Lab 1.0: AWS Setup

Objectives

The objectives for this lab are to set up the tools that will be required throughout the rest of the class.

- Create an Amazon Web Services (AWS) account.
- Secure the default root user.
- Create a new user for AWS labs.
- Set up AWS Command Line Interface (CLI).
- Set up an AWS SSH keypair and install PuTTY if needed.
- Set up Git.
- Configure Virtual Machines.

This lab needs to be completed before you will be able to complete subsequent labs. Please ask for help right away If you need it! For OnDemand students, please email online-sme@sans.org.

Prerequisites

In order to set up an AWS Account and test your access, you will need the following:

- Credit card to set up AWS account
- A phone to verify your account via a phone call from AWS
- A laptop or virtual environment where you can install new software
- Internet access with firewall ports and proxies open to access AWS services
- At least 40 GB of free space for the files that will be copied or downloaded to your local machine

Overview

The tools and configuration in this lab are required to complete the labs throughout the remainder of this course. Please set up and test the tools as explained in the appendices below prior to starting other labs.

Appendix A: Create an AWS Account

Appendix B: Secure the Root User

Appendix C: Create an AWS User

Appendix D: AWS CLI Setup

Appendix E: SSH on AWS

Appendix F: Setup Git

Appendix G: VM Networking

Tip: Create a folder for the class where you can store all the related files created in the labs below (key and credential files, virtual machines, and any notes you take during the process).

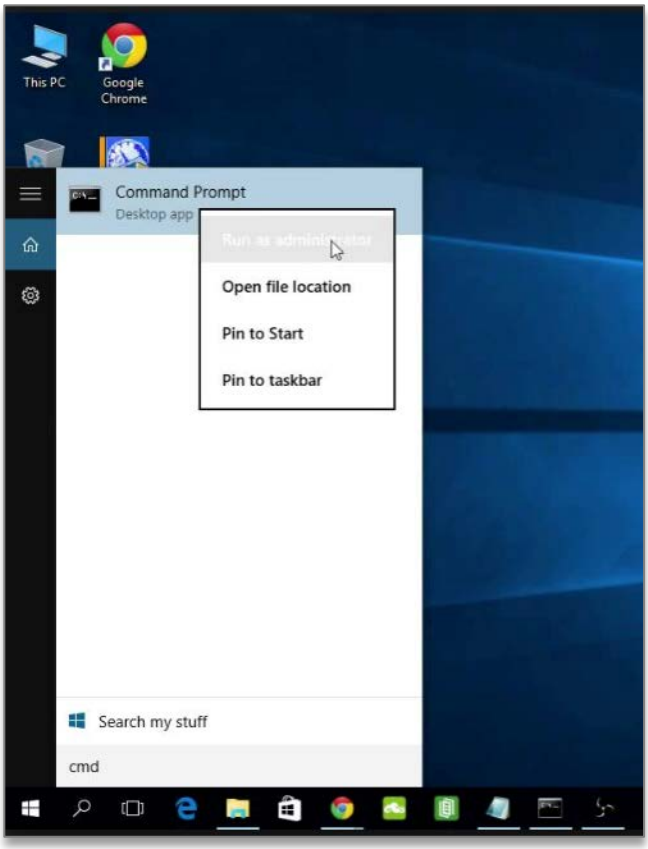
Command Line

When you see text in courier font like this is, it indicates a command that should be typed into a command line window. The commands will take actions on your computer or a remote computer. For example, you can type the command below on any system used in this class at a command prompt to print out the text to the screen.

```
echo “This is a sample command typed at the command line”
```

On a Mac, click on the search magnifying glass on the right and type “**terminal**,” then hit enter. Linux will usually have a terminal icon on the desktop or in the start or app menu— this may vary depending on the version of Linux.

Make sure you know where to type these commands. On Windows, go to the start menu, type “**cmd**,” and hit enter to open a command prompt or click on the “Command Prompt” icon shown in the screenshot below. Sometimes you need to run a command with administrator privileges. Right-click on the cmd icon and choose “Run as Administrator” to open the command prompt with the administrator privileges. This may be the preferred method, as some commands may require administrative privileges in the labs.



In Linux, you can use the **su –** to switch to the root account but you need to know the root password. You can use **sudo** to run a single command with root privileges if your account is in the sudoers file. You can also use **sudo su –** to get an interactive shell if you don't know the administrator password.

```
Gemini:/ daveshackleford$ su -
Password:
Gemini:~ root#
```

On a Mac, you may only be able to use **sudo** because the administrator account is disabled by default. You can also use sudo -i on a Mac to get an interactive root shell.

```
2SL:~ tradichel$ sudo -i
Password:
2SL:~ root#
```

Amazon Web Services (AWS) Account

You will need an AWS Account in order to complete the labs. We will follow some best practices when setting up this new AWS account. We recommend you use a new AWS account for this class to avoid conflicts while running the labs. You can delete the account after class is over.

Please turn to Appendix A and follow the instructions to create an AWS account.

Secure the Root User

One of the AWS IAM (Identity and Access Management) practices on AWS includes setting up multi-factor authentication (MFA) on the root user account, any high privileged user, and ideally any AWS user. Additionally, the root user should have no programmatic access (that is, access via the AWS command line and other API-based methods).

Please turn to Appendix B to secure the root user account.

AWS Lab User

Another AWS best practice is to create a separate account from the root user account and only use the root user account to take actions that require it. The root user account is very powerful with the ability to change the billing, contacts, and permissions, and delete the account altogether. When creating this new user, we will create the **Access Key Id and Secret Key** that will be used in labs throughout the class. Make sure you create and save these credentials for later use in the class.

Please turn to Appendix C to create a new user named SEC545.

AWS Command Line Interface (CLI)

The AWS cloud platform allows users to create resources, query the platform, and take actions on resources in the environment programmatically. One of the tools that can be used to access AWS programmatically is called the AWS Command Line Interface, most often referred to as the **AWS CLI**. The AWS CLI needs to be installed and configured with the AWS credentials created in Appendix C.

Please turn to Appendix D and follow the instructions to install and/or configure the AWS CLI.

SSH on AWS

SSH commands on AWS are the same as any other environment. The default user will be "**ec2-user,**" and you will use an SSH key to connect. When deploying a Linux virtual machine (called an EC2 instance) on AWS, users are given the option to use an existing SSH keypair or create a new keypair. If you are using an older version of Windows, you will need to use PuTTY to connect via SSH to an instance on AWS. We will create an SSH key, called an **EC2 keypair** on AWS, that will be used to access the Linux instances we create throughout the class.

Please turn to Appendix E and follow the instructions SSH setup instructions.

Git

When developers write code, it is typically stored in a source control system that tracks versions of the files as they are updated and allows developers to roll back to a prior version of the code if necessary. One popular source control system is called GitHub (github.com). In order to interact with GitHub programmatically, developers will use a tool called Git. We will be using Git and retrieving files from GitHub in some of the labs.

Please turn to Appendix F and install Git.

Virtual Machines

Some Virtual Machines (VMs) are included on the USB for this course. If you do not already have a way to run virtual machines on your system, you'll need to install VMWare Player or, for more functionality, VMware Fusion (Mac) or Workstation (Windows). The networking on the virtual machines needs to be configured correctly for the labs as well. Note that you may not have your USB if you're setting this up before class, and this Appendix exercise may not be fully completed when class starts. At a minimum, make sure you have VMware installed and operational.

Please follow the instructions in Appendix G to run the virtual machines and test the networking.

Summary

You should have the following after finishing the instructions above:

- An AWS account
- Root user with no Access Keys and MFA
- A new user with Access Keys and MFA
- A working AWS CLI configuration
- An EC2 Keypair (for SSH)
- PuTTY installed and configured if needed
- Git installed
- Virtual Machines configured, and networking verified

After completing the steps above, you should be set up to run the labs!

Appendix A: Create an AWS Account

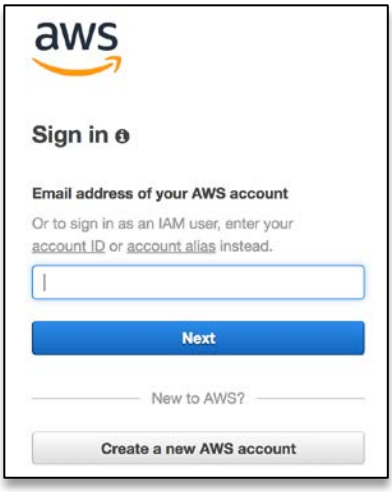
We recommend creating a new AWS account. Using an existing account may lead to networking conflicts and other problems in future labs. You can cancel the account at the end of the class if you want.

1: Navigate to aws.amazon.com and click the button on the top right.

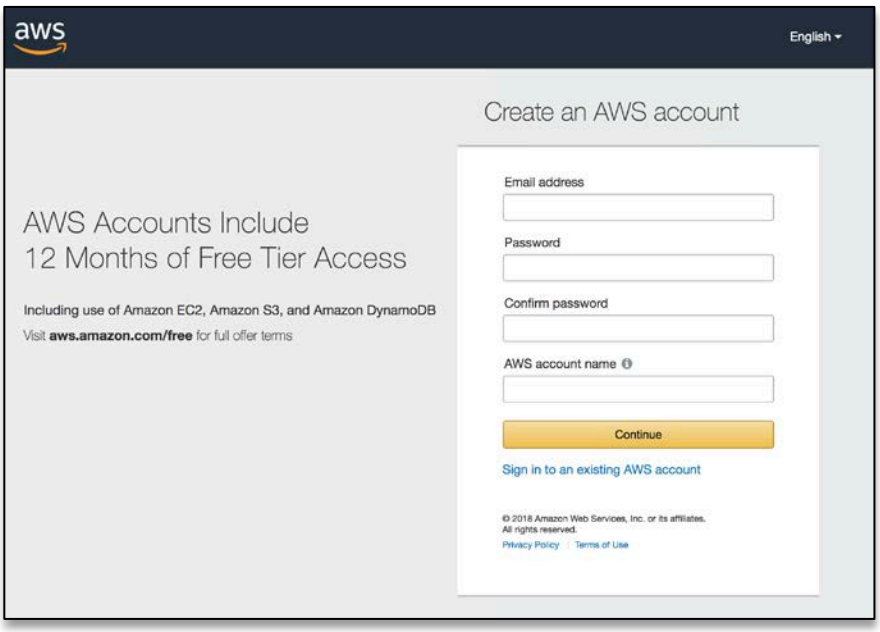
Note that the text on the button may be different depending on whether or not you have visited the site or created an account previously. It could say "Sign up" or "Create an Account" or "Sign In to the Console."

2: Follow the instructions to create a new account.

Note: If you are presented with a login screen, click "Create a new AWS Account."



Otherwise you should see a screen such as the following right away:



Note: To complete the process, you will need to enter a credit card number and receive a phone call from Amazon to enter a code. It will ask you if you want to create a support plan. Just select the free, basic support plan to avoid additional charges.

Best practice when setting up a new AWS account for an organization would be to use an alias that will remain valid even if the person setting up the account leaves the company. For example, for an account set up for the pen testing division of 2nd Sight Lab, the alias might be aws-pentesting@2ndsightlab.com and emails might go to the person in charge of the pen testing division, CIO, and a person responsible for AWS billing. For now, just use any email you want.

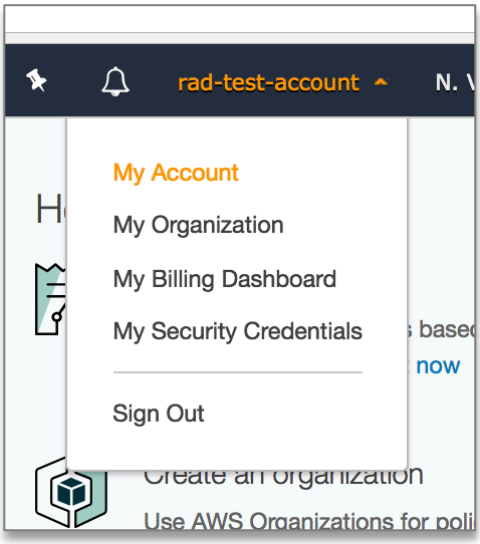
Tip: If you start your email aliases with "aws" they will follow the instructions on each screen until you have successfully created an account.

Note that you will have to validate your account by responding to a phone call. If you are setting up an account outside the US, you may have additional steps provided to you by Amazon.

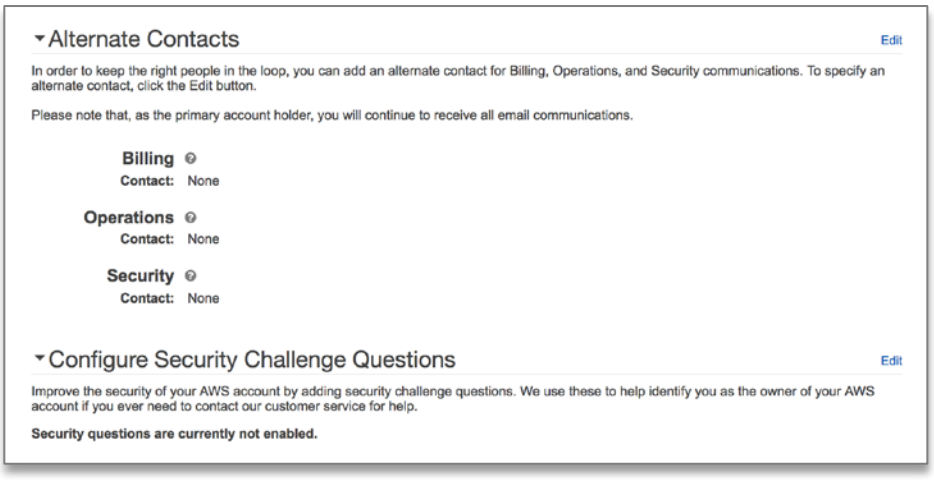
After completing the process, you will be returned to the login screen at the end of the account creation process.

3. Sign into your new account by clicking "Sign In to the Console" at the top of the page.

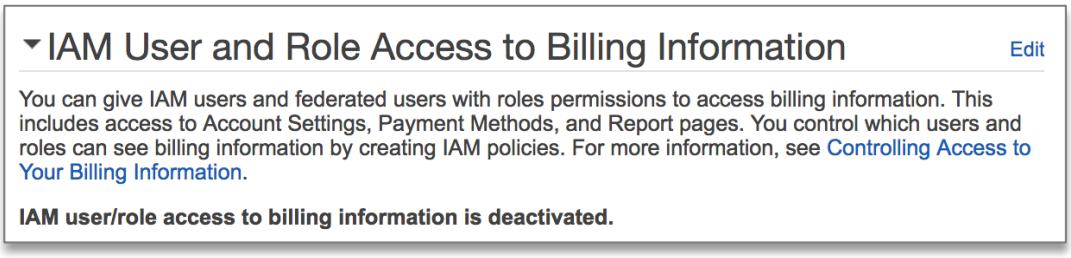
4. Take a look at the billing information by clicking on the account name you selected. It will be on the right side of the black bar at the top of the screen. Click on "My Account."



This section shows where you would set up alternate billing contacts and create challenge questions.

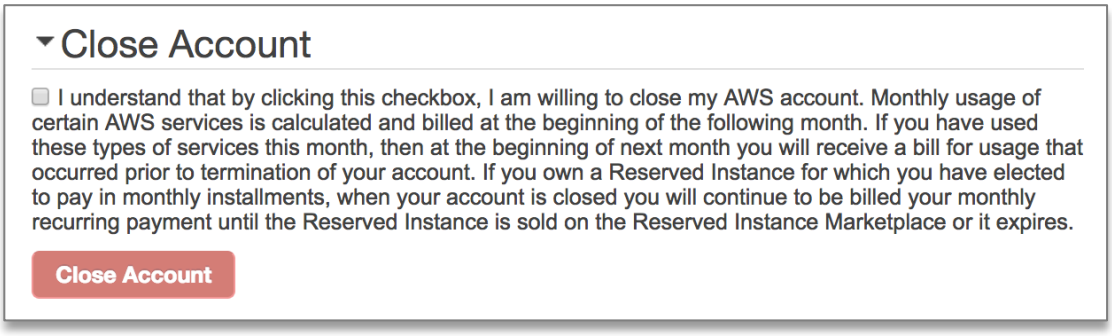


This section allows turning on or off access to billing for anyone but the root user in the account. Large companies may want to consider using AWS Organizations and Consolidated Billing to segregate billing responsibilities into a separate account. For more information about organizations see: **https://aws.amazon.com/organizations/**



This screen is also where companies can sign up for GovCloud if part of the US Government.

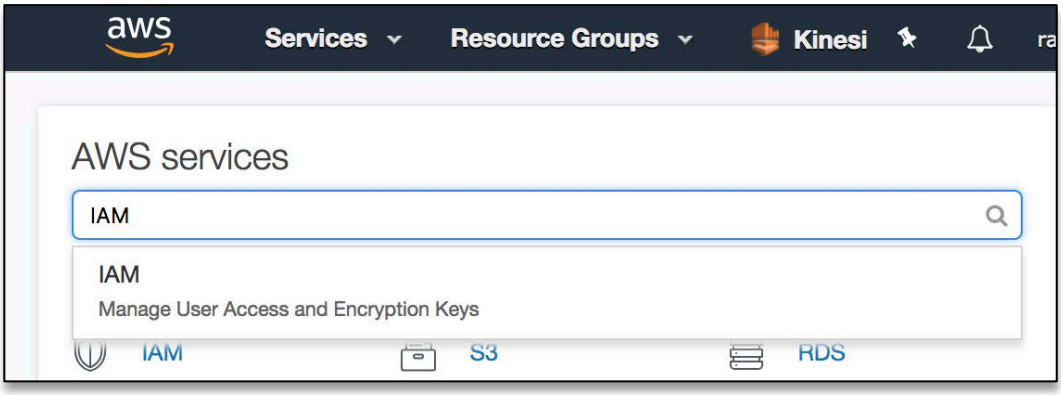
You can close the account after the class is over if you don't want to keep it or get charged additional fees outside of what is done in the class on this screen.



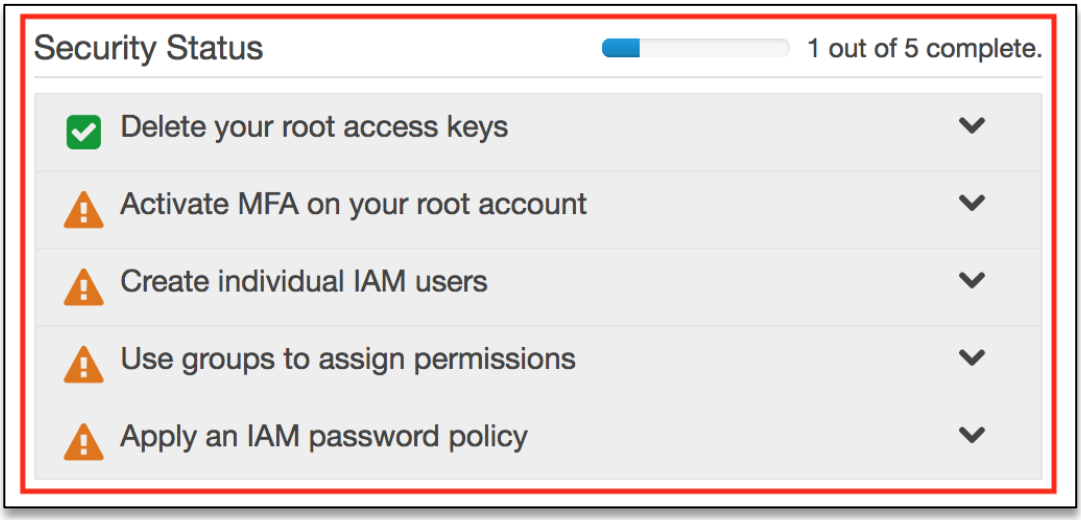
Appendix B: Secure the Root User

AWS best practices include some immediate changes to protect the default root user. These protections are added via the IAM (Identity and Access Management) service in AWS, which allows you to manage access to AWS services in your account.

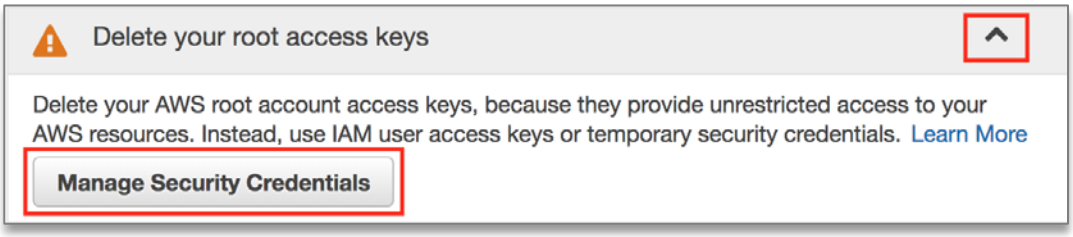
1. Navigate to the IAM service. Click on the AWS Logo. Type IAM in the search box and click IAM.



