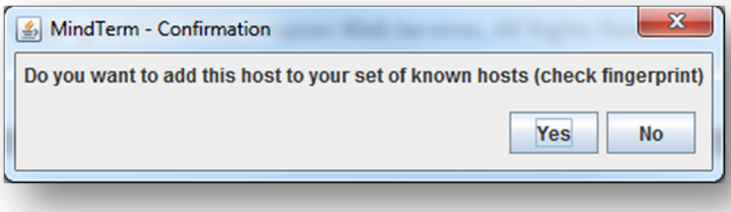
You will be asked to create a directory for MindTerm. Click Yes:



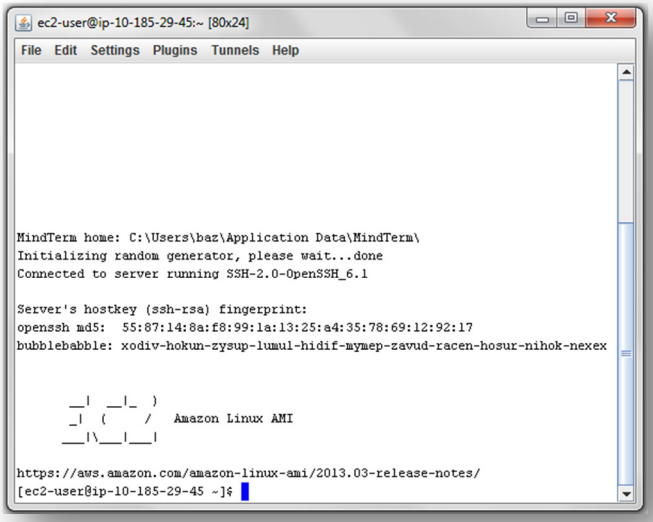
Next you will be asked to create a directory for MindTerm to use to store host keys. Click Yes:



And finally you will be asked if you want to store the host key for your Instance. At this point you have the option to verify the host key MindTerm is seeing with the host key provided by the AWS console to verify that you are connecting directly to your EC2 instance and not some third-party in the middle. Click Yes:



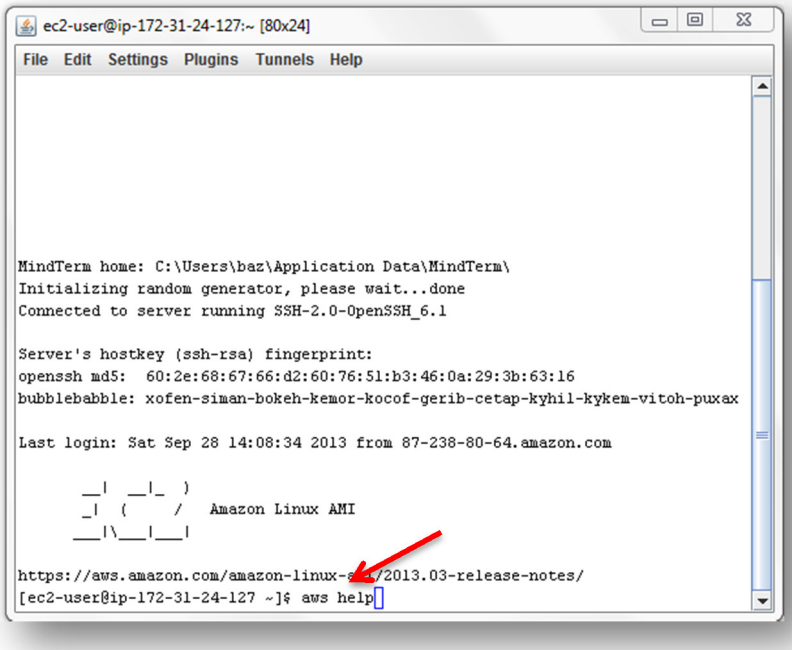
You should be logged into your Instance:



5. Launch the AWS CLI help manual from the terminal command line
   Type the following AWS CLI command:

```
$ aws help
```

   Continue to press the **SPACE** key to scroll through the manual until you reach the end. Press the **Q** key to return back to the command line.

