Lab-5: Aamzon Web Service (AWS)

Objectives

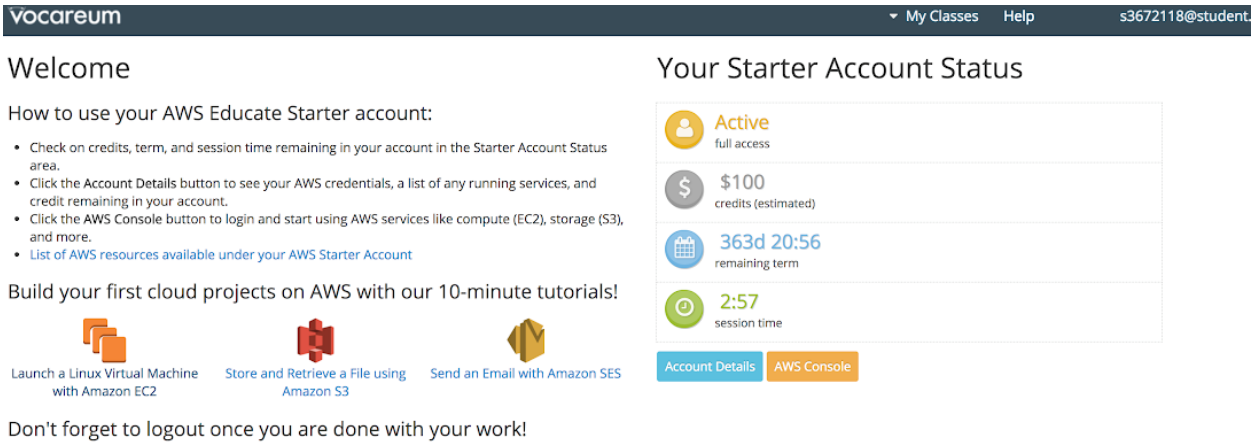
- Learn how to create and configure a virtual machine in AWS
- Learn how to communicate with virtual machines
- Learn how to use S3 and DynamoDB
- Learn how you can build your own platform by installing services.

Requirements

- You signed up for AWS educate and have access to your AWS console
- You have created your security keypair (.pem and/or .ppk file)
- You have created your security credentials

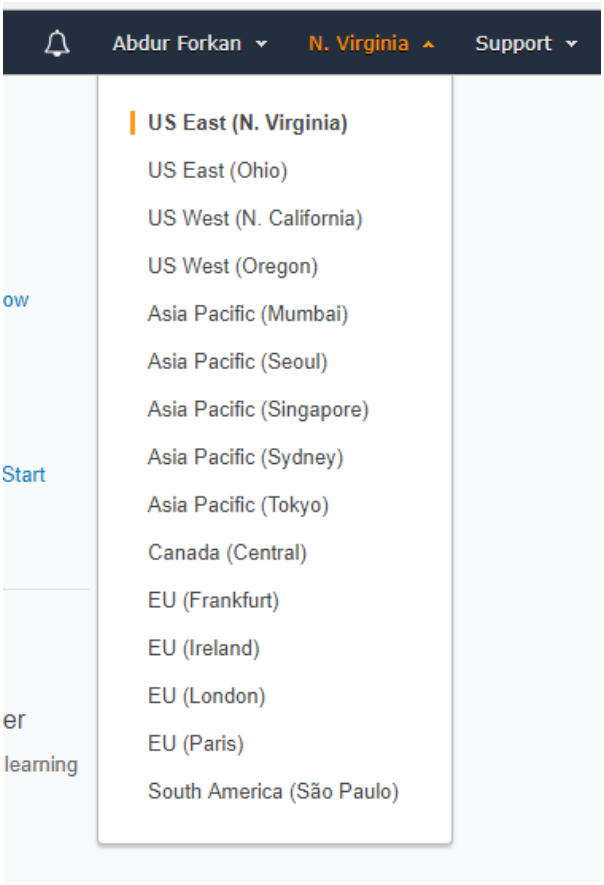
1 AWS Educate

You need to login to your AWS Educate and click on the **AWS Console** button.