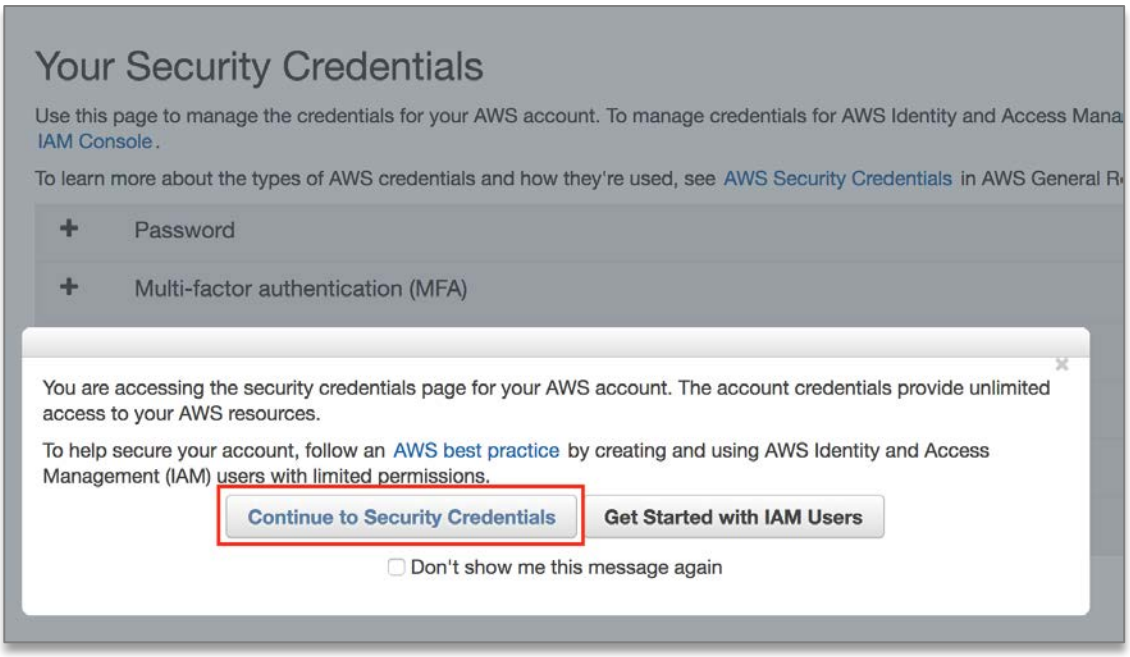
2. Verify there is a green checkbox next to "Delete your root access keys."



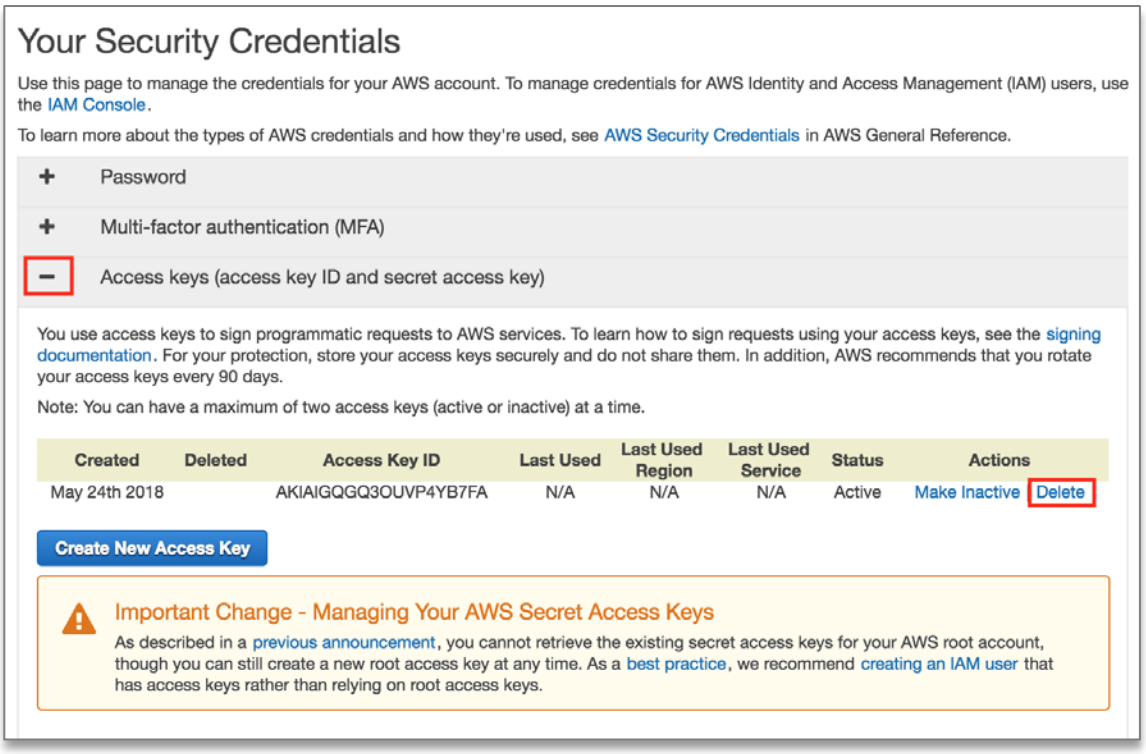
2a. If you see a yellow triangle, click on the down arrow and then click "Manage Security Credentials."



2b. Click on "Continue to Security Credentials."



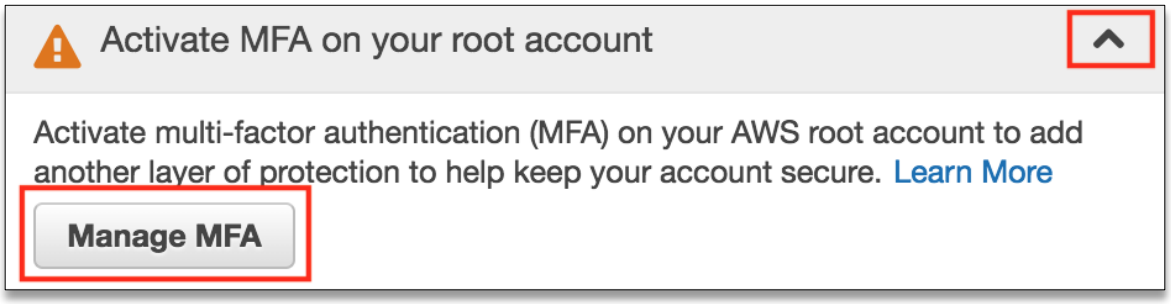
2c. Click the plus sign (+) next to Access keys and click "Delete" then the back arrow in your browser.



Now you should see the green checkbox as shown in step 2.

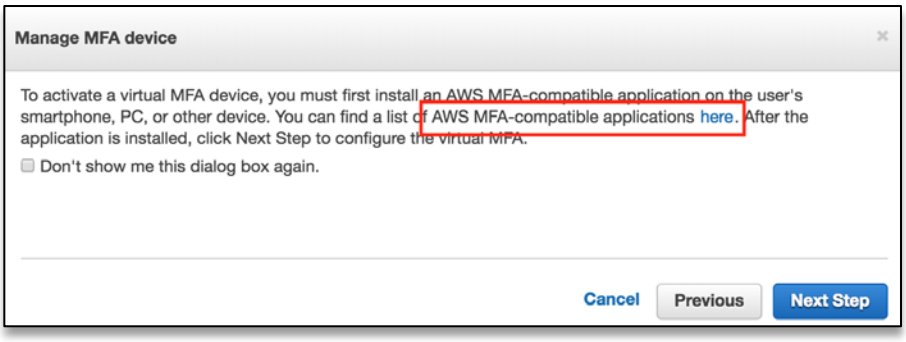
3. Set up MFA

AWS security best practice includes setting up MFA (multi-factor authentication) on your root account. The root account can cancel your account, change the contract information, and delete resources from your account. It is important to set up MFA (multi-factor authentication) on this and any other user in your account that has a great deal of administrative access. Click the down arrow next to "**Activate MFA on your root account,**" click "**Manage MFA,**" and follow the instructions.



4. Choose "A virtual MFA device" (your cell phone) and click Next Step.

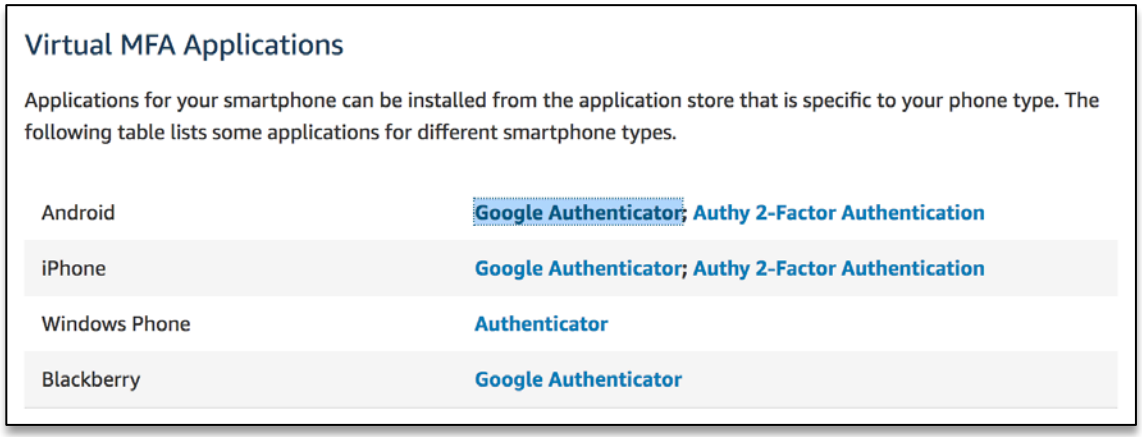
5. Follow the link to the instructions to set up an application on your phone to use for MFA.



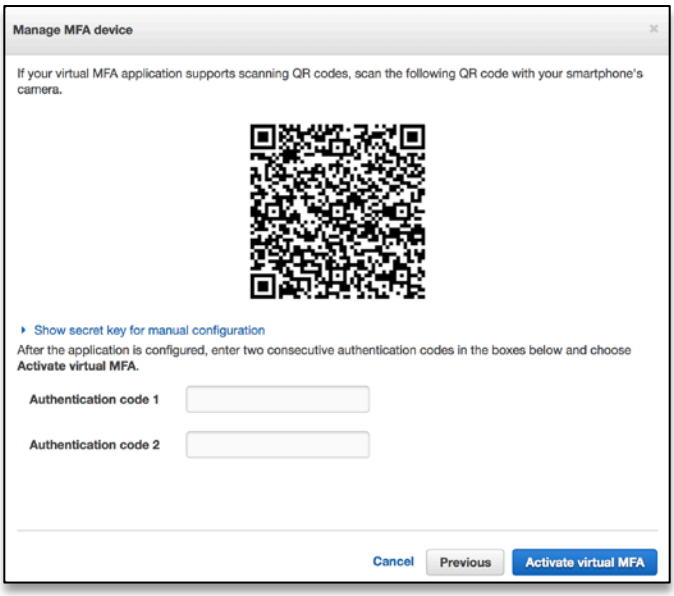
6. Scroll to the middle of the page and choose an application that works with your device.

Follow the instructions to install the application on your device.

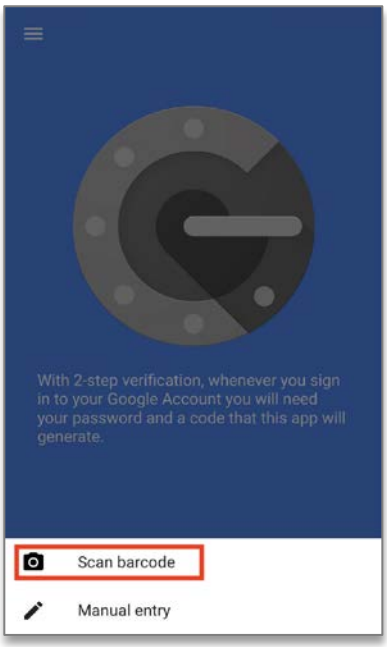
Note: Google Authenticator is likely the most popular choice at the time of this writing. Usually it's just a matter of going to the app store on your device, searching for "Google Authenticator," and choosing to install it.

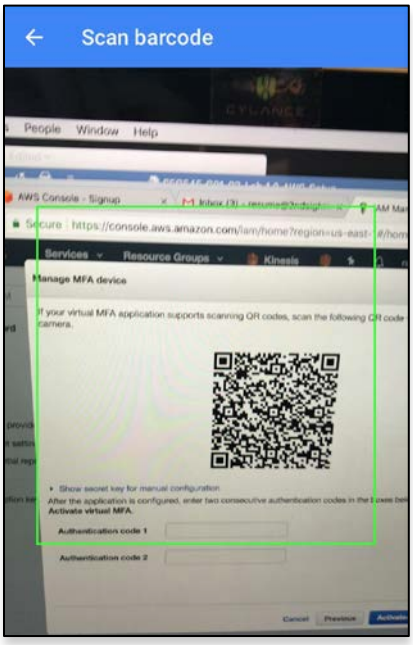


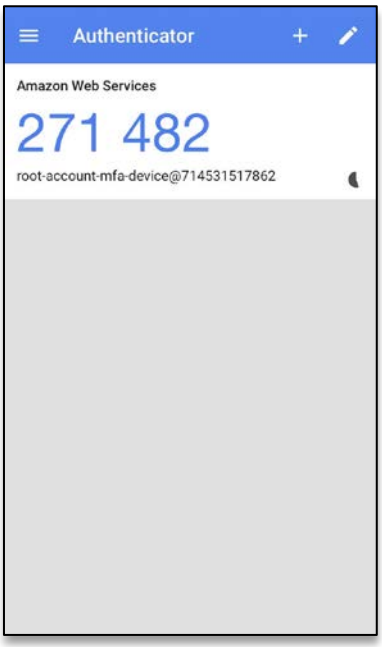
7. Click Next Step in the AWS Console.



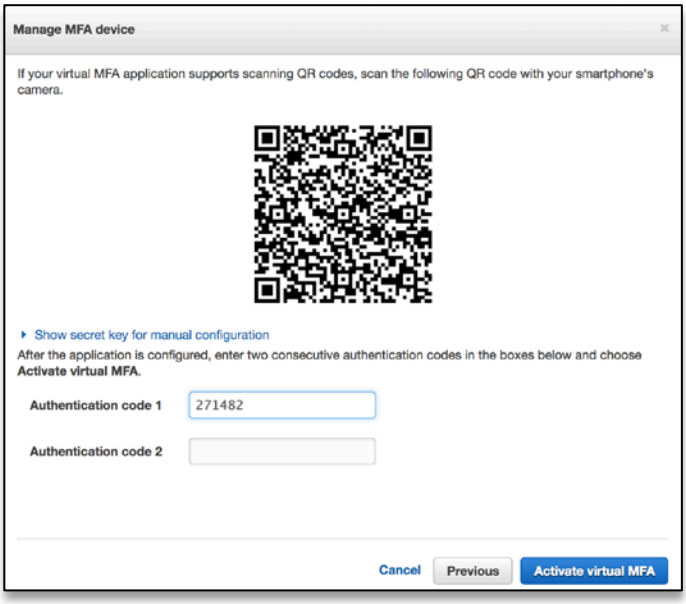
8. Scan the QR Code to add your AWS account to the authenticator app.







9. Enter the numeric code from the authenticator into the AWS Console. Then wait for a new code to appear in the authenticator. Enter the second code. Then click "Activate virtual MFA."

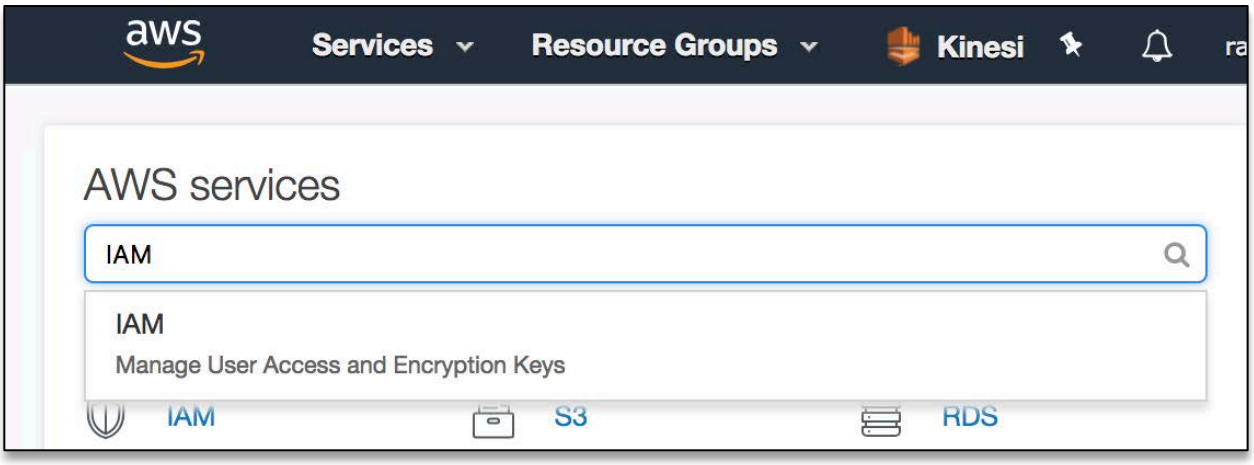


At this point, you have added MFA to your root account. From now on, you'll need to enter the code from the authenticator application on your phone in order to log into the account. Note that this QR code should be protected. Anyone who can access the QR code can add it to their own authenticator application. Some companies print out the QR code, store it in a safe, and require two people to access it to log in as the root user of an account.

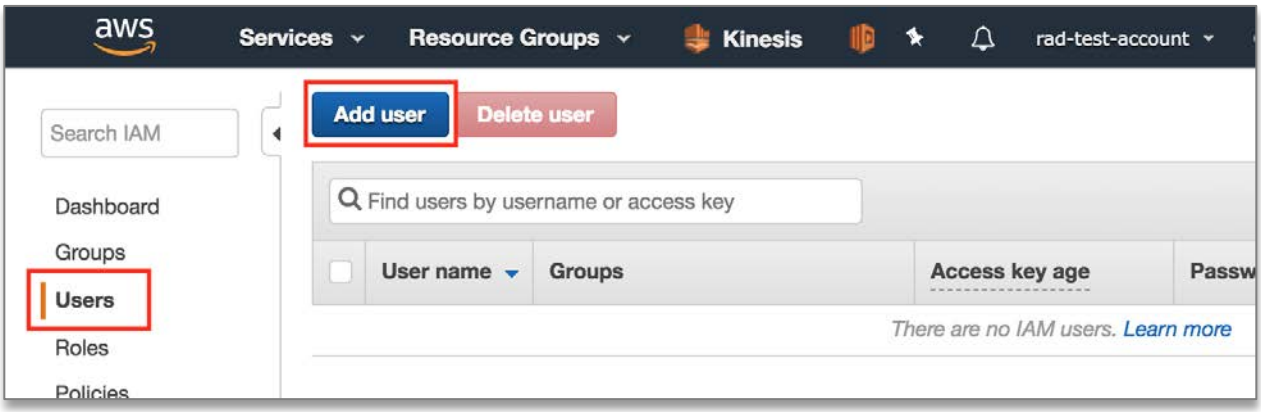
Appendix C: Create an AWS User

Best security practices for AWS IAM include not using the root account but instead setting up a new user and then using separate user accounts for normal operations. Lock away the root account access in a safe manner. *SANS SEC401 Security Bootcamp* discusses various mechanisms for securely storing sensitive, high-risk credentials.