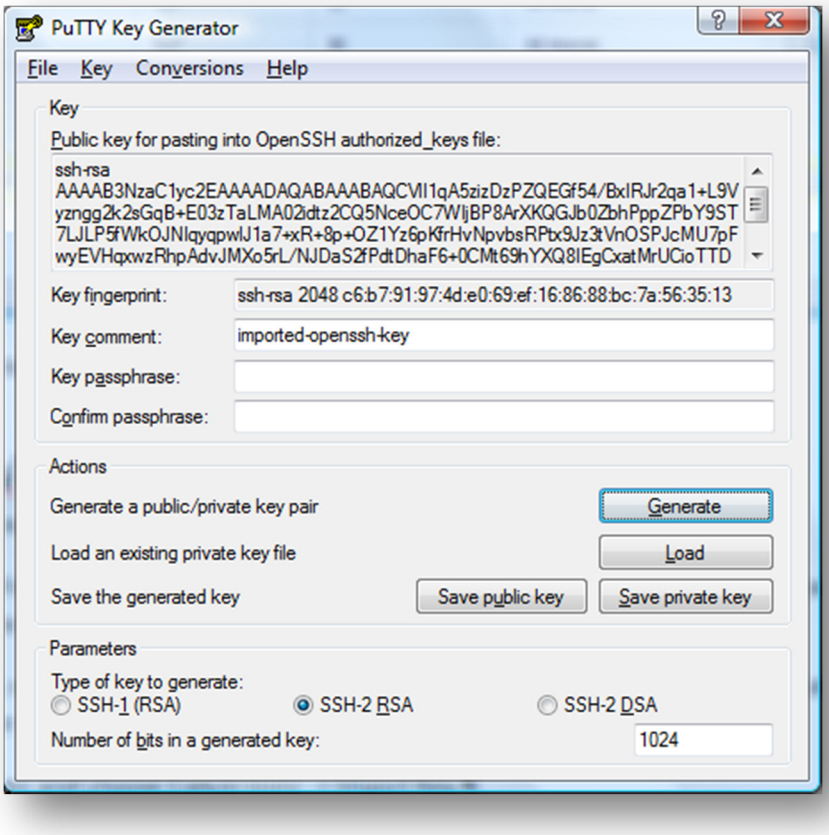


Appendix C – Using a 3rd Party SSH Client

Windows (PuTTY)

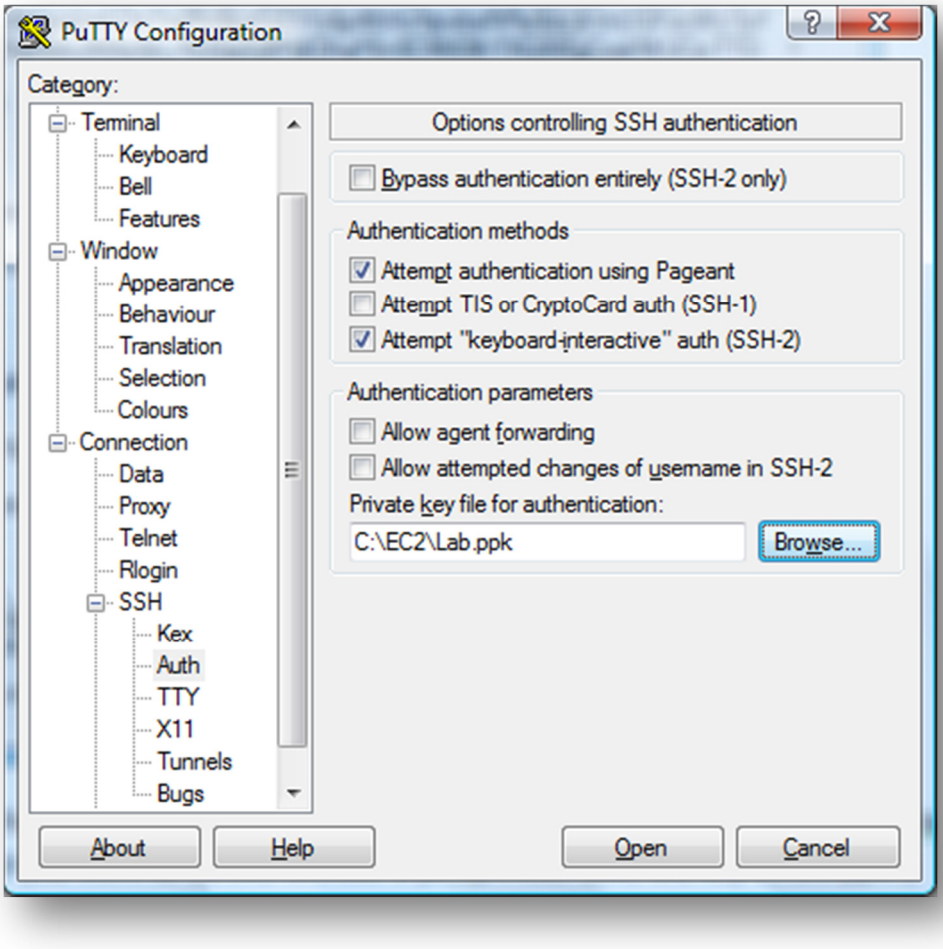
This is a Windows-only step, because other operating systems have SSH built in.
Download and install Putty. The single word “putty” in Google will return a list of download sites. Be certain that you install both Putty and PuttyGen