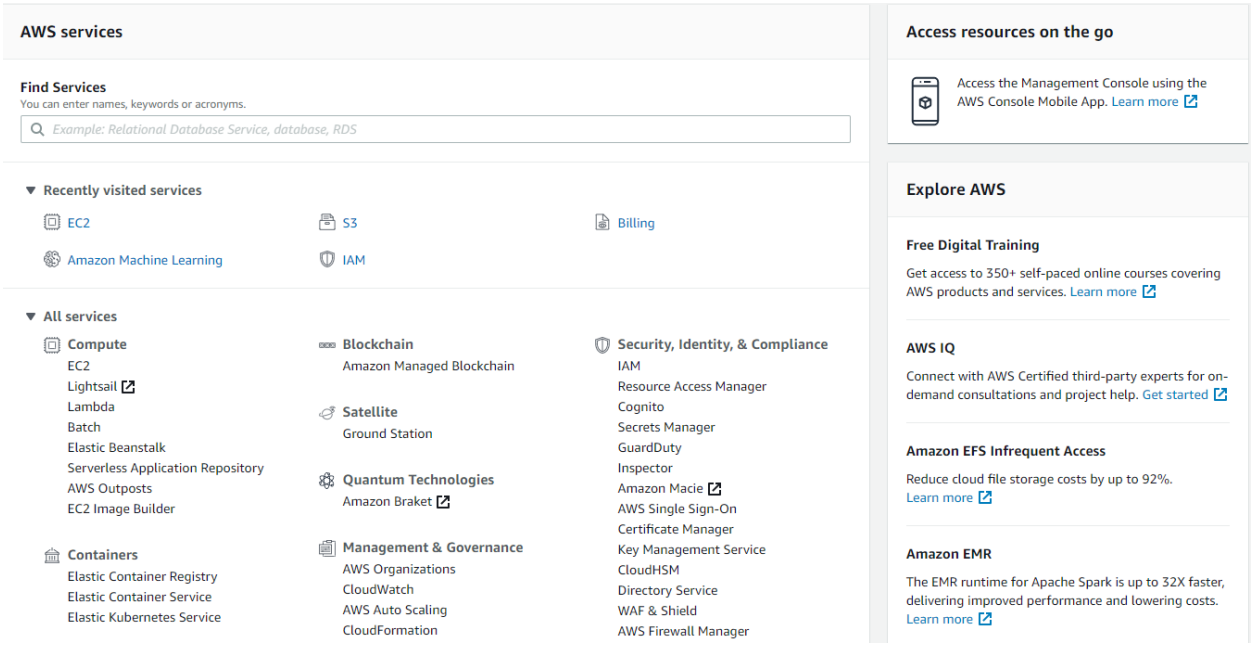


2 Create an EC2 instance

1- Go to your AWS console. Make sure that you use **US East (N. Virginia)** as your region.



2- Click on Services menu. You will see the following main console.

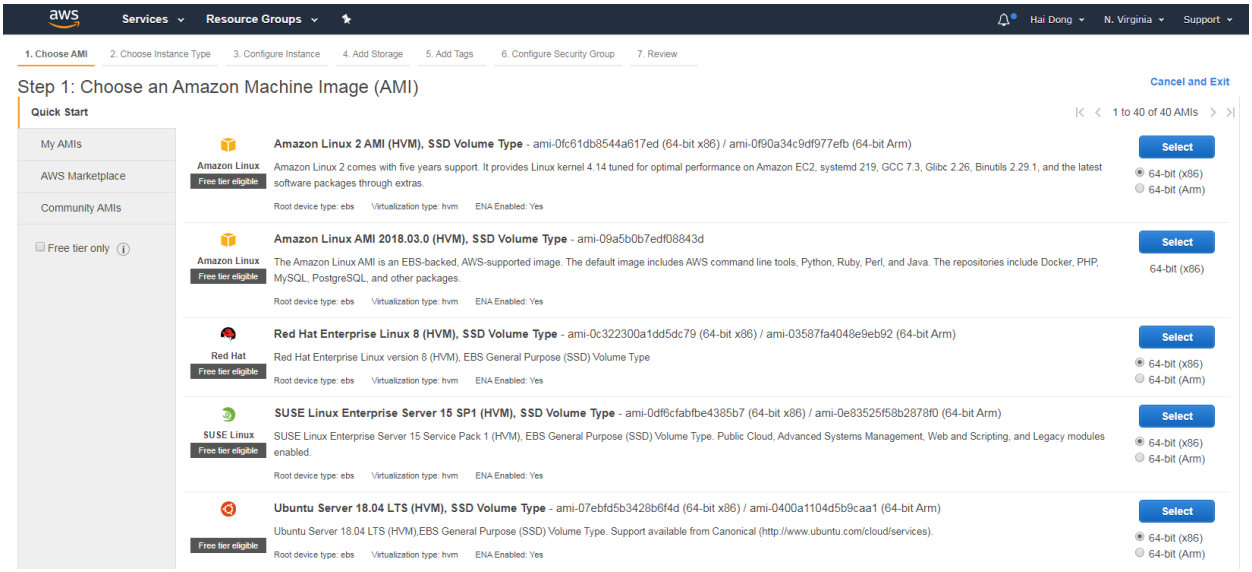


3- Select EC2 from Compute section.

4- From the next page click "**Launch Instance**". In this example we will launch a default Amazon Linux Instance and will configure this virtual machine

2.1 Create a linux virtual machine (VM)

1- For this lab we will only do **Ubuntu Server 18.04 (64-bit)**.