1. Navigate to the IAM page in the AWS Console. Click on the AWS Logo. Type IAM in the search box.

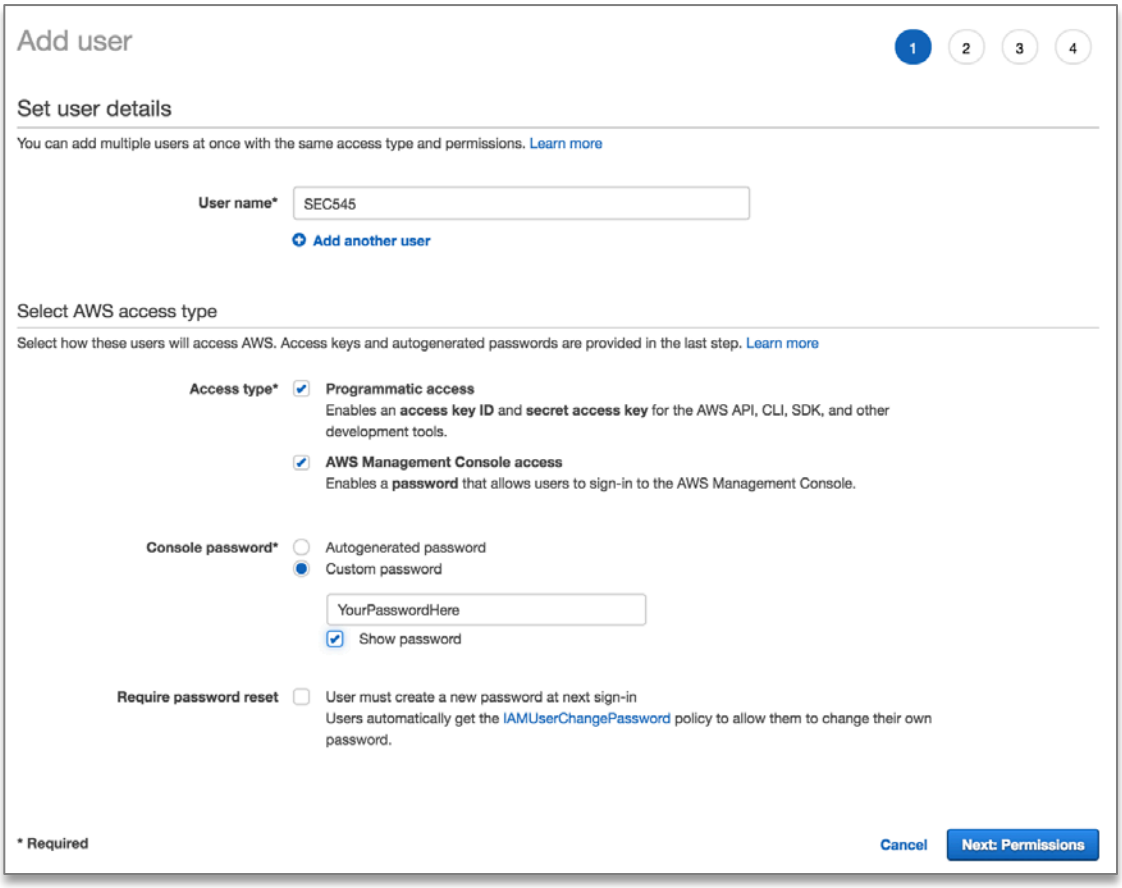


2. Click on "Users" on the left and then click "Add user" at the top of the screen.



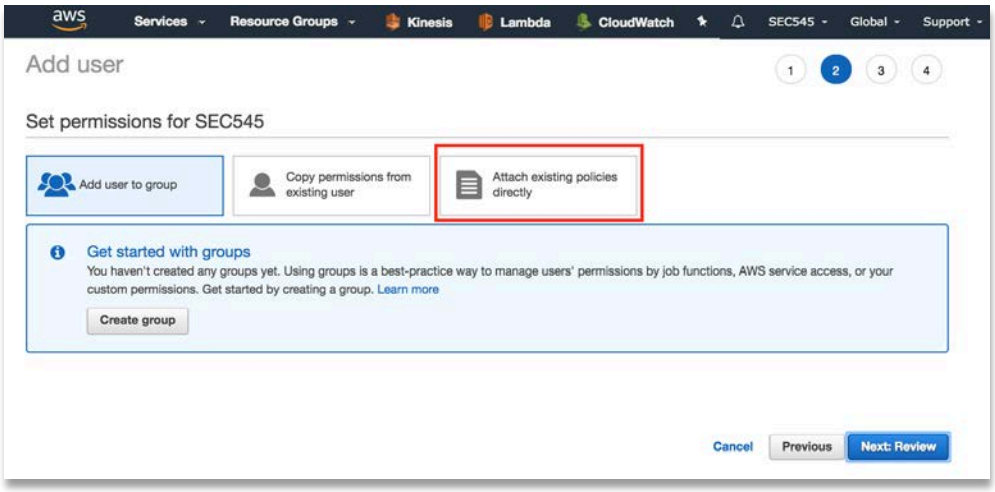
3. Add a new user with the following settings.

Note the name of the user is **SEC545**. This is the user account you will be using in the course labs to perform actions in your AWS account. Change **YourPasswordHere** to a password you can remember. Also, be sure that both "Programmatic access" and "AWS Management Console access" are checked.

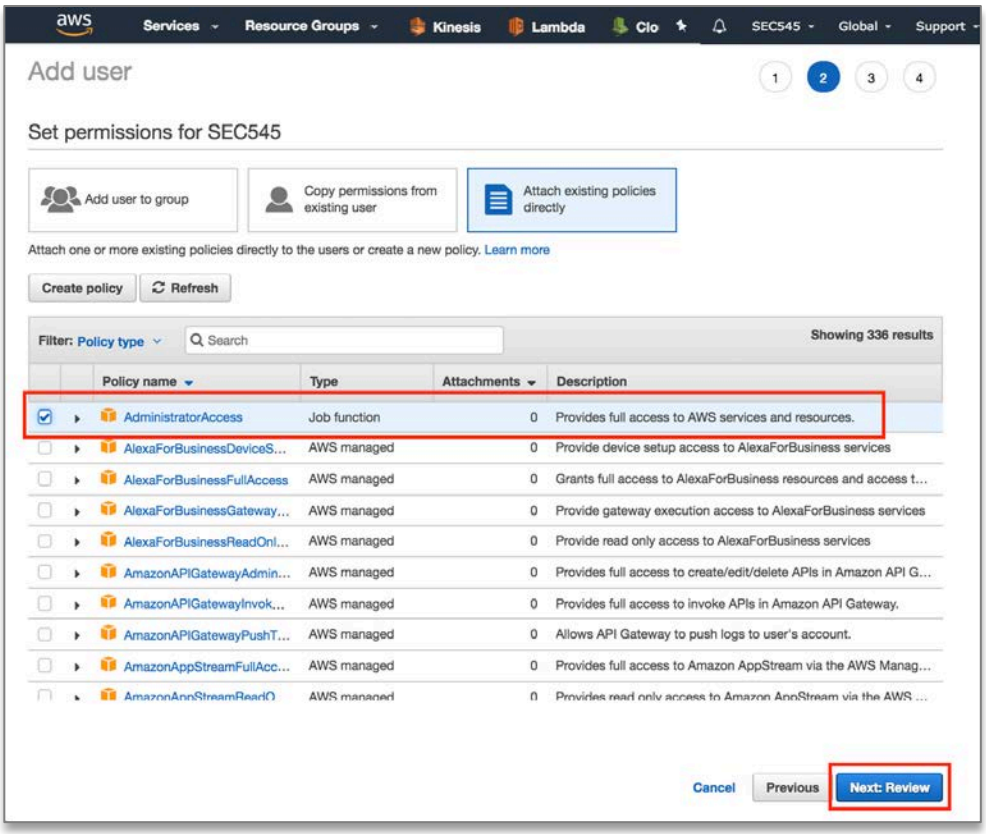


4. On the next screen, choose the last box, "Attach existing policies directly."

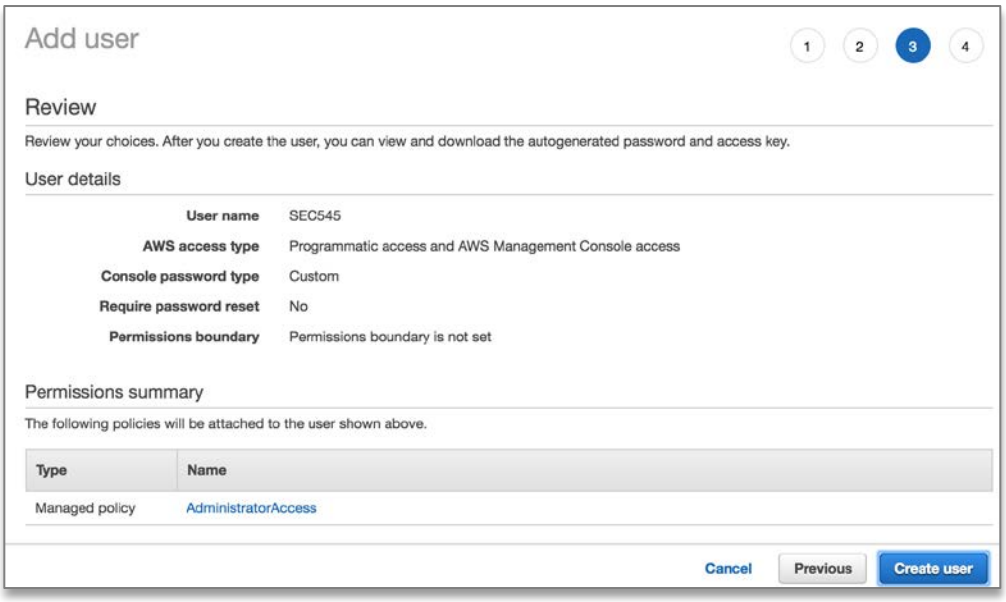
This will be explained in more detail later in the course.



5. Select “AdministratorAccess” and click “Next: Review.”

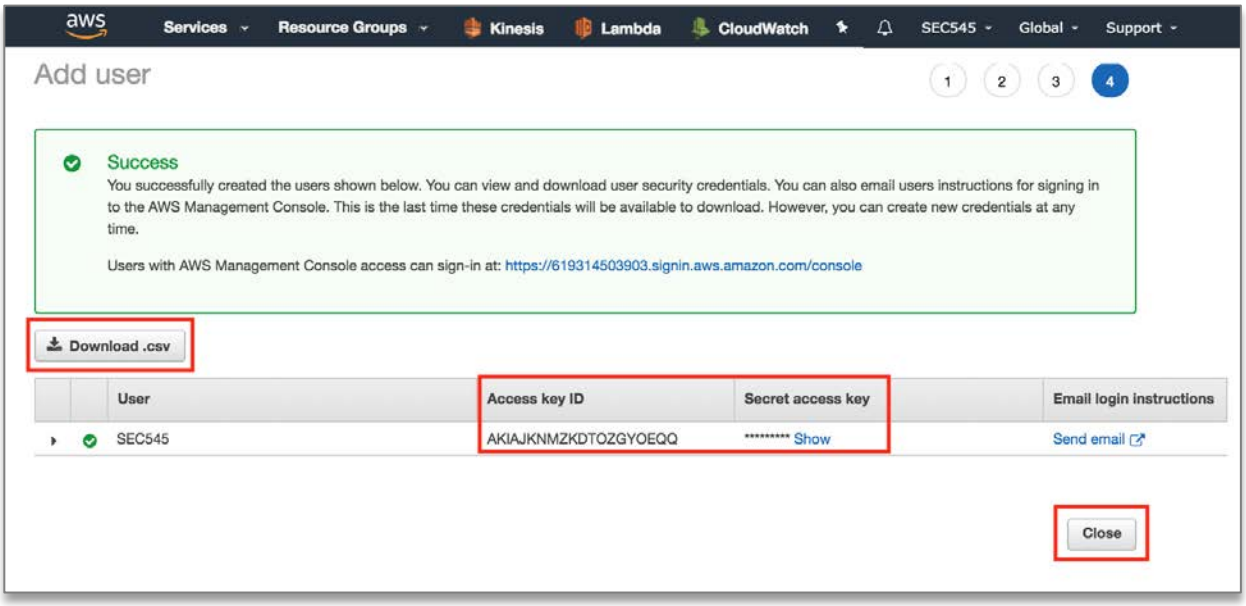


6. Review and click “Create user.”



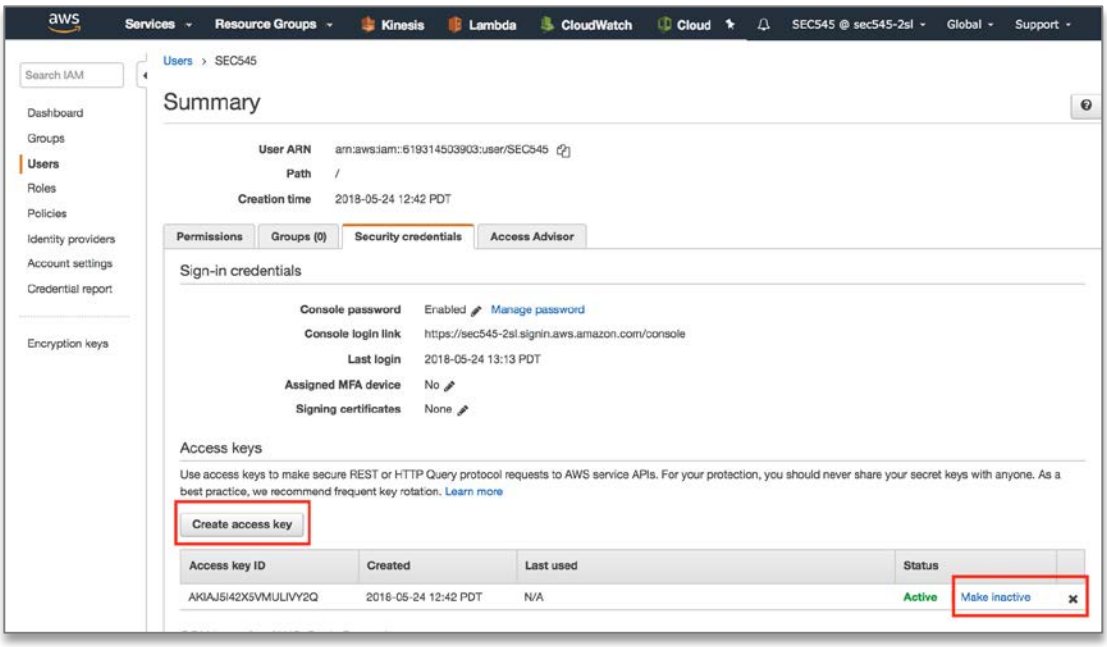
7. Save the Access key ID and Secret access key. You won't see them again!

After the user is created, you will be presented with a screen that provides you credentials. You will need these credentials later to run programmatic actions on the AWS Cloud. These credentials include an Access key ID and Secret access key. Once you leave this page, you will not be able to see these credentials again (though you can delete these and create new ones). You'll want to capture this information before proceeding. You can click the "Download .csv" button to get a file containing the credentials. Alternatively, you can click the "Show" button to view the credentials and copy them off the screen. Save them for later. Then click "Close."

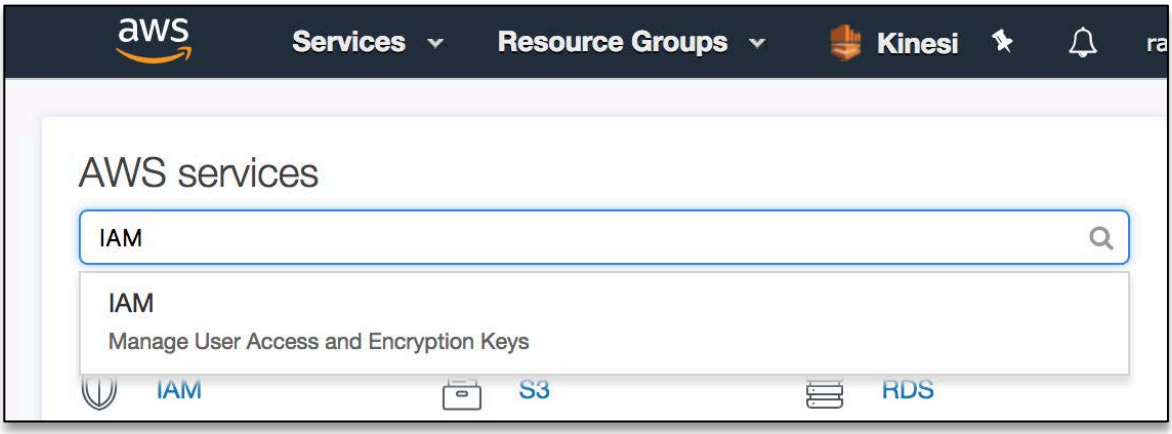


Best Practice Note: Since you are the only one using this user account, you can capture these credentials now. If you were setting this user account up for another person, the best practice would be to delete these credentials and then provide instructions to the user to set these credentials up later after they log into their account. If you do not take this precaution, then you cannot guarantee that actions in the logs using these credentials were performed by the person associated with this user account, since multiple people had access to them.

To create a new access key and secret key for a user, click on the user in the user list, then click on the security credentials tab (shown below). Then click the “Create access key” button. You can also make any existing access keys inactive or delete them.

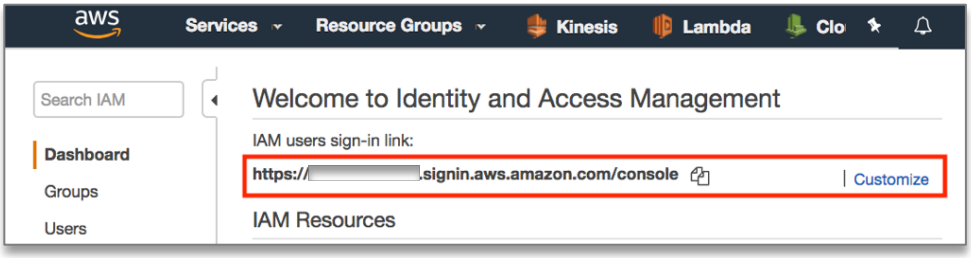


8. Navigate to the IAM service. Click the AWS Logo, search for IAM, and click on IAM.

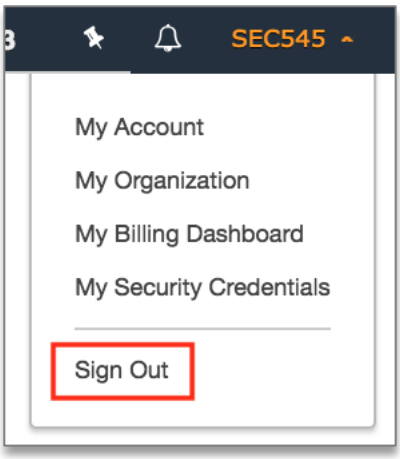


9. Use the IAM users sign-in link to log in as the user you just created. Customize it if you want.

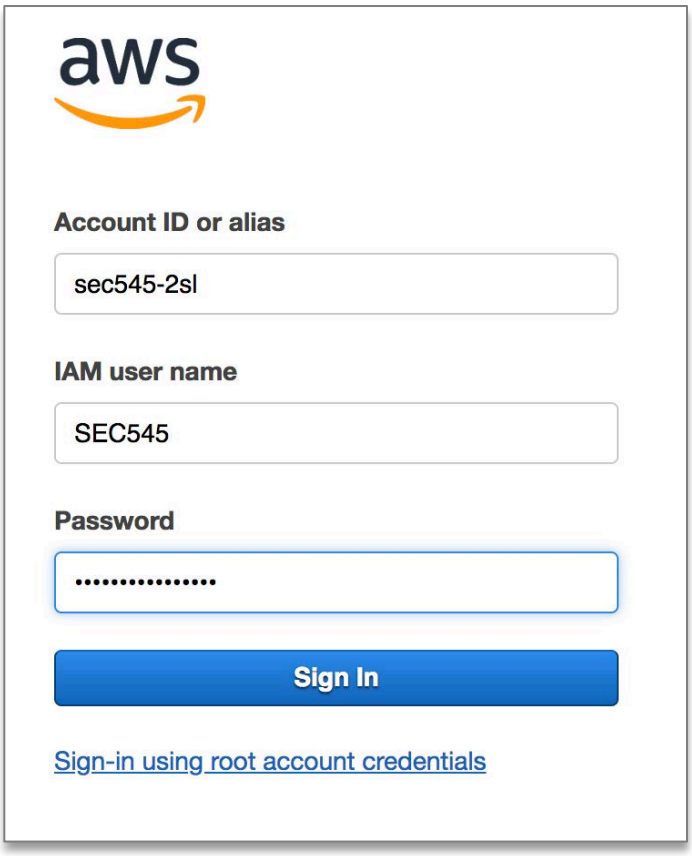
Hint: You might want to create a bookmark for this link!



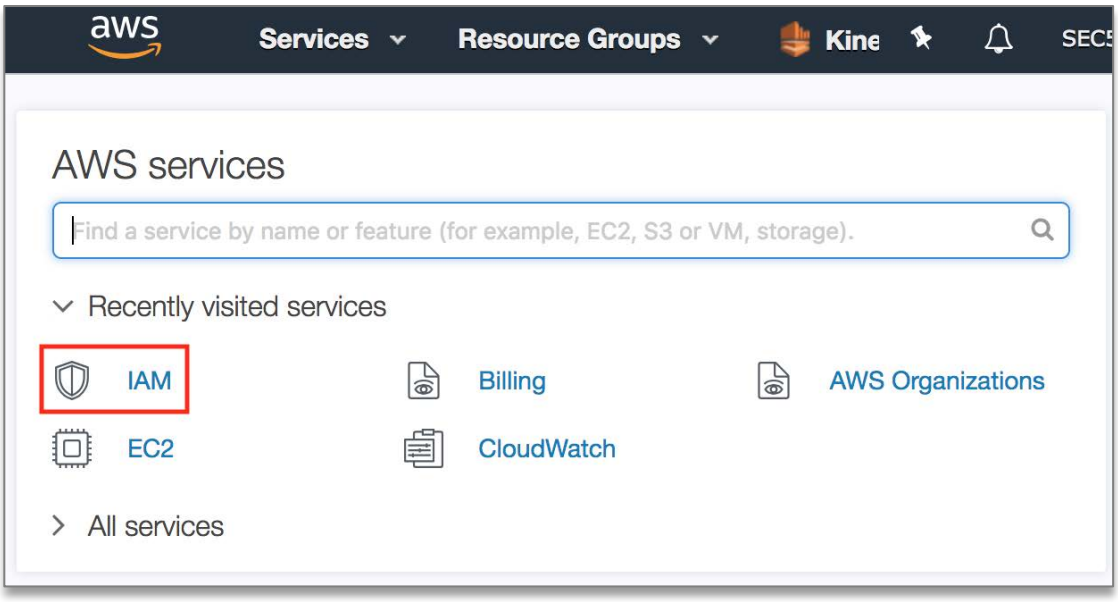
10. Sign out of your AWS Account.