Launch PuttyGen and choose Conversions -> Import Key.
Browse for **Bootcamp.pem** and import the key. The result will look similar to this:

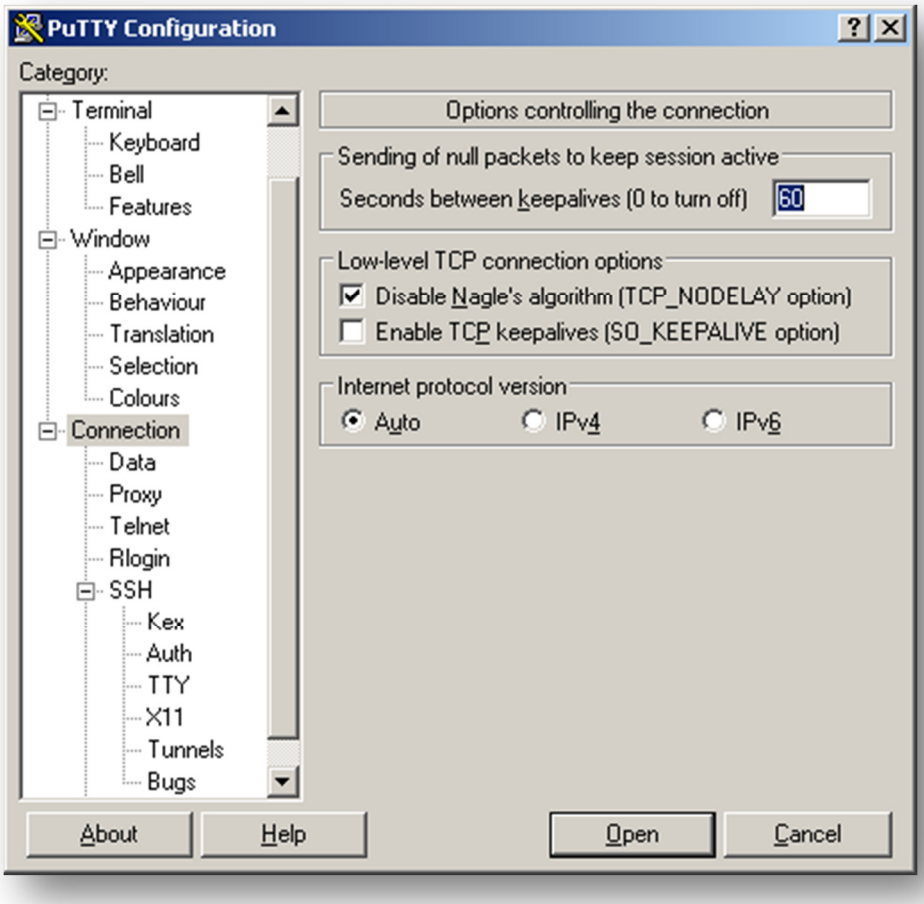


Save the key as the same file name with a .ppk extension (for the rest of these instructions, we will use the name Lab.ppk). Click on File -> Save as Private Key. Ignore the dialog that asks if you want to do this without a passphrase. Save the key as Lab.ppk.
Close PuttyGen.
Using Putty, login in via SSH as follows:

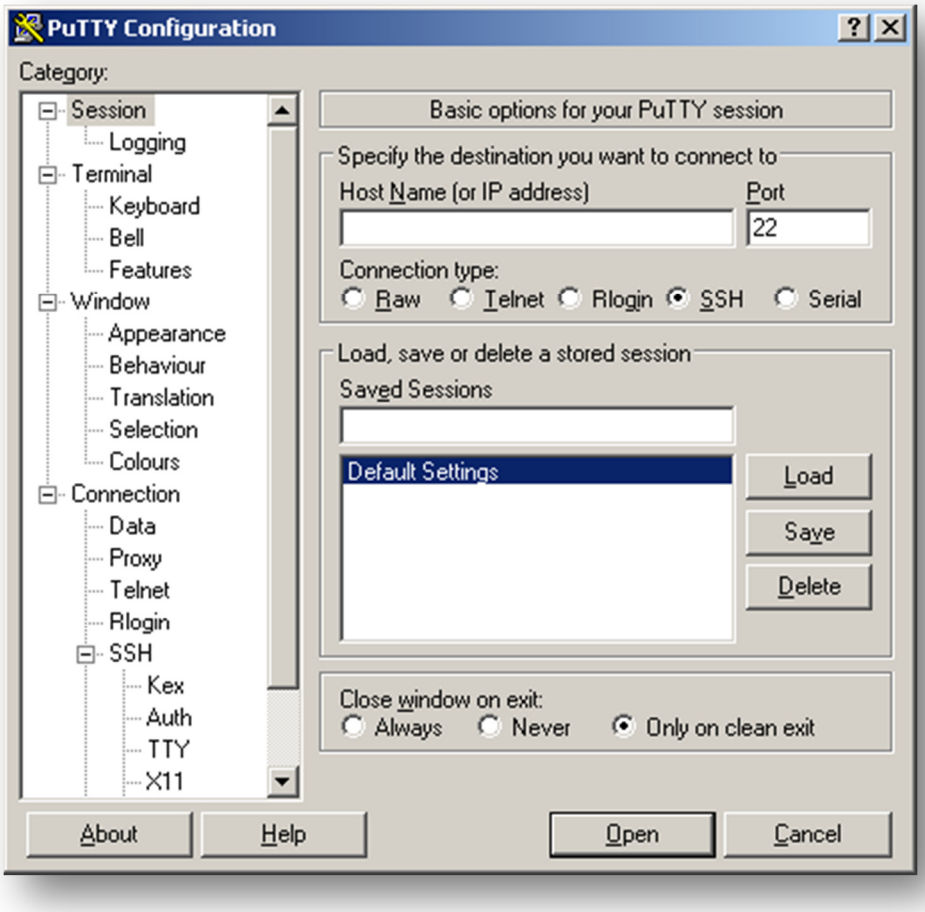
Launch Putty, then expand the SSH node and select the Auth sub-node. Enter Lab.ppk as the key name (shown below).



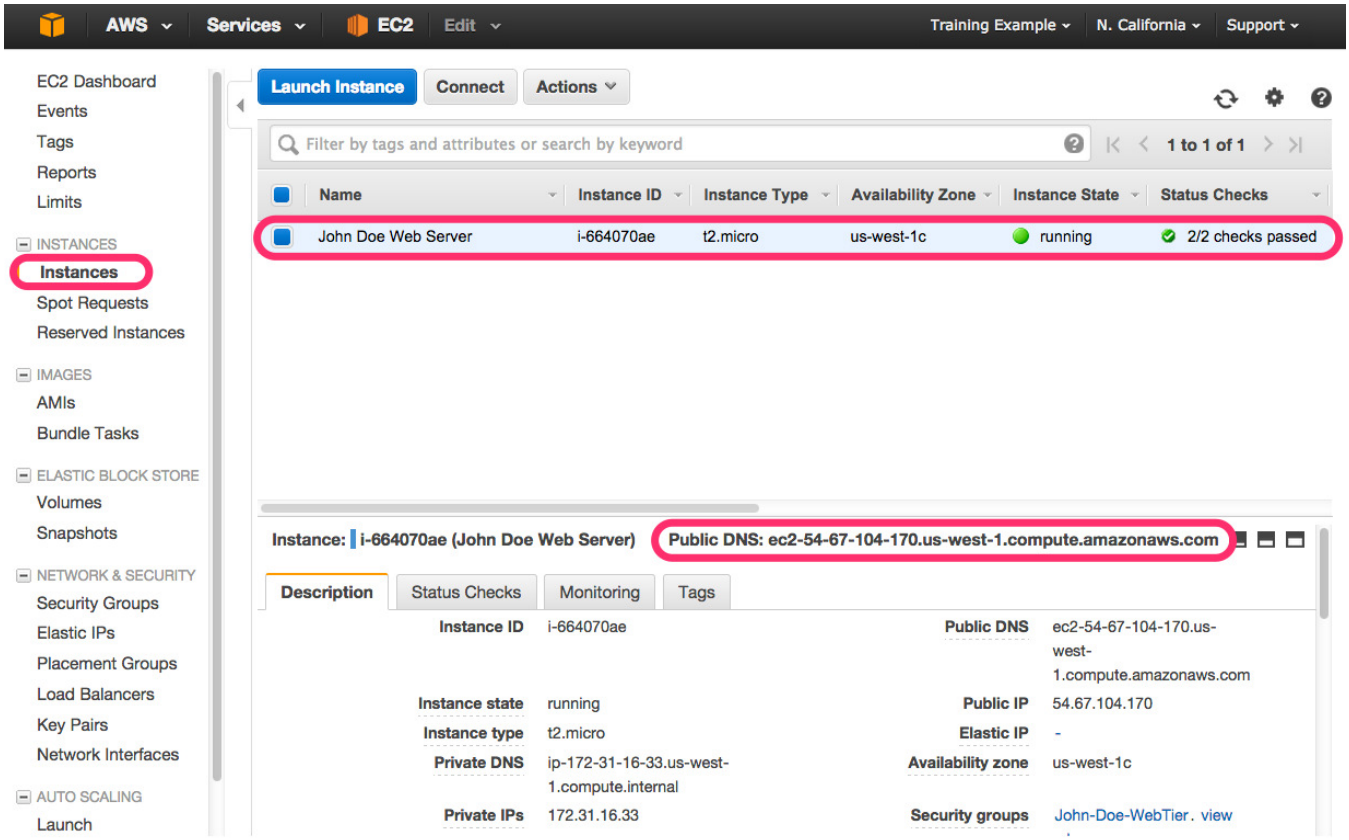
Make certain that *keepalive* has a value greater than zero. Otherwise your session will time out, which is annoying.