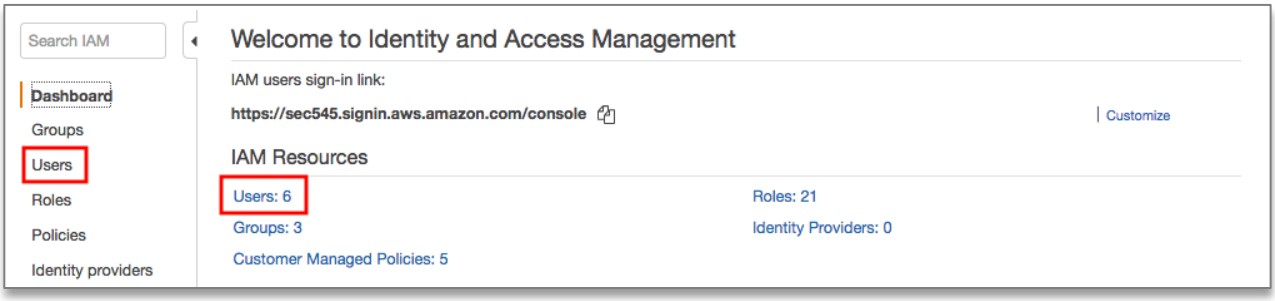
11. Sign into your AWS account at the URL in Step 9 with the new user credentials.



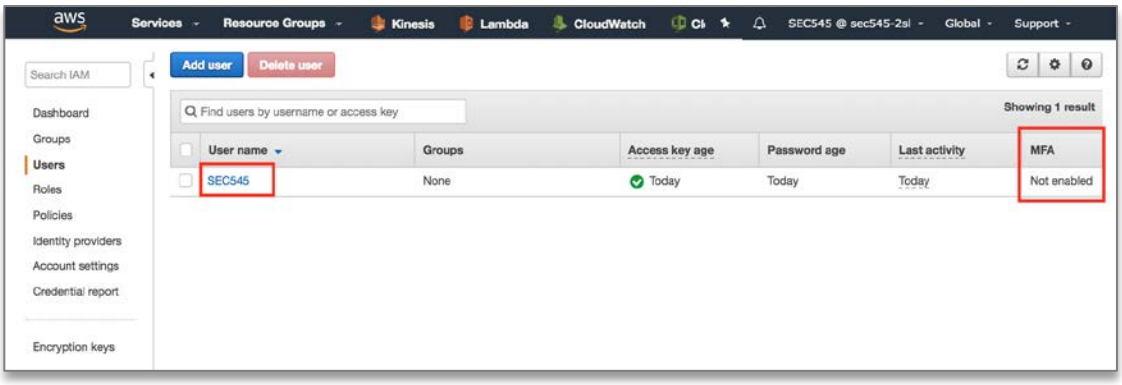
12. Navigate to the IAM service by searching for it or clicking the IAM icon.



13. Click on Users on the left or in the middle of IAM Dashboard.

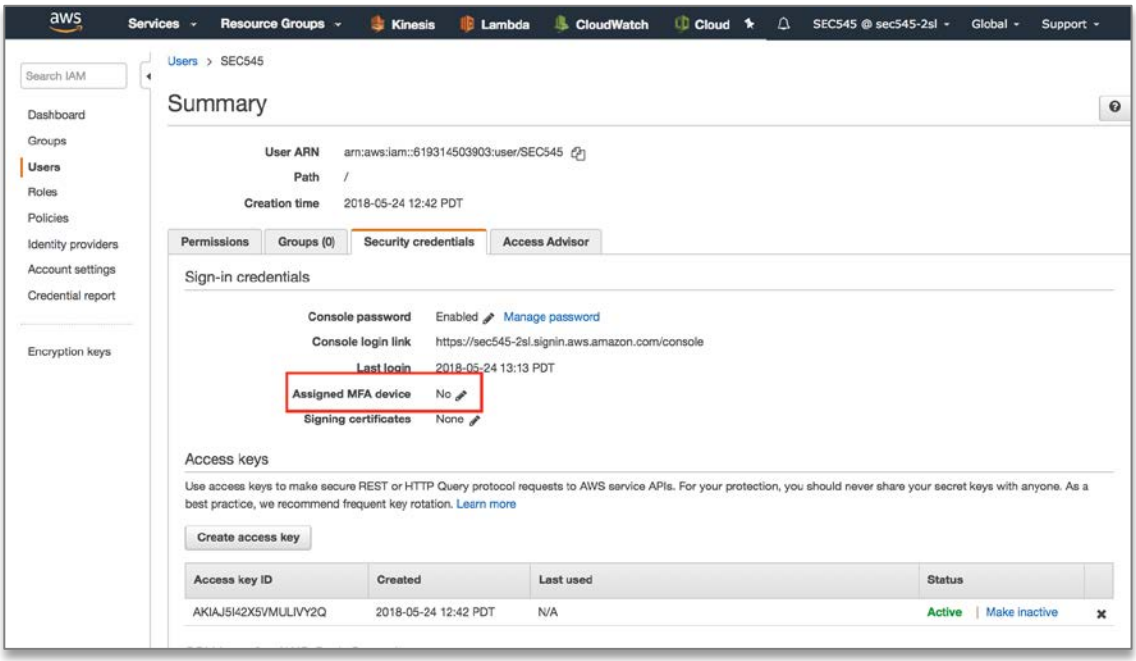


14. Click on the user name to edit the user. Note that MFA is not enabled.



15. Click on the user. Click the Security Credentials tab. Click the pencil next to Assign MFA device.

Follow the instructions to set up MFA. The process to add an MFA device to this user will be the same as adding MFA to the root account as we did in Appendix B.



Appendix D: AWS CLI Setup

Set up the AWS CLI

In order to set up the AWS Command Line Interface (CLI) tool, you will need the **Access Key ID** and **Secret Key** you created in Appendix C. You can install and run the AWS CLI from your own computer, or you can use the SEC545-Ubuntu virtual machine, which has the AWS CLI installed on it.

Either way you'll need to configure the AWS CLI as explained here—use your Access Key and Secret Key from Appendix C and choose the region you are using in the AWS console, as explained in the following documentation:

https://docs.aws.amazon.com/cli/latest/userguide/cli-chap-getting-started.html

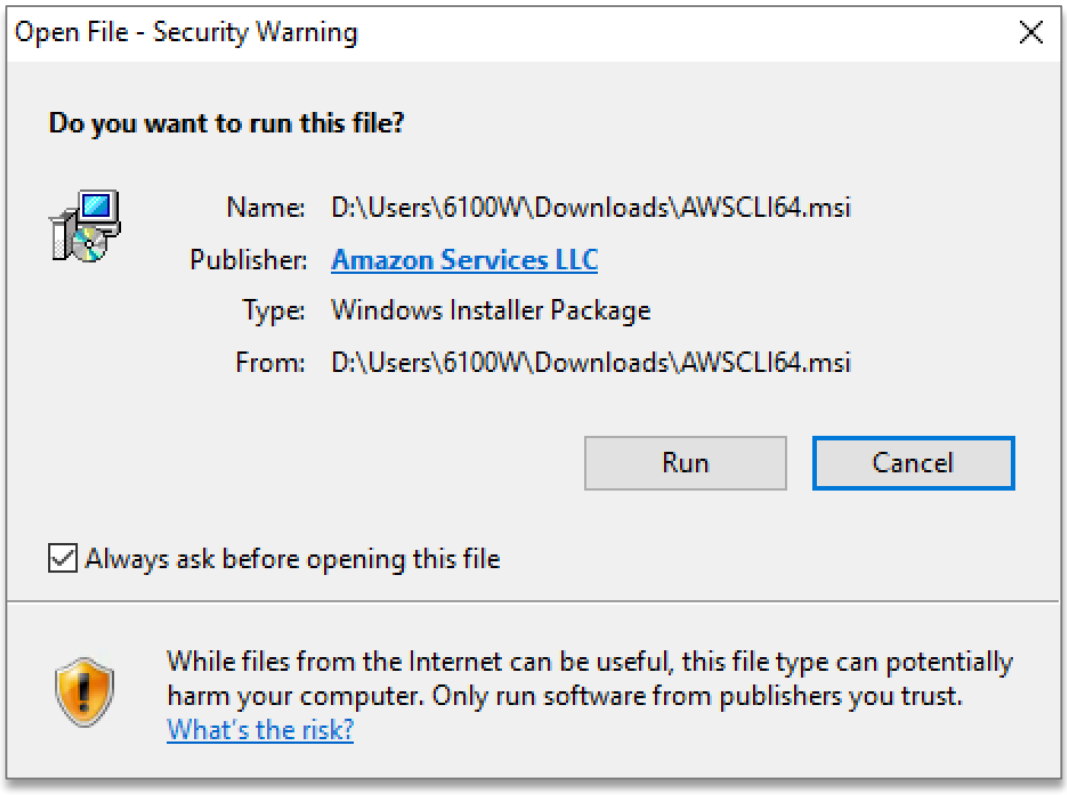
If you choose to use the SEC545-Ubuntu virtual machine, you'll need to wait until you are in class to get the VM. Please refer to Appendix G to ensure your system is set up to run virtual machines. Once you get to class, finish configuring the virtual machine and then follow the CLI configuration instructions.

If you want to install the AWS CLI on your laptop, follow these instructions from the AWS website:

https://docs.aws.amazon.com/cli/latest/userguide/installing.html

Note that although you can use Python to install the AWS CLI, some bundled installers are also provided for different operating systems. Windows users may want to download and install the Windows MSI installer to avoid Python problems.

https://docs.aws.amazon.com/cli/latest/userguide/awscli-install-windows.html#install-msi-on-windows



Check your AWS CLI installation

If you can run the following commands at the command line and they work correctly, you should be good to go. If you get unexpected results, proceed to AWS CLI Troubleshooting.

```
aws --version
aws iam get-account-summary
aws iam list-users --output table
```

It is very important to install the AWS CLI and make sure it is working properly before proceeding to the other labs. Please ask for help right away if needed.

AWS CLI Troubleshooting

If you are having problems installing the AWS CLI, try the following troubleshooting steps as needed.

Use a Virtual Machine

Use the AWS CLI installed in your SEC545-Ubuntu virtual machine, which has the AWS CLI pre-installed, if you are having issues.

Alternatively, you can try downloading a Windows 10 .iso for VMWare or some other virtual machine and installing the AWS CLI on it. Choose this option only if you are familiar with VMs and how to properly configure them, as your instructors won't be able to troubleshoot all variations of VMS that are downloaded and installed. As an example, you can download a copy of the latest Windows 10 ISO here: **https://www.microsoft.com/en-us/software-download/windows10ISO**

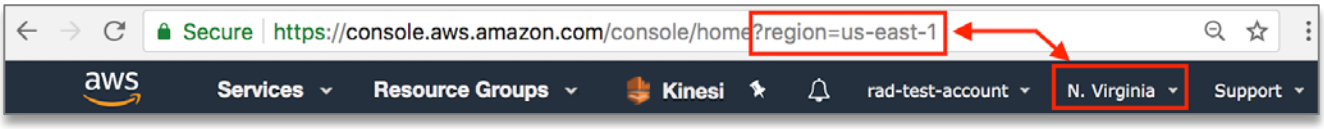
Do not install the AWS CLI, Python, or PIP with sudo or su

Do not install the AWS CLI using sudo if using Mac and Windows. The AWS CLI configuration expects the user that installed the CLI to be the one that is running the commands and puts configuration files into the home directory of the user that installed and configured the CLI.

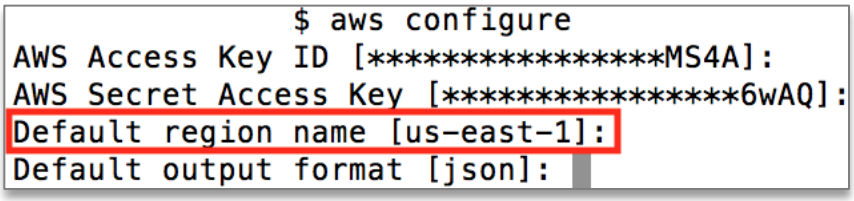
Check Your Region

One of the most common mistakes made when people start using AWS is to configure the AWS CLI in one AWS region and create resources in that region, then open the console and not be able to find any of the resources they created.

In the example below, the console is currently showing resources in the AWS North Virginia, US, region, which has an id of **us-east-1.**



When you run the aws configure command, make sure you enter the region ID from the console when it asks for "Default region name," which from the example above is us-east-1. The Default output format will by default be [none]. The default is "json." You can also set this to "text" or "table." For more information, see: https://docs.aws.amazon.com/cli/latest/userguide/controlling-output.html



New regions are added frequently. The following page contains a complete list of AWS regions:

https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

Issues When Running Old Versions of Python

Running the **pip** command in the AWS documentation with an old version of Python might produce this:

```
2SL:~ tradichel$ pip install awscli --upgrade --user
Cannot fetch index base URL https://pypi.python.org/simple/
Could not find any downloads that satisfy the requirement awscli in /Library/Fra
meworks/Python.framework/Versions/2.7/lib/python2.7/site-packages
Downloading/unpacking awscli
Cleaning up...
No distributions at all found for awscli in /Library/Frameworks/Python.framework
/Versions/2.7/lib/python2.7/site-packages
Storing debug log for failure in /Users/tradichel/.pip/pip.log
```

A **curl** command confirms that the URL in the error message is no longer valid:

```
2SL:~ tradichel$ curl https://pypi.python.org/simple/
<html><head><title>301 Moved Permanently</title></head><body><center><h1>301 Mov
ed Permanently</h1></center></body></html>2SL:~ tradichel$
```

At this point, you'll want to upgrade Python because pip isn't working properly.

Download the latest version of Python from the python web site and install:

https://www.python.org/downloads/

That page should detect your operating system and give you a button to click to download and install the latest version of Python.

Once you install the upgraded version of python, use the commands python3 and pip3 instead of python and pip.

For example, to install the AWS CLI, use **pip3**:

```
pip3 install awscli --upgrade --user
```

To check the Python3 version, use **python3:**

```
python3 --version
```

Pip says AWS is already installed and won't update the version.

Force pip to not use the cache directory when installing the new version.

```
pip3 install --no-cache-dir awscli --upgrade --user
```

If you get “command not found” when you try to execute pip or python, your path is not set correctly so your operating system can find Python. See the instructions below for setting the python path if you want to try to figure out how to set the patch correctly.

However, if you are having a lot of problems with Python or can’t get it working, try the alternate installations methods at the bottom of the AWS CLI instructions:

https://docs.aws.amazon.com/cli/latest/userguide/installing.html

Multiple Versions of Python

To see if you are running an older version of Python, type **python --version**

```
Python 2.7.10
2SL:~ tradichel$
```

To see if you are running a version of Python 3, type **python3 --version**

```
2SL:~ tradichel$ python3 --version
Python 3.5.1
```

To see where python is installed on a Mac, type **which python, which python3,** or a specific version such as **which python3.6**

```
2SL:~ tradichel$ which python
/Library/Frameworks/Python.framework/Versions/2.7/bin/python
```

To find Python on Windows, search for python from the Windows menu. Right-click the file name in the results, select **Properties,** and find **Location**. This will be the path to the python executable.

As stated in the Python README, you can run a specific version of python by adding the version to the command:

Python 3 and Python 2 Co-existence

Python.org Python 3.6 and 2.7.x versions can both be installed on your system and will not conflict. Command names for Python 3 contain a 3 in them, `python3` (or `python3.6`), `idle3` (or `idle3.6`), `pip3` (or `pip3.6`), etc. Python 2.7 command names contain a 2 or no digit: `python2` (or `python2.7` or `python`), `idle2` (or `idle2.7` or `idle`), etc.

The problem is that the AWS CLI may get installed in a python folder that is not in the path, in which case you will get an error saying command not found.

Alternately, an older version of the AWS CLI and Python may execute when you type the AWS command that is not the version you want or expect. On this system, Python 3.6 and a newer version of the CLI is installed, but the system is picking up the older versions.

```
2SL:site-packages tradichel$ aws --version
aws-cli/1.15.0 Python/2.7.10 Darwin/17.5.0 botocore/1.10.0
```

In either case, find where the AWS CLI was installed by searching for it on the file system using your favorite command for finding files.

sudo find / -name awscli

This system happens to have four different versions of the AWS CLI installed, and the system is using the one installed in the first directory below.

```
/Library/Frameworks/Python.framework/Versions/2.7/bin/aws
```

And …

```
/Users/tradichel/Library/Python/3.5/bin/aws
/Users/tradichel/Library/Python/2.7/bin/aws
/Users/tradichel/Library/Python/3.6/bin/aws
```

Look at the path on a Mac by typing the $PATH command or look at the path in the Windows environment variables by typing the set command at the command prompt and looking for the "PATH" variable and setting.

```
2SL:~ tradichel$ $PATH
-bash: /Library/Frameworks/Python.framework/Versions/2.7/bin:/Library/Frameworks/Python.framework/Versi
ons/3.5/bin:/Library/Frameworks/Python.framework/Versions/3.5/bin:/Library/Frameworks/Python.framework/
Versions/3.5/bin:/Library/Frameworks/Python.framework/Versions/3.5/bin:/usr/local/bin:/usr/bin:/bin:/us
r/sbin:/sbin:/usr/local/go/bin:/usr/local/MacGPG2/bin:/Applications/Wireshark.app/Contents/MacOS: No su
ch file or directory
```

As shown above, the first line of the path is pointing to the old version of the AWS CLI (2.7). Additionally, the rest of the path is pointing to Python 3.5 (multiple times unnecessarily). Change the path as described in the AWS CLI documentation to point to the version of the CLI and Python you want to use.

For this particular system, the solution is to edit the Mac ~/.bash_profile, remove the multiple versions of 3.5 and 2.7, save, close the terminal window, and re-open a new one, then try the $PATH command again. Updating the path for your system may be different, as explained in the AWS CLI documentation.

```
2SL:~ tradichel$ $PATH
-bash: /Library/Frameworks/Python.framework/Versions/3.6/bin:/Library/Frameworks
/Python.framework/Versions/3.5/bin:~/Library/Python/3.6/bin:/usr/local/bin:/usr/
bin:/bin:/usr/sbin:/sbin:/usr/local/go/bin:/usr/local/MacGPG2/bin:/Applications/
```

Now running the aws version command comes up with the desired versions.

```
2SL:~ tradichel$ aws --version
aws-cli/1.15.28 Python/3.6.5 Darwin/17.5.0 botocore/1.10.28
```

Beware of Proxies Blocking AWS Endpoints

If you are in an environment running a proxy, the proxy will need to allow access to the AWS endpoint URLs.

Firewalls Need to Allow Access

The firewall on your local machine and any firewalls between your system and AWS need to be open to send traffic to AWS on any required ports. Most of the AWS API calls made by the AWS API in this class require port 443. Some AWS services may require other ports.

Sync Your Time

If your system clock is out of sync, authorization may fail. Make sure NTP on your system is correctly configured and your system clock shows the accurate time.

Appendix E: SSH on AWS

Overview

SSH on AWS is very similar to using SSH with any other system. If you are not familiar with the steps to access a remote machine via SSH, please read through the following instructions and install the required software if using an older version of Windows. We will test logging into a remote host in Lab 1.1.

Depending on which version of which operating system you are running, you will need to log in differently. Regardless of how you log in, you will need an SSH Key (EC2 Key Pair) so we'll create it now. It will be used in subsequent labs. When logging in via SSH to VMs on your local machine, you'll just be using a user name and password, not a key file.

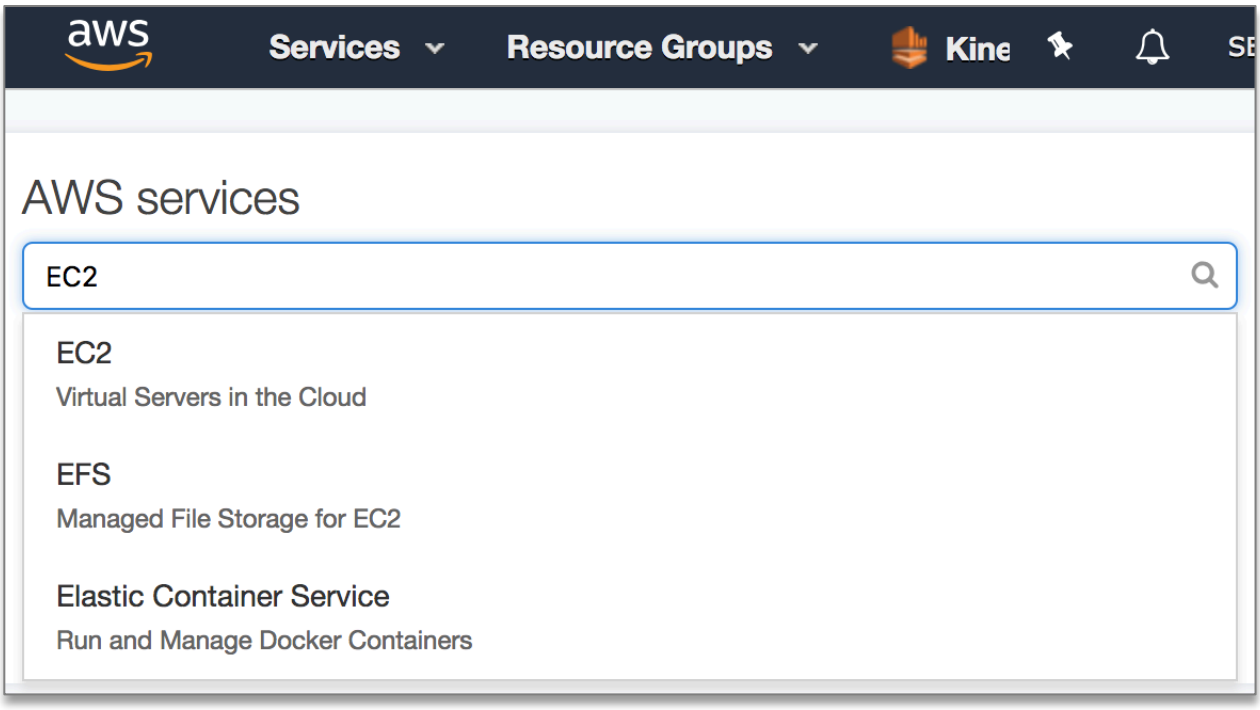
It is very important to make sure you understand how to log in to a remote host using SSH in order to complete subsequent labs in this book. Please ask for help right away if needed.

Create an SSH Key

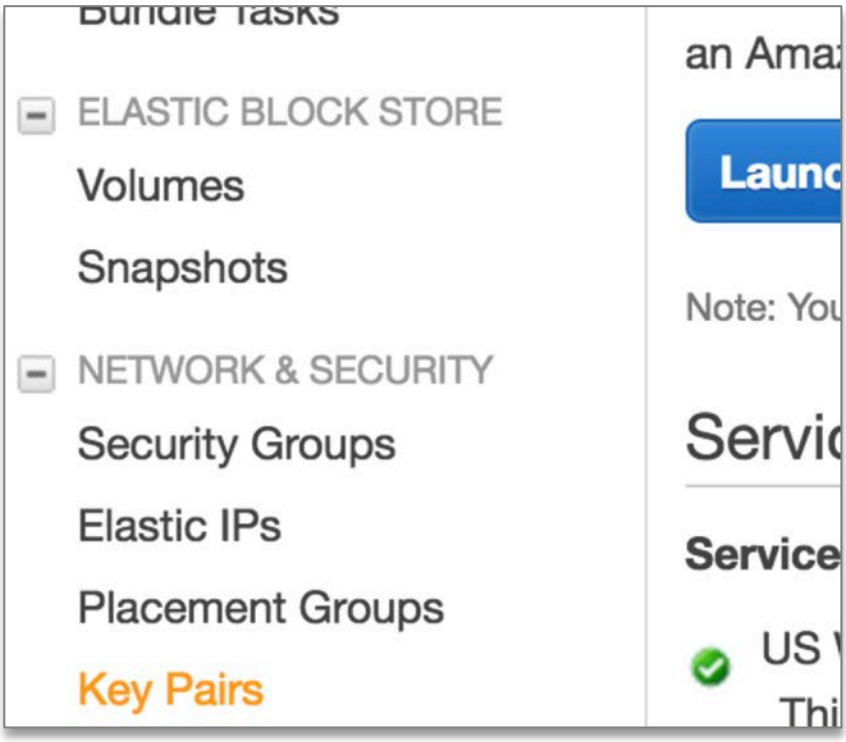
AWS allows you to deploy Linux hosts (called EC2 Instances) in your account. You can then access these instances remotely with SSH using an SSH key (called an EC2 Keypair). Each time you create a new Linux EC2 instance, you are given the option to choose an existing keypair or create new one. Let's create a keypair that we can use throughout the rest of the class. Make sure you save this keypair somewhere you'll remember for future labs.

1. **Log in using the SEC545 user and URL from Appendix C.**

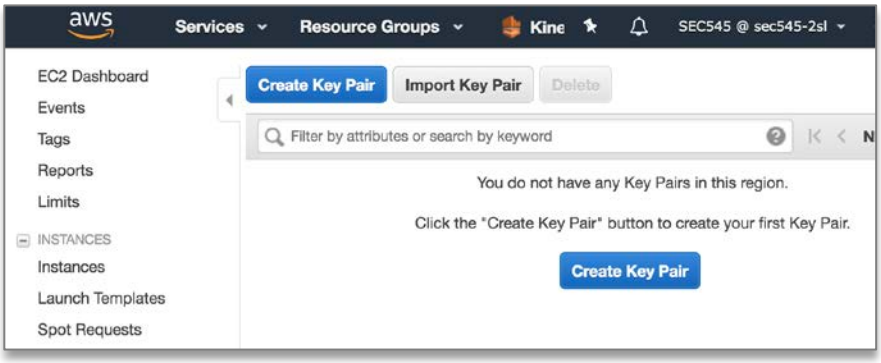
2. Navigate to the EC2 service. Click the AWS Icon, then type EC2 in the search box.



3. Scroll down to Key Pairs in the left menu.

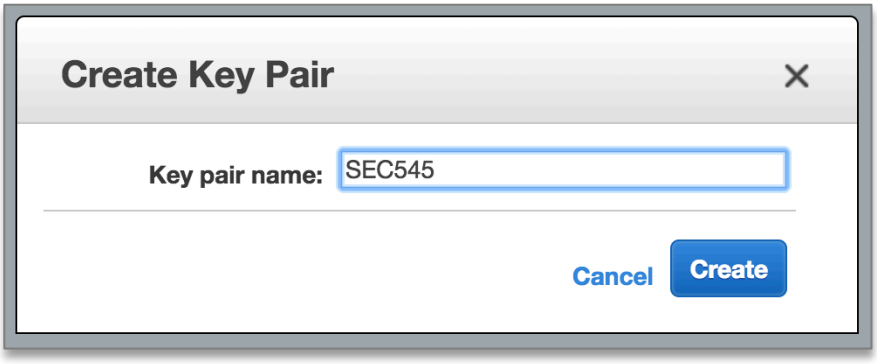


4. Click "Create Key Pair."



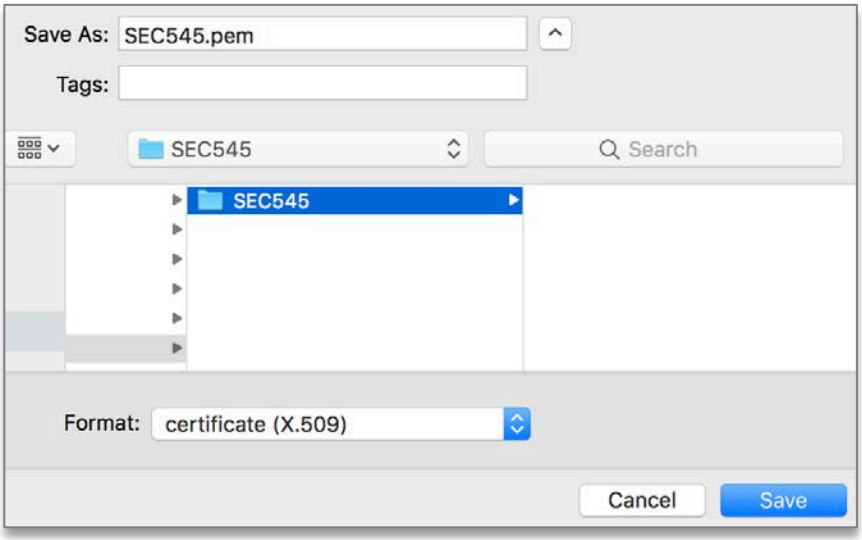
5. Name it SEC545 and click Create.

This is the name of the key pair you will choose when you create a Linux instance on AWS (we'll do that in Lab 1.1).



6. Save the file in a place and with a name you'll remember!

Once you click "Create," the key file will be downloaded however your browser is configured to download files. The file name will match the name of the key with ".pem" as the extension—in this case, SEC545.pem. This file will be used to connect to remote hosts in the cloud. This is the only point at which you will be able to download this key, so keep it somewhere you can remember.



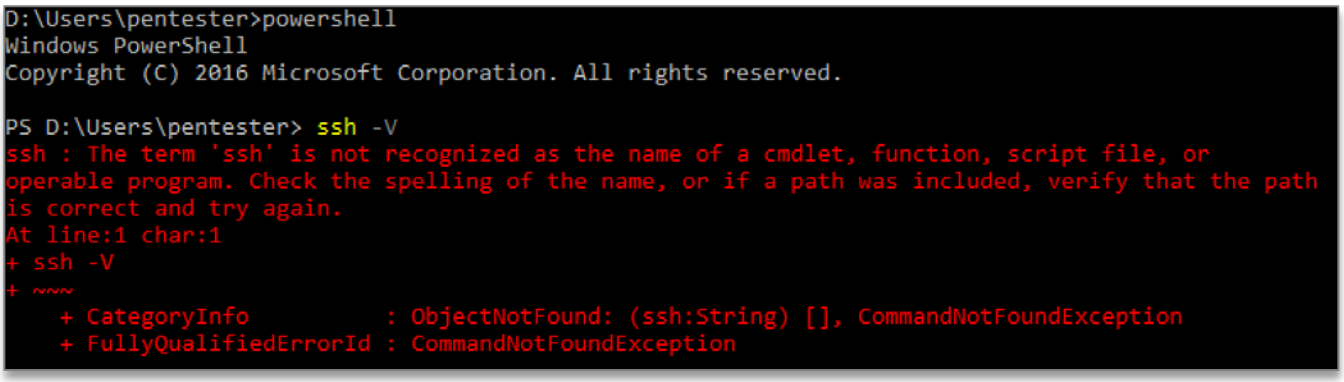
SSH on Windows using the Default SSH Client

Starting with the April 2018 update for Windows 10, an SSH client is installed by default. If you have that version of Windows or later, here are the steps to use the SSH client built into Windows.

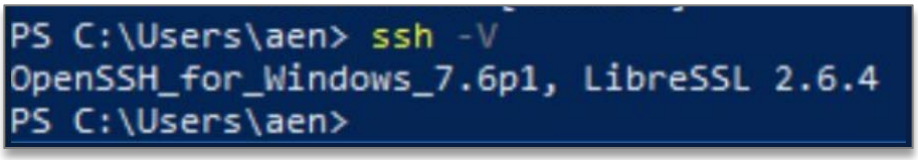
To see if you have the built-in SSH client, first start Powershell, then run ssh -V to get your ssh version.

```
powershell
ssh -V
```

If you get an error that ssh is not a recognized command, update Windows or install PuTTY as explained in the next section.



If you have SSH on your system, then proceed to the Linux/Mac instructions that explain how to use SSH from the command line.

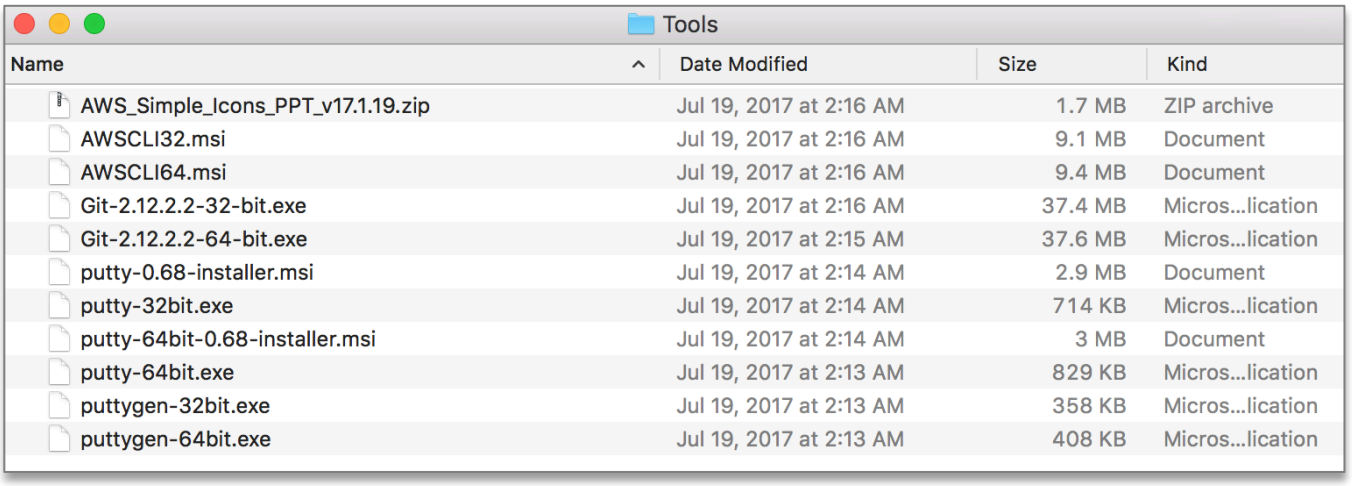


SSH on Windows using PuTTY and PuTTYgen

Install PuTTY and PuTTYgen

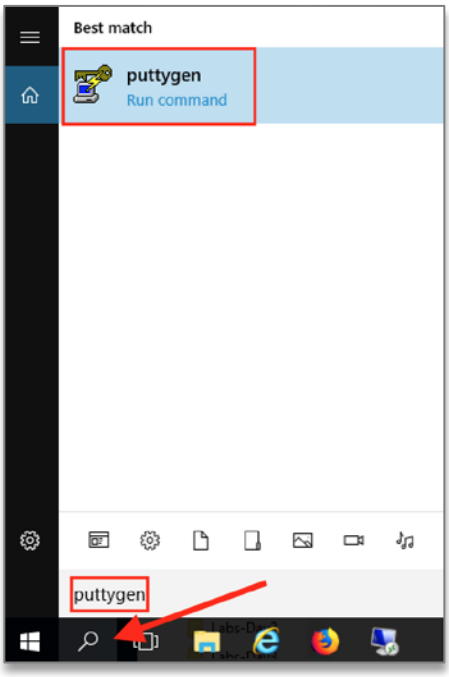
If you are using a version of Windows that does not have an SSH client by default, you will need to install PuTTY and PuTTYgen to SSH to Amazon EC2 instances (virtual hosts). PuTTY is a client that allows Windows hosts to log in to machines that support SSH. PuTTY requires SSH keys to be in the .ppk format, so you will have to use PuTTYgen to convert the .pem file from AWS to a .ppk file before you log in.

Install PuTTY and PuTTYgen from the tools directory on the USB. Use the correct version for your operating system.

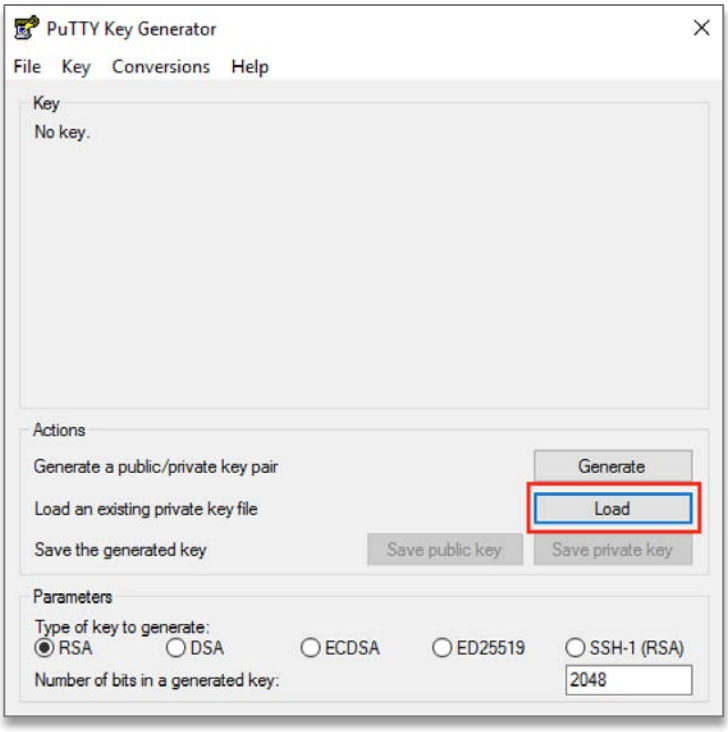


Convert SSH Key to PPK

1. From the Start menu, choose All Programs > PuTTY > PuTTYgen or search for it.

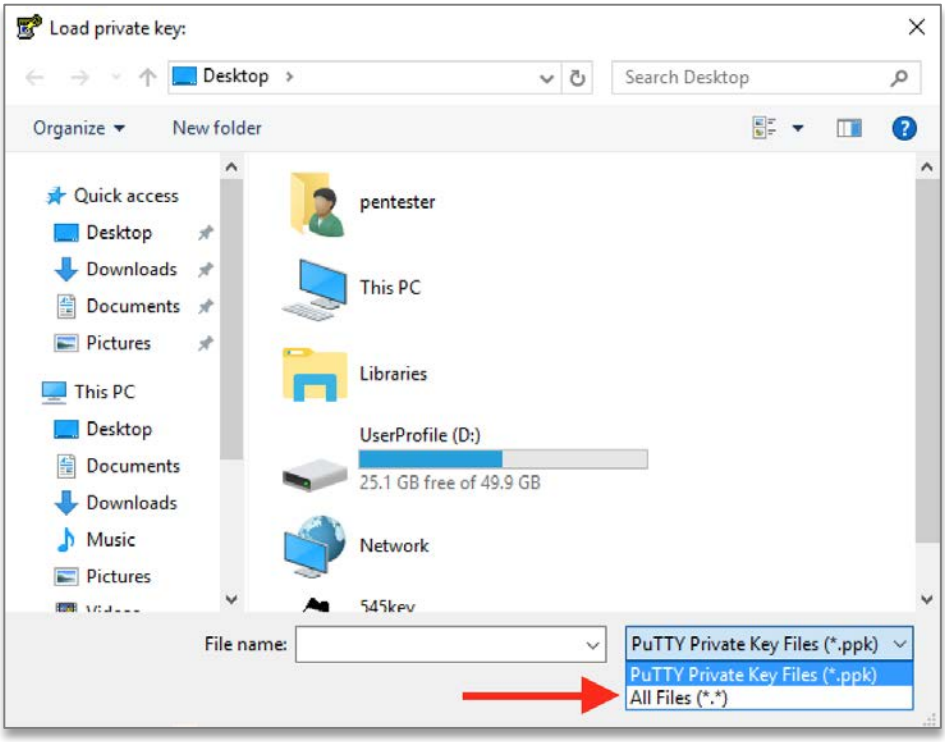


2. Choose Load.

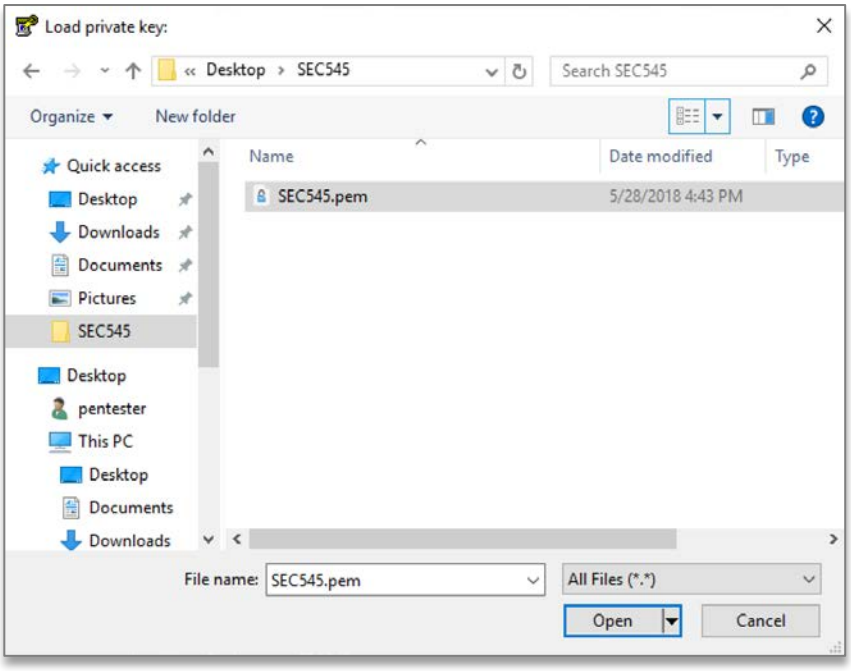


3. Select the option to display files of all types.

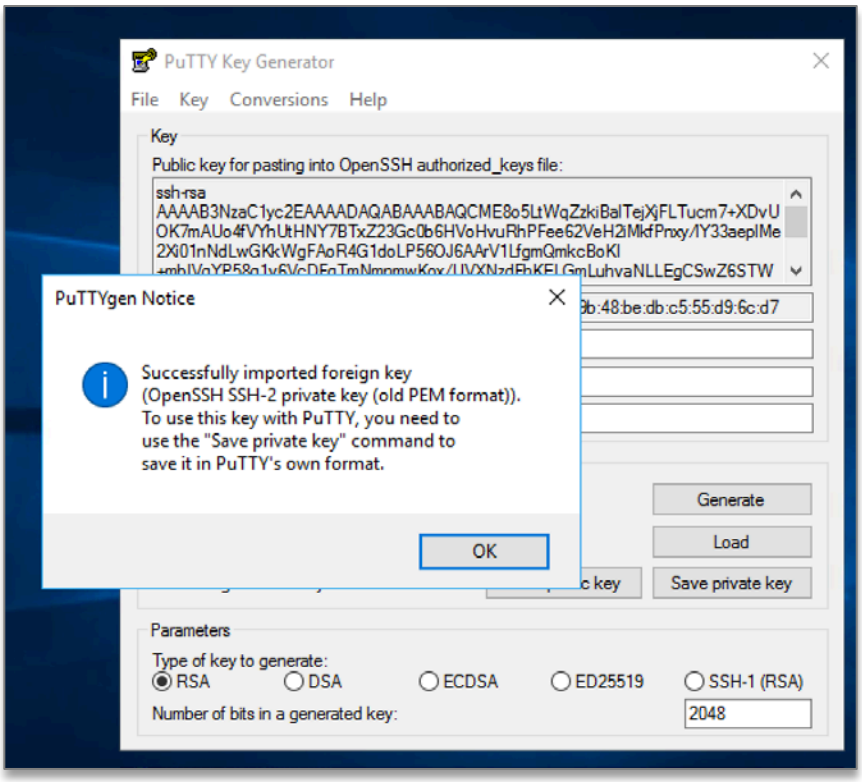
By default, PuTTYgen displays only files with the extension. ppk.