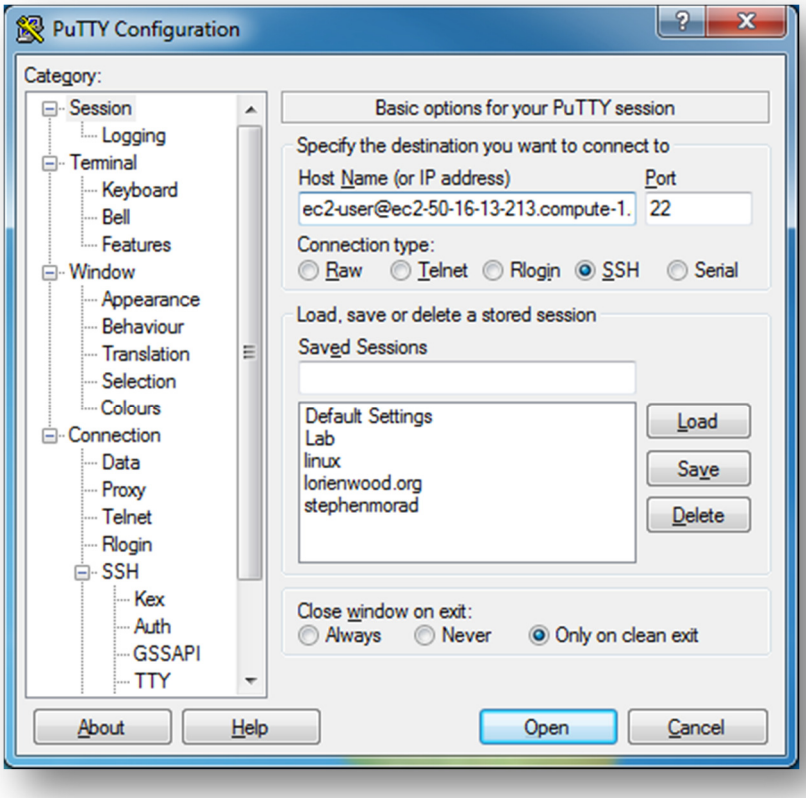
At this point (before entering the host address in the next step), it's a good time to save the settings. You can either highlight *Default* and update the settings, or pick a new name such as *Lab*.