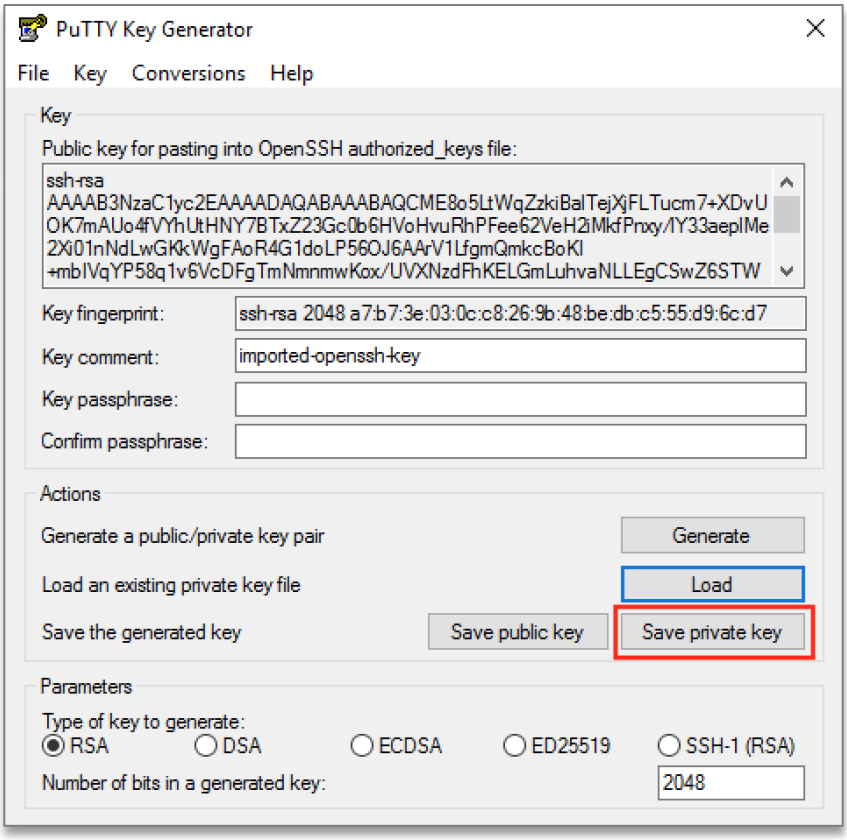
4. Select your SEC545.pem file created in previous steps.



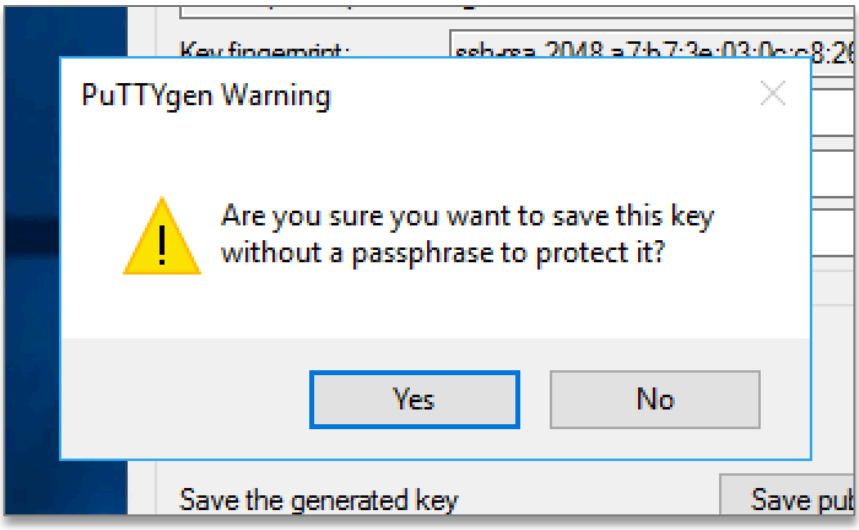
5. Choose OK to dismiss the confirmation dialog box.



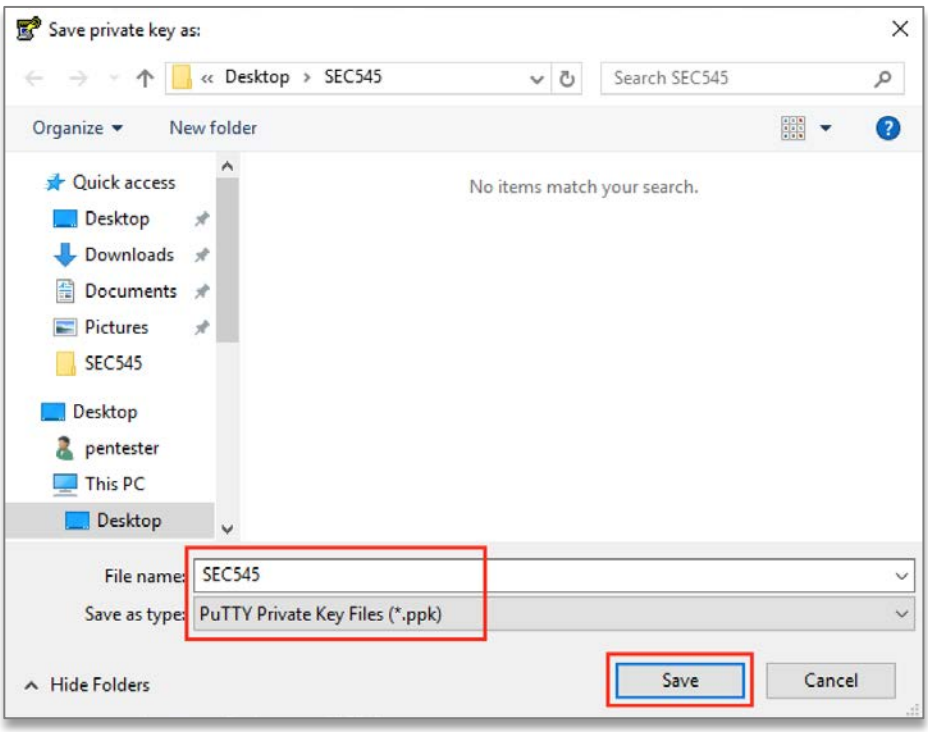
6. Choose Save private key to save the key in the format that PuTTY can use.



7. PuTTYgen displays a warning about saving the key without a passphrase. Choose Yes.



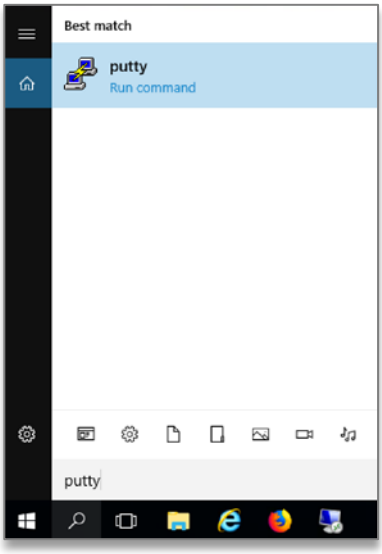
8. Specify the name SEC545 and make sure Save as type is .ppk. Click Save.



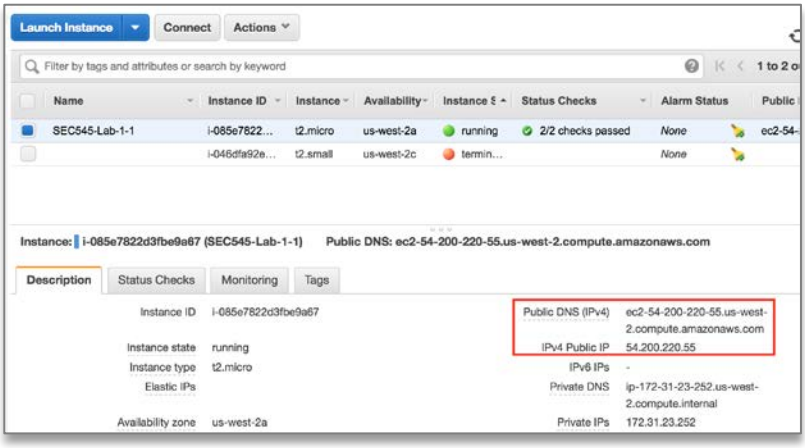
SSH Connection Using PuTTY

We will take these steps in Lab 1.1, but they are provided here for your reference.

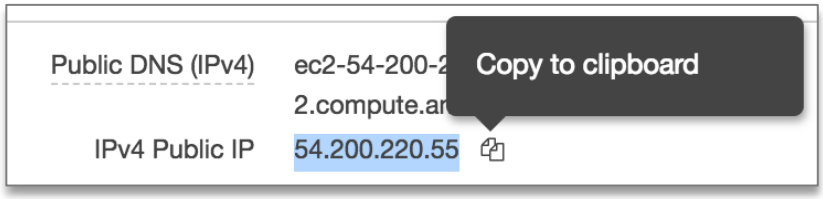
1. Run PuTTY.



2. Obtain your AWS Public (not private) IP address or Domain Name (more in Lab 1.1).

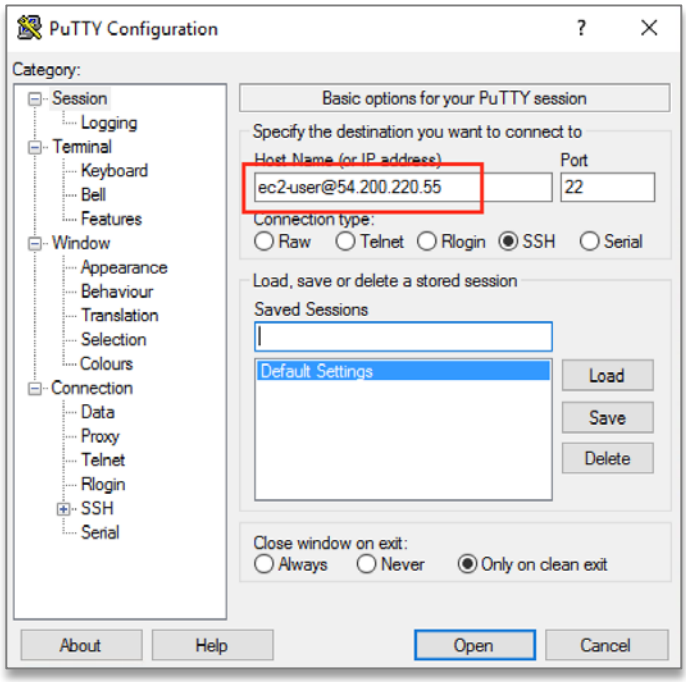


Tip: Hover over the IP address and click the copy icon, then paste the IP address into PuTTY.

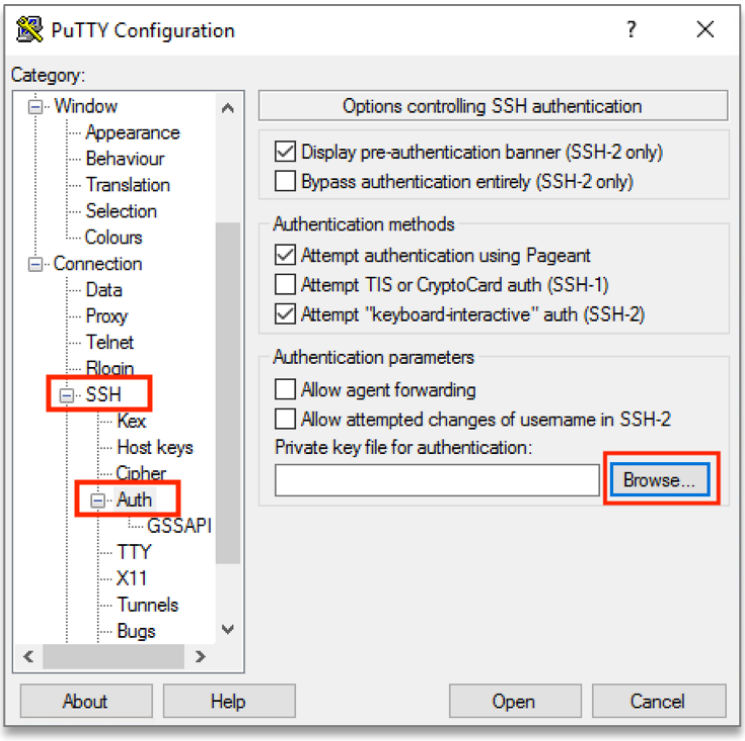


3. Enter the default user name (ec2-user) + @ plus the public IP or DNS name of your host as shown.

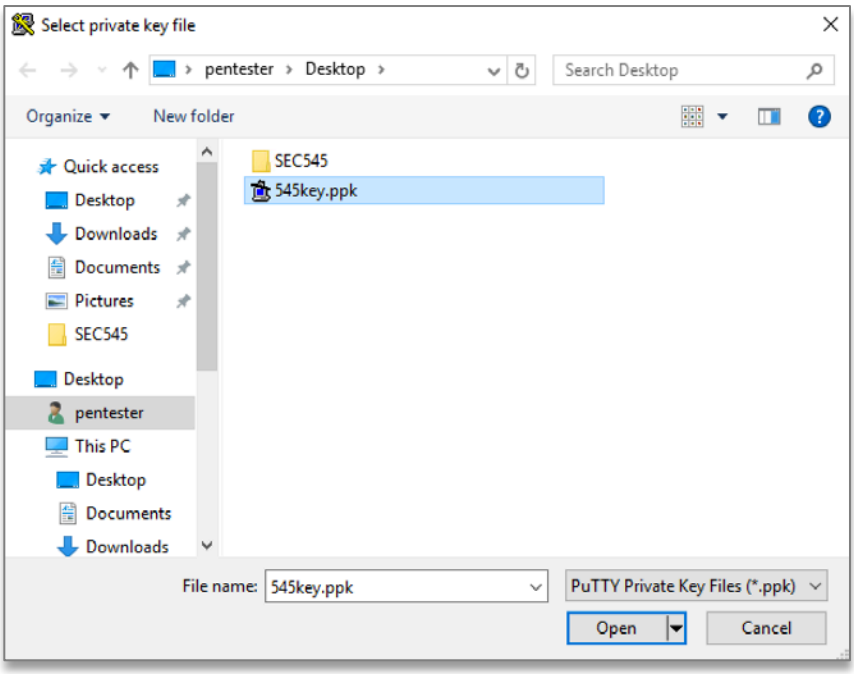
ec2-user@<Your-Public-IP> (Leave port set to 22 and SSH selected.)



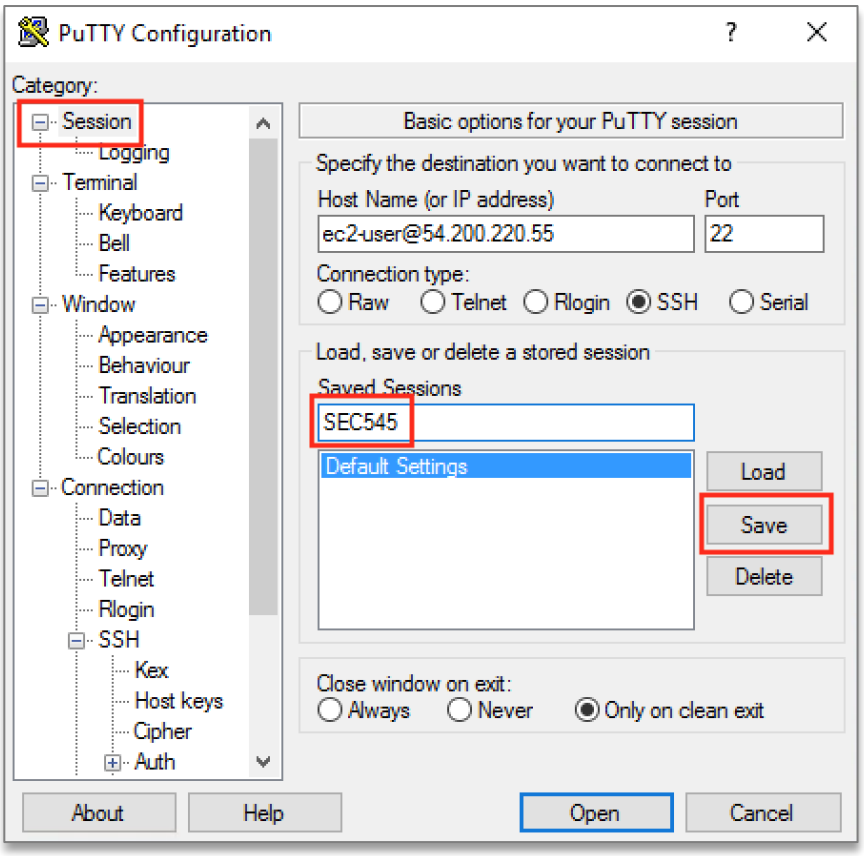
4. Select Connection -> SSH -> Auth. Click Browse…



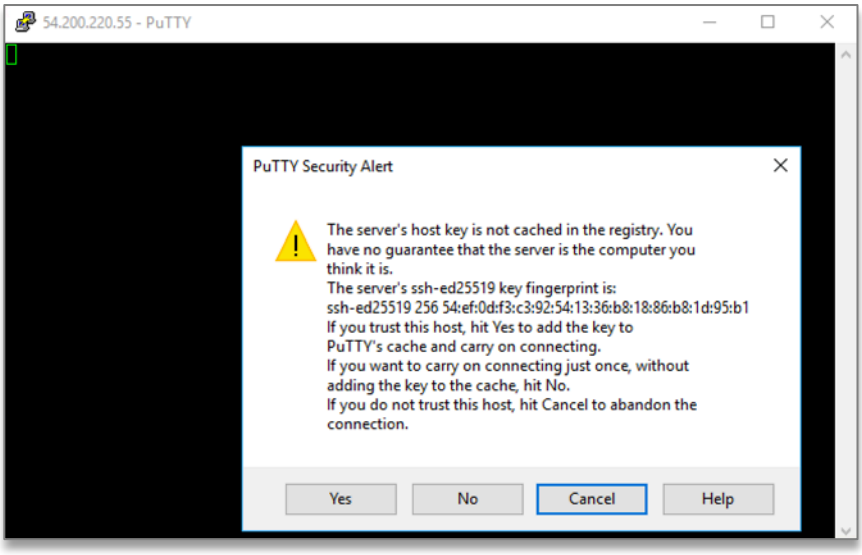
5. Select your ppk file.



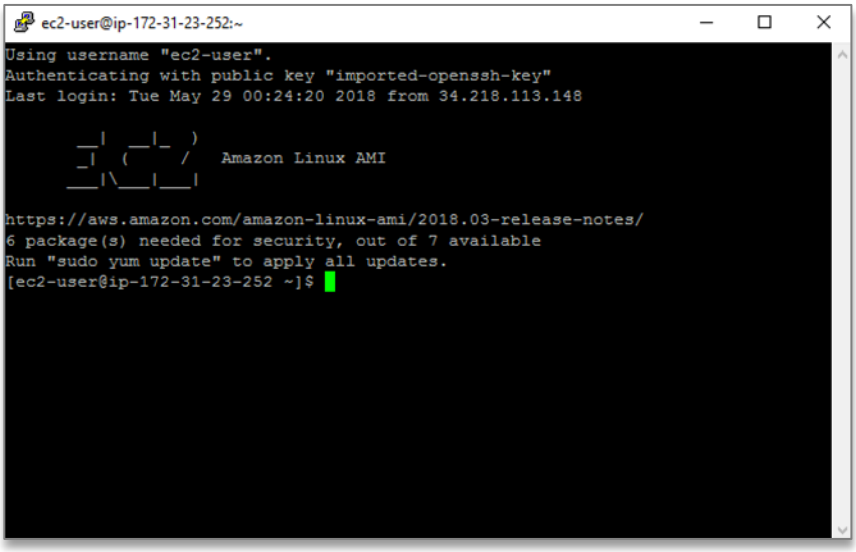
6. Select Session again in the left-hand menu. Enter a session name and save it.



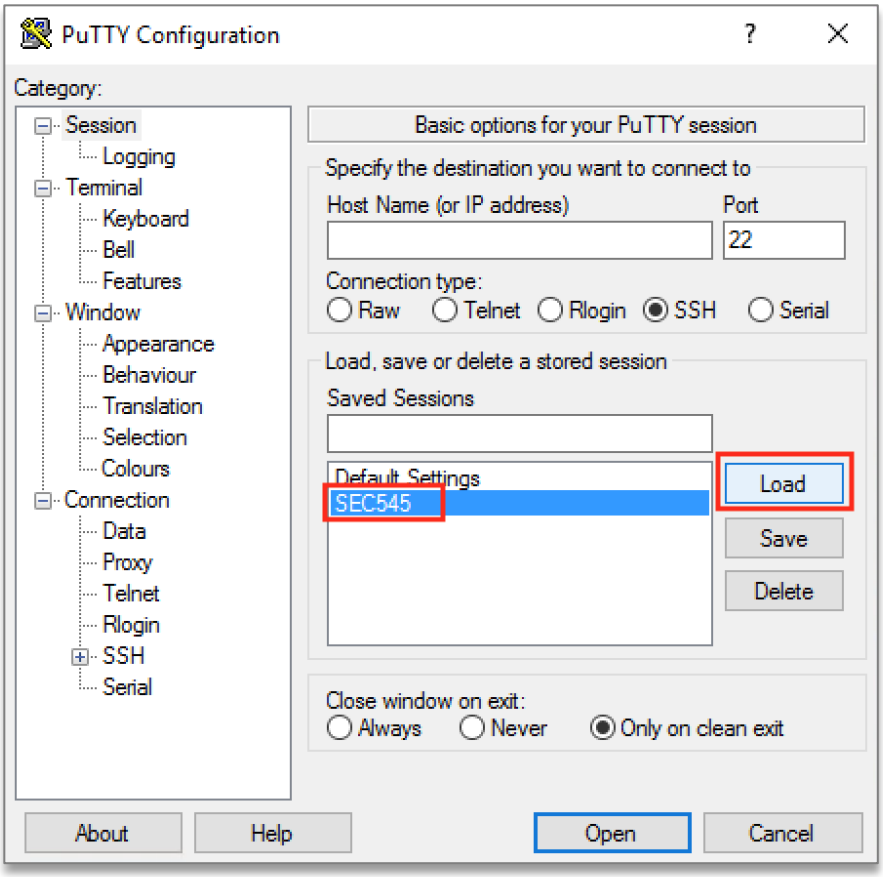
7. Click “Open” to start the session. Click “Yes” to accept the warning.



8. You should see a banner such as the following. Go to troubleshooting steps if you don’t.



Tip: To reuse the same settings after closing this connection, click on the connection name you saved and click “Load” then “Open.”



SSH on Linux / Mac

Step 1: Open a terminal window.

Step 2: Change to the directory where your .pem file is stored.

```
cd <directory>
```

Step 3: Change the permissions of the key, or SSH connections will throw an error.

```
chmod 400 SEC545.pem
```

Step 4: Type the following command to SSH to a remote instance.

```
ssh –i SEC545.pem ec2-user@<instance public IP>
```

If you get a warning, accept it and log in. This only reflects the fact that the key and connection in use are the first pairing of the keys for client and server—they're all ours, so we're ok.

SSH Troubleshooting

If you have problems using SSH during future labs, you can refer back to these troubleshooting tips.

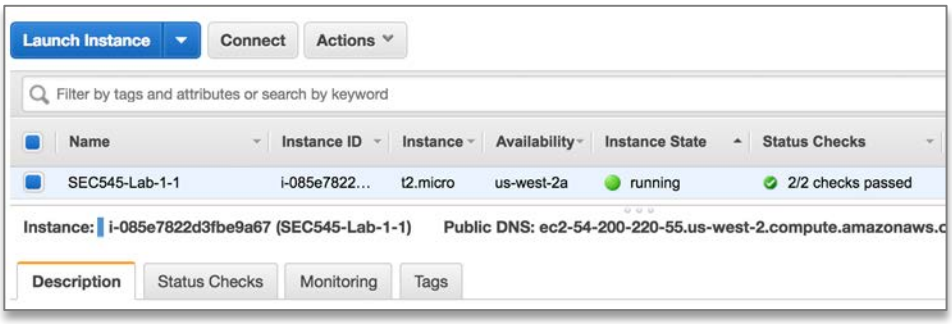
Check the AWS Troubleshooting Documentation

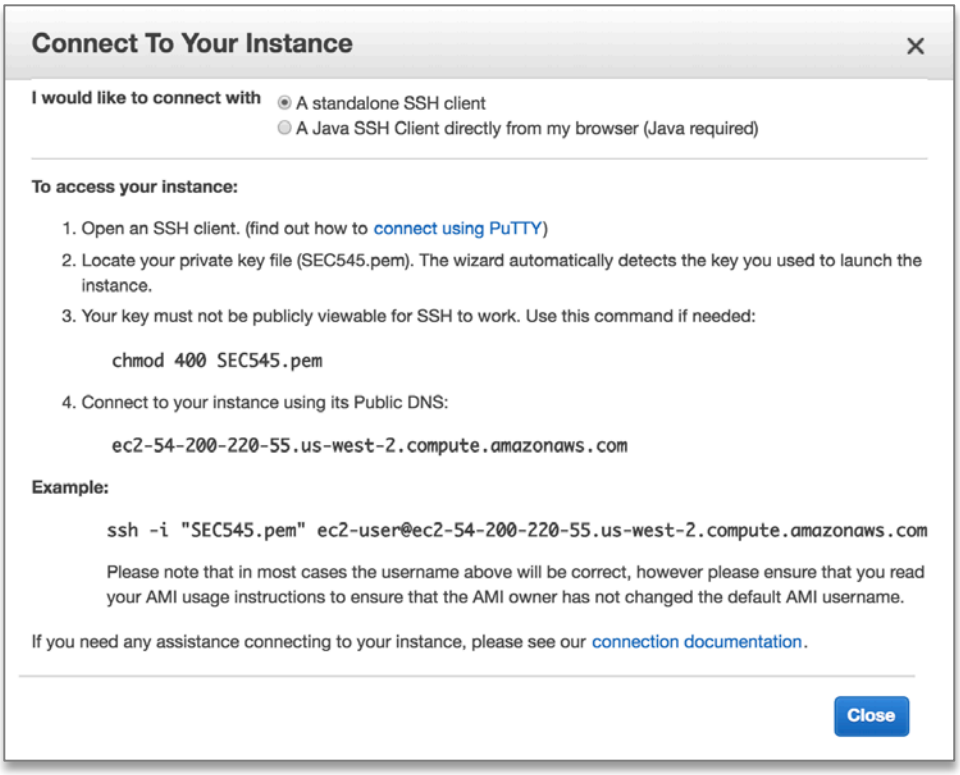
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/TroubleshootingInstancesConnecting.html

IP Address/Domain Name

Make sure you chose the option to assign a public IP address to your instance and are using the correct public IP address when you SSH to that instance. Check the EC2 instance details in the console. Make sure you are using the public IP address or domain name, not the private IP address of the EC2 instance. This is covered in Lab 1.1.

You can quickly get connection information for an EC2 instance by clicking on the instance, then clicking the "Connect" button:





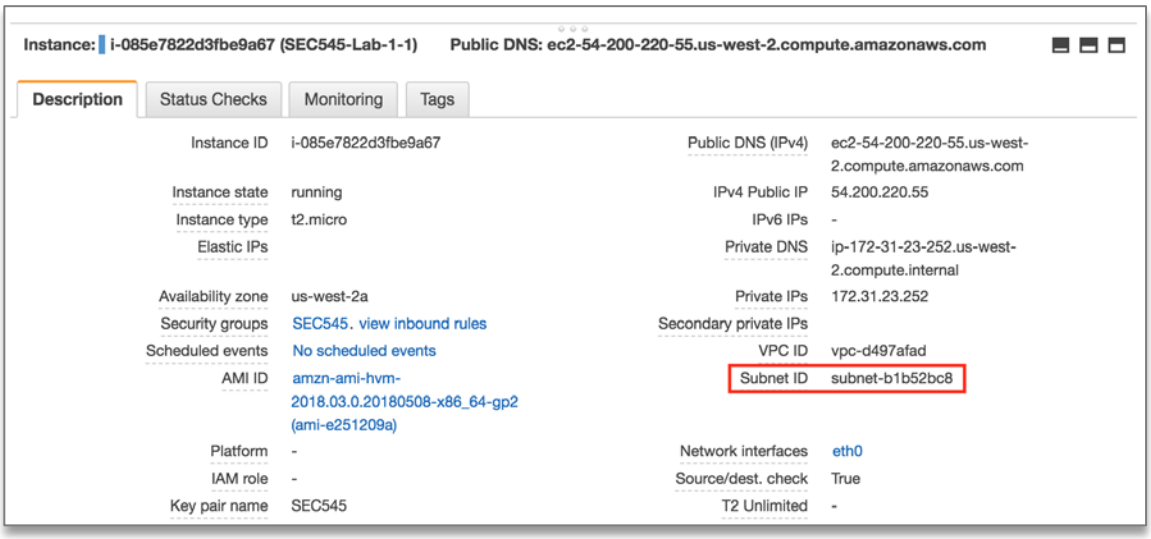
Make sure you are using the right key file!

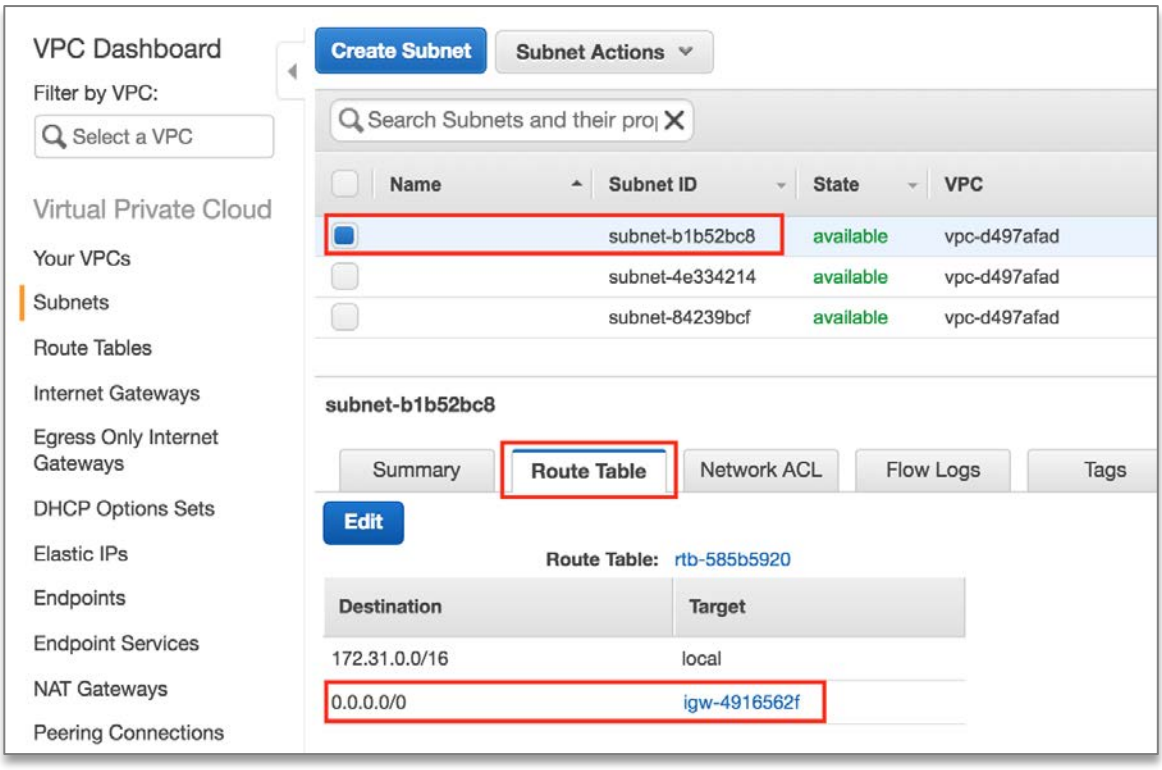
If you followed the instructions above, the key file name is SEC545.pem. Sometimes people create new keys instead of the key we created above. If you followed the instructions, the name of the keypair on your instance in the screenshot above will be SEC545 and the name of the key file on your laptop will be SEC545.PEM. If you converted it using PuTTYgen as described above, you will have a SEC545.PPK file that was used to configure PuTTY.

If you are using Linux or Mac, make sure you set the permissions correctly as specified in the instructions.

Check that your route table has an Internet Gateway route.

If you are using the default VPC in your account, this is not the issue. If you created a new VPC, make sure you have a route in your VPC and/or subnet route tables for an internet gateway. Resources in your account cannot access the Internet without an internet gateway route. This is covered in lab 1.1.


Instance: i-085e7822d3fbe9a67 (SEC545-Lab-1-1) Public DNS: ec2-54-200-220-55.us-west-2.compute.amazonaws.com
Description
Status Checks
Monitoring
Tags
Instance ID i-085e7822d3fbe9a67
Instance state running
Instance type t2.micro
Elastic IPs
Availability zone us-west-2a
Security groups SEC545. view inbound rules
Scheduled events No scheduled events
AMI ID amzn-ami-hvm-2018.03.0.20180508-x86_64-gp2 (ami-e251209a)
Platform -
IAM role -
Key pair name SEC545
Public DNS (IPv4) ec2-54-200-220-55.us-west-2.compute.amazonaws.com
IPv4 Public IP 54.200.220.55
IPv6 IPs -
Private DNS ip-172-31-23-252.us-west-2.compute.internal
Private IPs 172.31.23.252
Secondary private IPs
VPC ID vpc-d497afad
Subnet ID subnet-b1b52bc8
Network interfaces eth0
Source/dest. check True
T2 Unlimited -

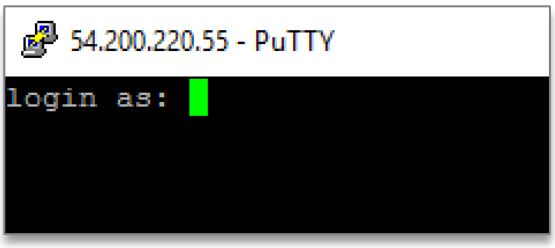

VPC Dashboard
Filter by VPC:
Select a VPC
Virtual Private Cloud
Your VPCs
Subnets
Route Tables
Internet Gateways
Egress Only Internet Gateways
DHCP Options Sets
Elastic IPs
Endpoints
Endpoint Services
NAT Gateways
Peering Connections
Create Subnet
Subnet Actions
Search Subnets and their pro
Name
Subnet ID
State
VPC
subnet-b1b52bc8 available vpc-d497afad
subnet-4e334214 available vpc-d497afad
subnet-84239bcf available vpc-d497afad
subnet-b1b52bc8
Summary
Route Table
Network ACL
Flow Logs
Tags
Edit
Route Table: rtb-585b5920
Destination
Target
172.31.0.0/16 local
0.0.0.0/0 igw-4916562f

PuTTY

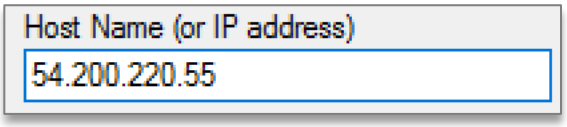
Make sure you are using the .ppk file created in the steps above, not the .pem file.

If you are using an existing PuTTY installation, check that all your settings match the screenshot.

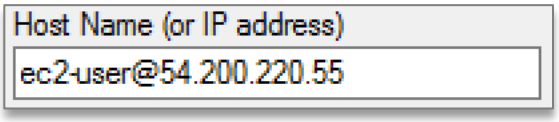
If you see this request for a user name, you didn't put the default user in the host box in PuTTY.



This:



Should be:



Linux/Mac

If you are using Linux or Mac, make sure you have typed the command correctly. The default user on an EC2 instance is ec2-user. Look carefully at the example command. If you are prompted for a user name, you likely didn't enter the user name in the command, the correct IP address, or the correct key file following the -i (lowercase I, not L) in the command.

Ensure your AWS networking rules allow proper access.

Make sure your instance is in a Security Group that has the correct ports open inbound (ingress) and outbound (egress).

In this class, we do not create NACLs (Network Access Control Lists) on subnets, but if you did or you used an existing account, then you'd have to check the subnet Network Access Control List (NACL) to make sure it had the correct network rules as well. NACLs are covered briefly in Lab 1.1 and on Day 2 in this class. If you are using a new account, don't know what NACLs are, and didn't change them, then this is not the problem.

Turn on VPC Flow Logs to see if any traffic is getting rejected on AWS. VPC Flow Logs are covered in Lab 4.5.

Firewalls

Your host firewall and any firewalls between you and the host you are trying to SSH into must allow remote access to port 22 on the internet and return traffic on ephemeral ports.

```
ssh: connect to host 54.200.220.55 port 22: Connection refused
```

If you have to wait awhile after you try to connect followed by this error, then a firewall somewhere between you and your instance is blocking the response to your SSH request on ephemeral ports:

```
ssh: connect to host 54.200.220.55 port 22: Operation timed out
```

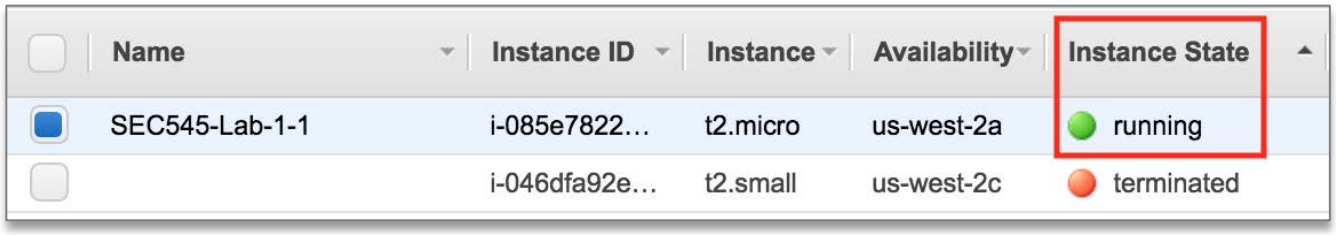
When the connection leaves your local machine, it sends the request on port 22. The return traffic comes back to your machine on ephemeral ports. If you don't know what ephemeral ports are, AWS does a pretty good job of explaining them here:

https://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_ACLs.html#VPC_ACLs_Ephemeral_Ports

If you are familiar with Wireshark or tcpdump, you can try using them on your local machine to see what's blocked or check any firewall logs to find where traffic might be rejected.

Check your EC2 instance to make sure it is in the running state.

Make sure your EC2 instance is fully in the running state before you try to connect.



Name	Instance ID	Instance	Availability	Instance State
SEC545-Lab-1-1	i-085e7822…	t2.micro	us-west-2a	running
	i-046dfa92e…	t2.small	us-west-2c	terminated

VPN and Security Software

If you are connecting over a VPN, try connecting without using the VPN.

Try turning off any security software temporarily to see if it is causing the problem.

Appendix F: Setup Git

Install Git if it is not already installed on your system. This will be used in a CTF (Capture the Flag) lab later in the class.

If you are running Linux or Mac OS X, you may have Git installed already. If you are running Windows, we have included Git installers for Windows in the "Tools" directory on your USB (both 32-bit and 64-bit). Accept all the defaults, and you will likely need to restart your command prompt. If that doesn't work, reboot.

To see if Git is installed or verify it is installed correctly, run the following command:

```
git --version
```

You can find additional, detailed instructions about how to install Git on different systems here:

https://git-scm.com/book/en/v2/Getting-Started-Installing-Git

Appendix G: VM Networking

A number of virtual machines (VMs) will be used to introduce various concepts in this class. The VMs will probably not work well, or at all, if you run them off the USB directly. We're going to copy the VMs off the USB and make sure they are working and are configured with the correct IP addresses.

The details for the VMs are as follows—we will check and set the IPs if needed in subsequent steps.