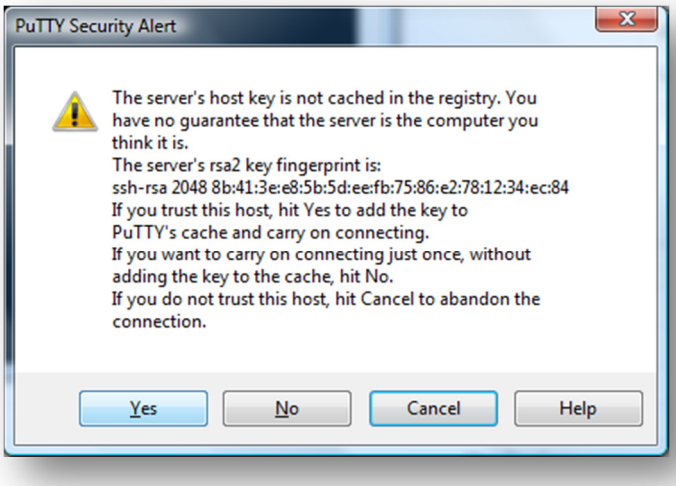
If you are not certain how to find the DNS name of the server, click on the running instance and look at the lower pane.



Find the Session node (top one in the list) and enter ec2-user@ followed by the DNS name of the running instance (you must initially login as "**ec2-user**" to Amazon Linux instances). Then click "Open" to connect. For example: ec2-user@ec2-50-16-13-213.compute-1.amazonaws.com

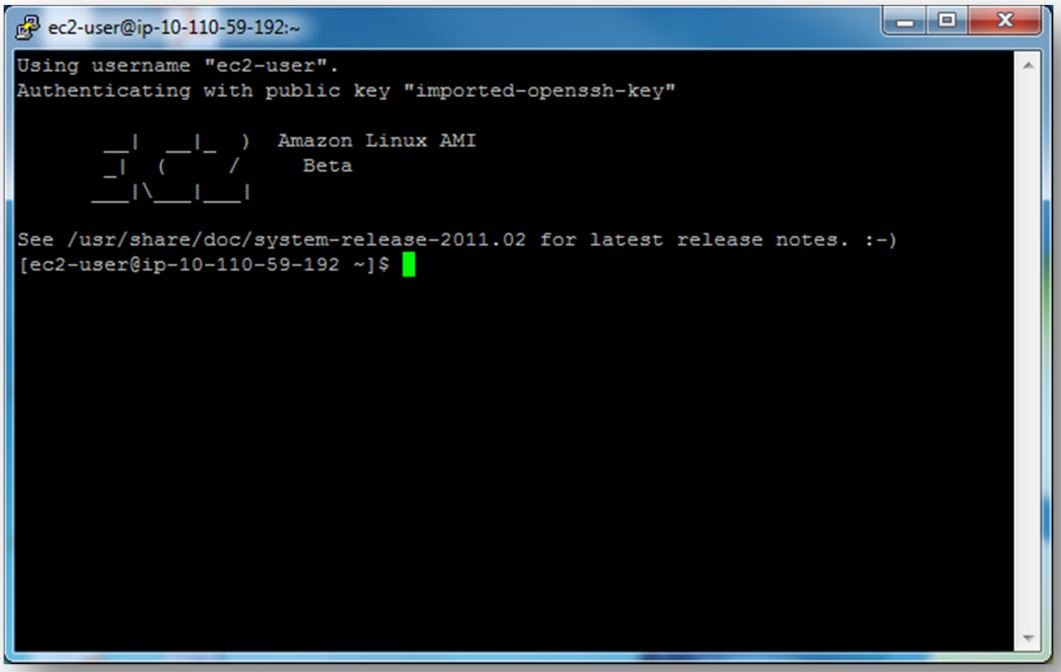


Click "Yes" to confirm that the fingerprint is OK.



Security Tip: The SSH fingerprint will eventually show up in the System Log and you can take that and compare it to protect against a Man in the middle attack.

You used the username “ec2-user”. The file Lab.ppk contains your password, so there is no need to enter one.



Mac OS X or Linux (OpenSSH)

By default, both Mac OS X and Linux operating systems ship with an SSH client that you can use to connect to your EC2 Linux instances. To use the SSH client with the key you created, a few steps are required.

1. Ideally, put the private key you downloaded while launching your EC2 instance (generic-qwiklab.pem) into the .ssh directory in your home directory. For example:

```
Prompt> mv qwiklab-l14-701.pem ~/.ssh
```

2. Make sure your private key is only readable and writable by you (this assumes your private key was copied into your .ssh directory as described above):

```
Prompt> chmod 600 ~/.ssh/qwiklab-l14-701.pem
```

3. Use your private key when connecting to the instance. The format of the ssh client is as follows: ssh -i <private_key> <user name>@<host name>

   Therefore connecting to your Amazon Linux instance will require a command similar to the following:

```
Prompt> ssh -i qwiklab-l14-701.pem ec2-user@<EC2 Host
Name or EIP>
```