File: /VMs/ORCH01/ORCH01.vmx

User: student

Password: Passw0rd (with a zero)

IP: 10.10.10.11

Use su – to change to root user with same password

File: /VMs/SEC545-CentOS7/SEC545-CentOS7.vmx

User: student

Password: Passw0rd (with a zero)

IP: 10.10.10.10

Use su – to change to root user with same password

File: /VMs/SEC545-Ubuntu/SEC545-Ubuntu.vmx

User: sec545

Password: Passw0rd (with a zero)

IP: 10.10.10.9

Use su – to change to root user with same password

File: /VMs/XenServer/XenServer.vmx

User: root

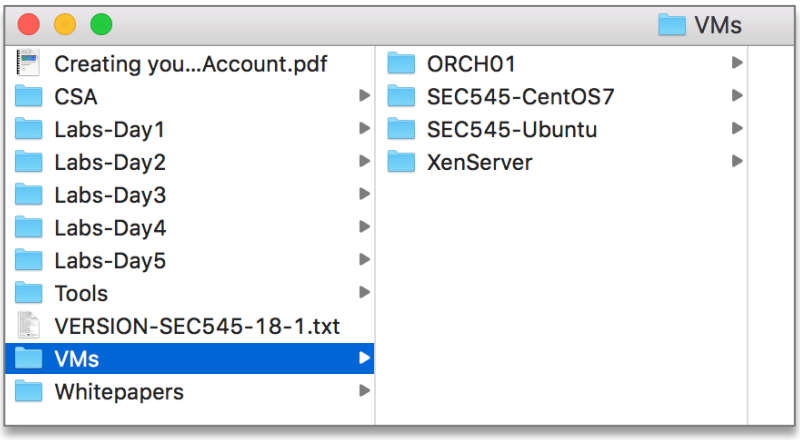
Password: Passw0rd (with a zero)

IP: 10.10.10.30

Use that information to configure the VMs, as explained in the steps on the subsequent pages.

Steps

1. **Copy the VMs folder on the USB to the hard drive of the computer you will use for the labs.**



2. **Install VMWare Player if you don't have it already.**

You can run virtual machines with the VMware software of your choosing if you already have something installed for this purpose. If not, please download and install VMWare Player for Linux or Windows:

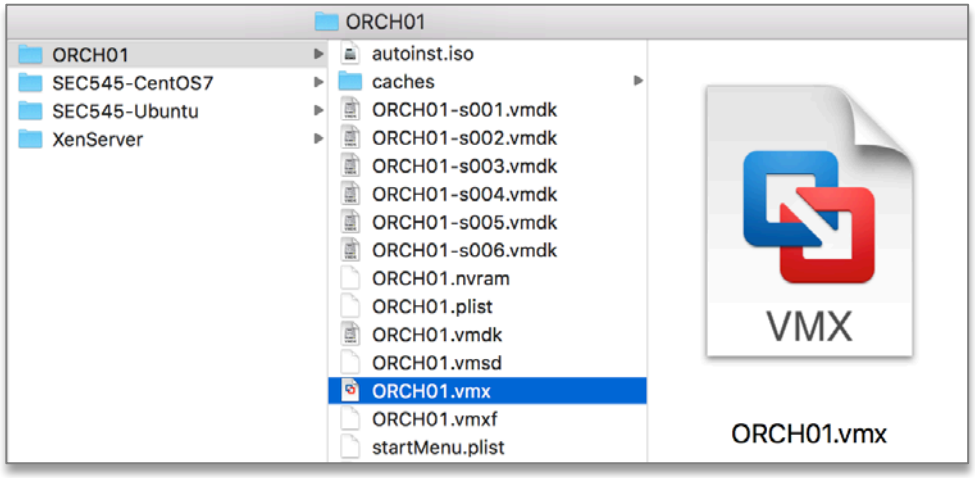
https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html

Download and install VMWare Fusion for Mac:

https://www.vmware.com/products/fusion/fusion-evaluation.html

3. For each VM, double-click on the .vmx file, log in, and verify the network settings (step 5+).

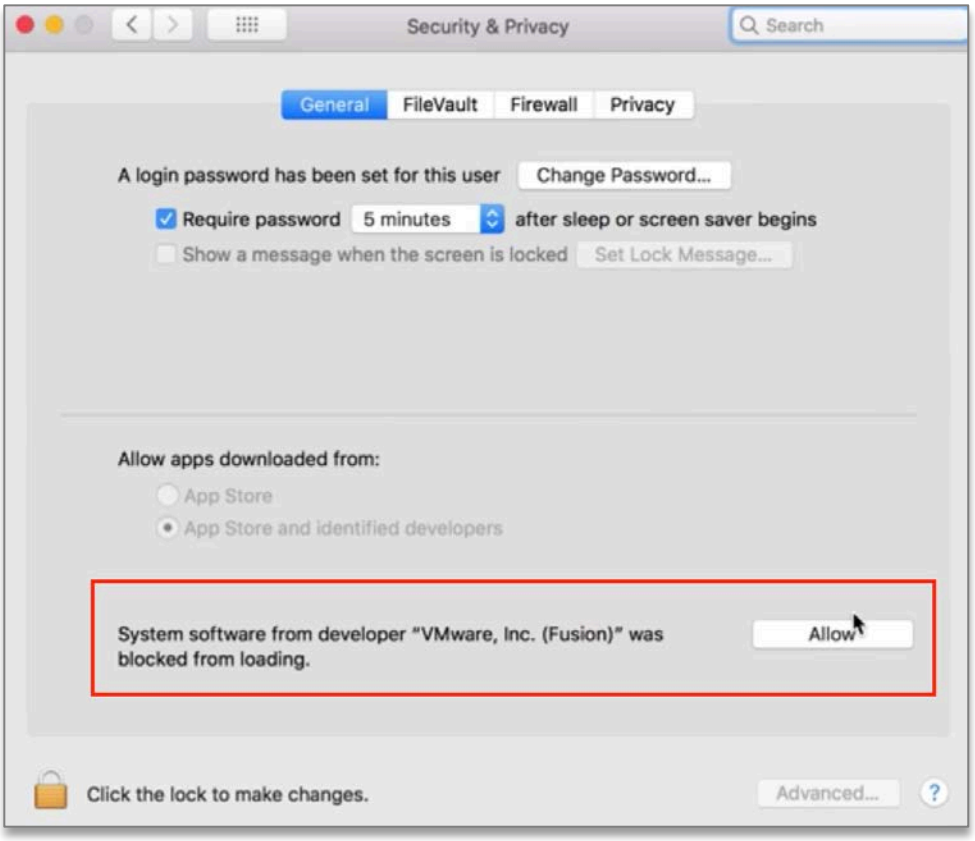
The details for the four VMs used in the labs are listed in the overview above. Select any default options.



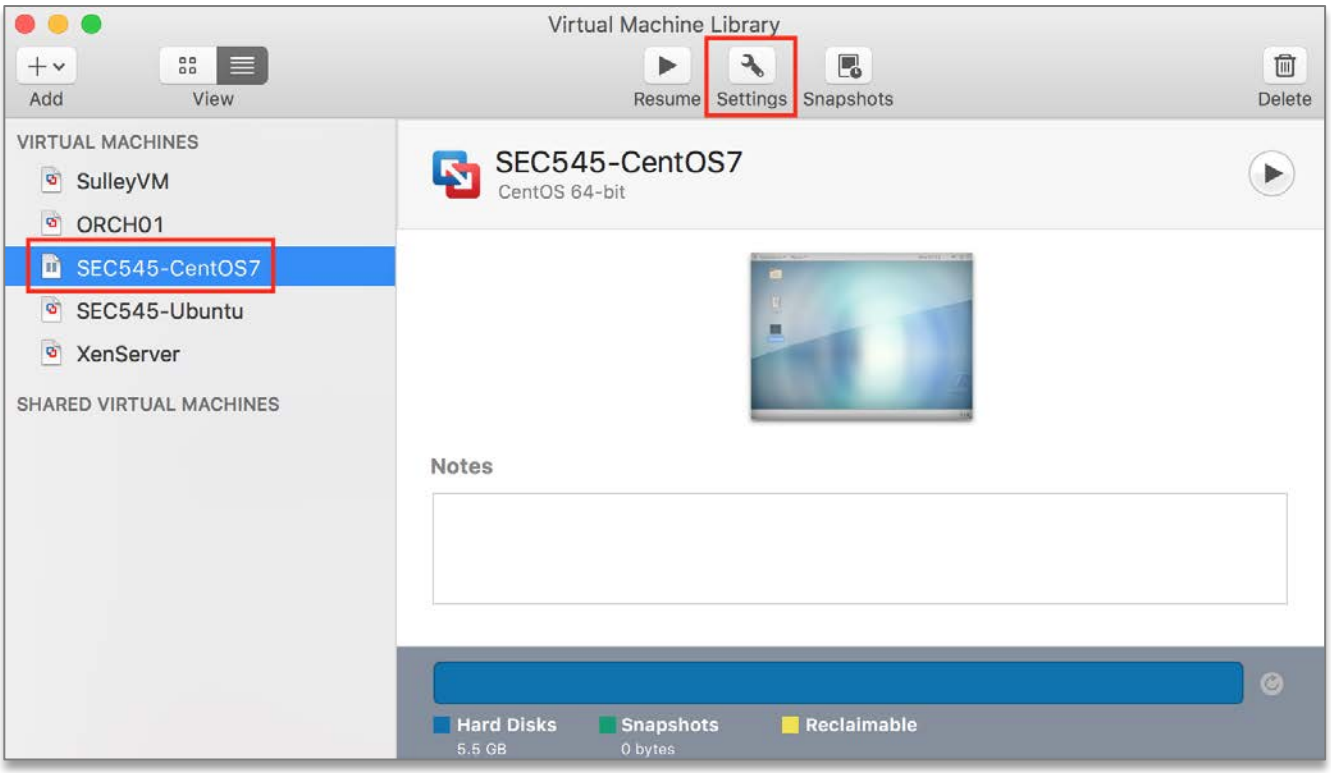
Mac Permissions Error:

If you get a permissions error on a Mac, click on the Apple icon on the top left, click on System Preferences, then Security and Privacy.

Allow the software from VMware to run that was blocked.

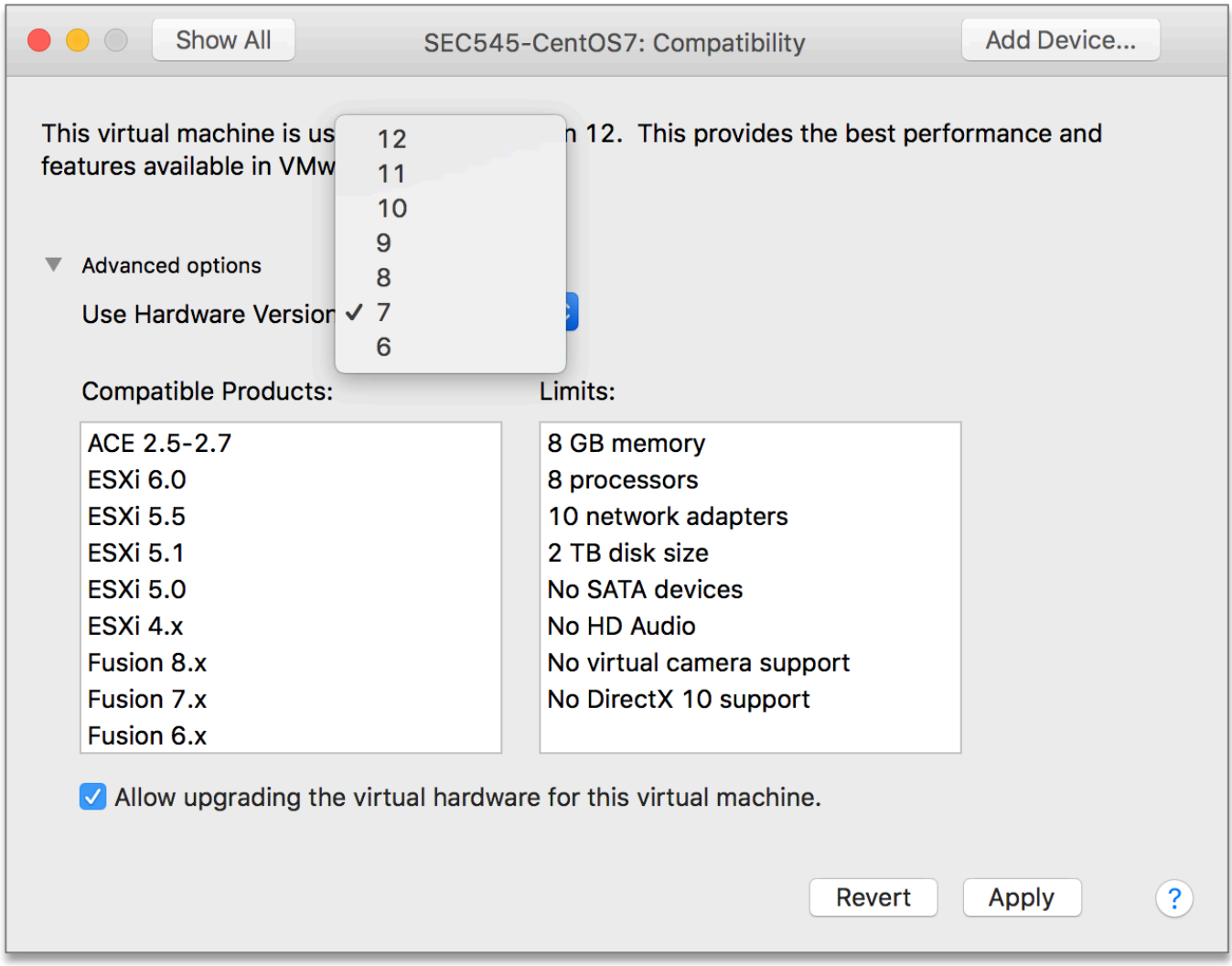


Mac: Incompatible Version



Click Compatibility

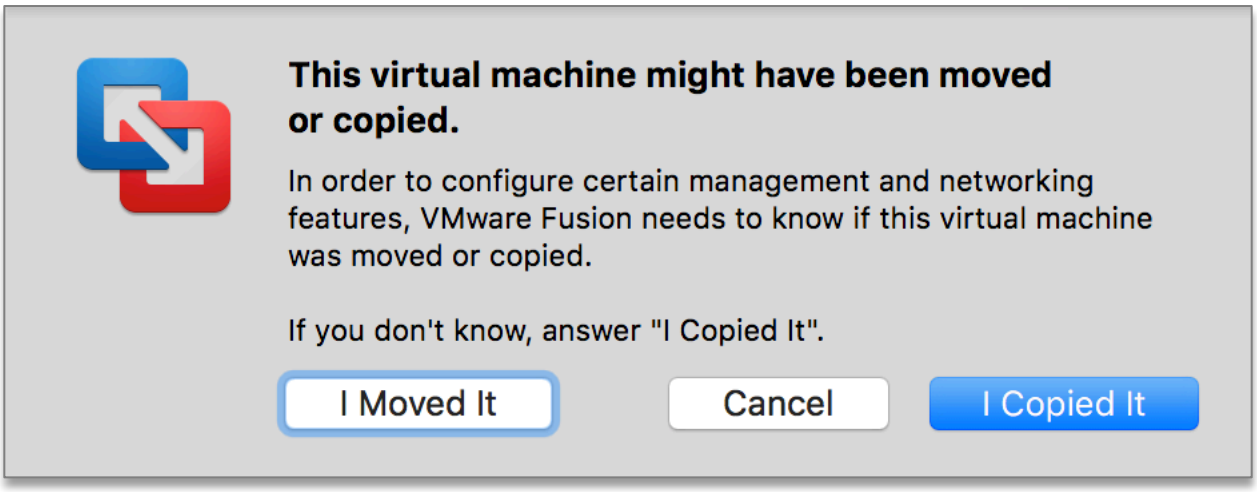
Change to version 7 and hit Apply.



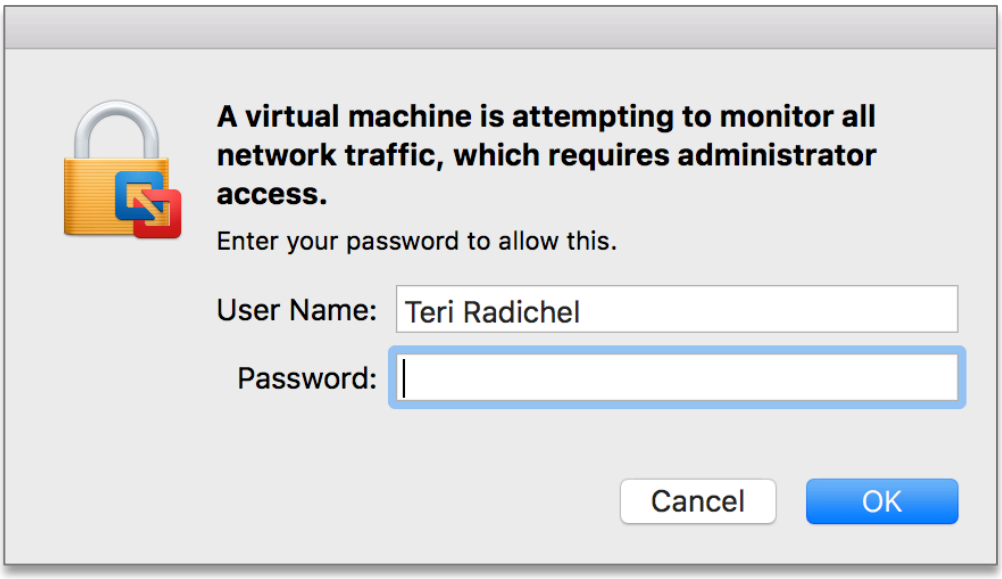
Windows: Incompatible Version

If you get an error saying “incompatible version” on Windows, edit the .vmx file in Notepad and change the “virtualHW.version” parameter from 12 to 7.

4. If you get a message asking if you moved or copied the VM, choose "I Moved It."



5. If you see a message on a Mac about a VM trying to monitor traffic, enter your credentials and click OK.



6. Verify Network Settings on each system by typing ifconfig and checking ens33 for the correct IP.

```
$ ifconfig
ens33     Link encap:Ethernet  HWaddr 00:0c:29:ad:22:b6
          inet addr:10.10.10.11  Bcast:10.10.255.255  Mask:255.255.0.0
          inet6 addr: fe80::c749:e2d2:f00:6d1e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:442 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:29540 (29.5 KB)
```

Tip: Use grep and -A1 to get the line after the string you're grepping to minimize the output.

```
$ ifconfig | grep ens33 -A1
ens33     Link encap:Ethernet  HWaddr 00:0c:29:ad:22:b6
          inet addr:10.10.10.11  Bcast:10.10.255.255  Mask:255.255.0.0
```

Verify that each VM has the correct IP address:

ORCH01 IP: 10.10.10.11

SEC545-CentOS7 IP: 10.10.10.10

SEC545-Ubuntu IP: 10.10.10.9

XenServer IP: 10.10.10.30

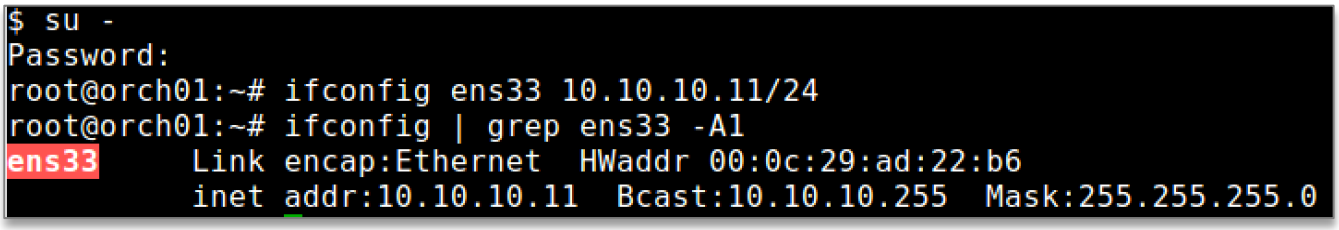
For students who plan to use the SEC545-Ubuntu system for AWS CLI labs, you will also need its second interface to be connected to the internet. This interface, ens34, is already configured for "Bridged" mode and should automatically connect to your laptop's internet-enabled interface. If you need to acquire a new DHCP address for this interface, type the following at the command line:

```
su –
[enter root password Passw0rd]
dhclient ens34
```

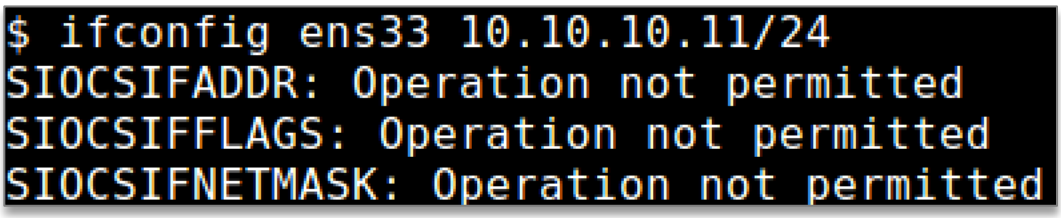
7. Change the ens33 IP address if necessary by logging in as root (if not root already) and using ifconfig.

```
su –
[enter root password Passw0rd]
ifconfig ens33 [correct IP address]/24
```

For example, on the ORCH01 VM:



If you did not use su – to switch to root first, you will see this error.



8. While logged into ORCH01, verify you can ping the SEC545-CentOS7 and SEC545-Ubuntu VMs.

CentOS:

```
ping 10.10.10.10
```

Ubuntu:

```
ping 10.10.10.9
```

9. While logged into the XenServer VM Verify you can ping the SEC545-CentOS7 VM.

CentOS:

```
ping 10.10.10.10
```