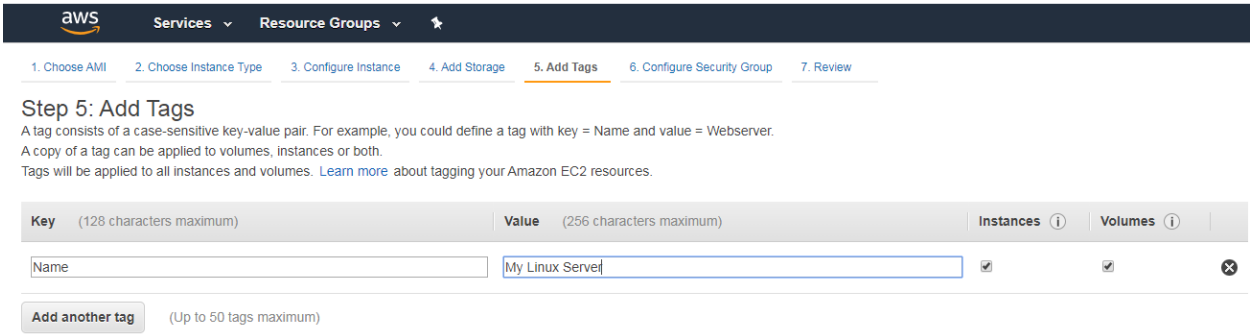
2- Choose an Instance Type – **t2.micro** instance type is already selected. Click "**Next: Configure Instance Details**" to go to Step 3.

Note in step 2 you are choosing the resources (e.g. number of CPU cores, RAM, hardware storage) of your virtual machine.

3- Configure Instance Details – Keep everything as default and click at "**Next: Add Storage**"

4- Keep the storage details as default and click on "**Next: Tag Instance**"

5- Tag Instance: Choose a name you prefer for your instance and type it into the "Value" field. I named it "MyLinuxServer".



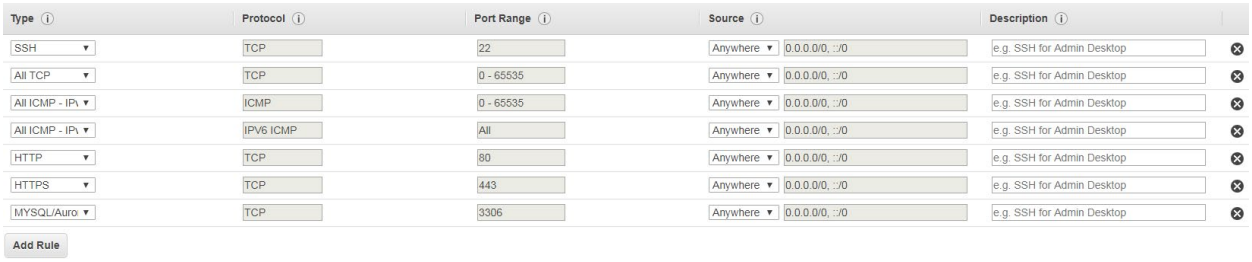
This name, more correctly known as a tag, will appear in the console once the instance launches. It makes it easy to keep track of running machines in a complex environment. Click, **Next: Configure Security Groups** to continue on to Step 6.

6- Configure Security Group: which is a similar concept to set firewall rules. Rename the security group with a suitable name (e.g. MyLinuxSecurityGroup). You can also add a description to your Security Group. Port 22 (SSH) is a default rule. Select, **Add Rule** to open the following ports.

a) SSH
b) HTTP
c) HTTPS
d) All TCP
e) All ICMP - IPv4
f) All ICMP – Ipv6
g) MYSQL/Aurora

For this Lab, select source **Anywhere** for all. But in real-life it should not be like that. You will need to add a custom ip or My IP. For example, if you select my IP for SSH then you will be able to SSH only from your current machine.

After adding the ports the screen will look like the following



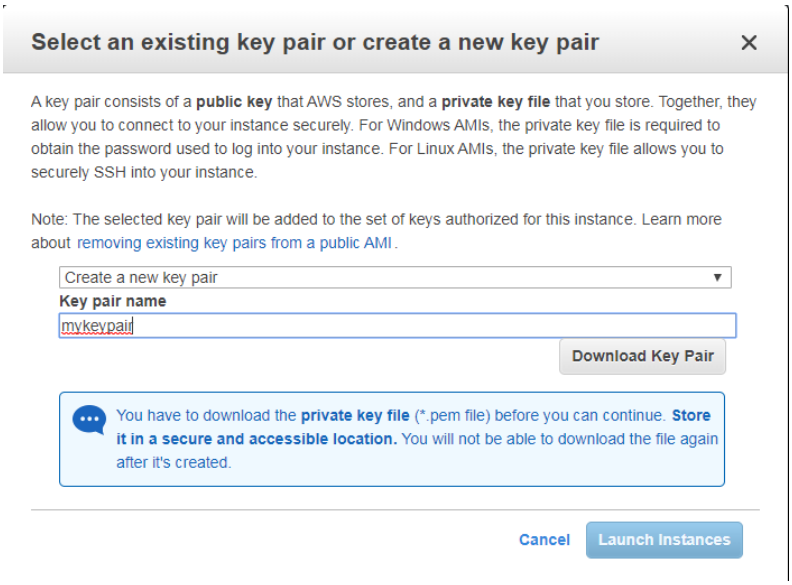
After that, Click "**Review and Launch**" to continue on to Step 7.

7- Review your choices, and then click "**Launch**".

8- You will receive a popup window to select a "**key pair**" or **create a new one**. Select the keypair if you already created one. Else, create a new one.

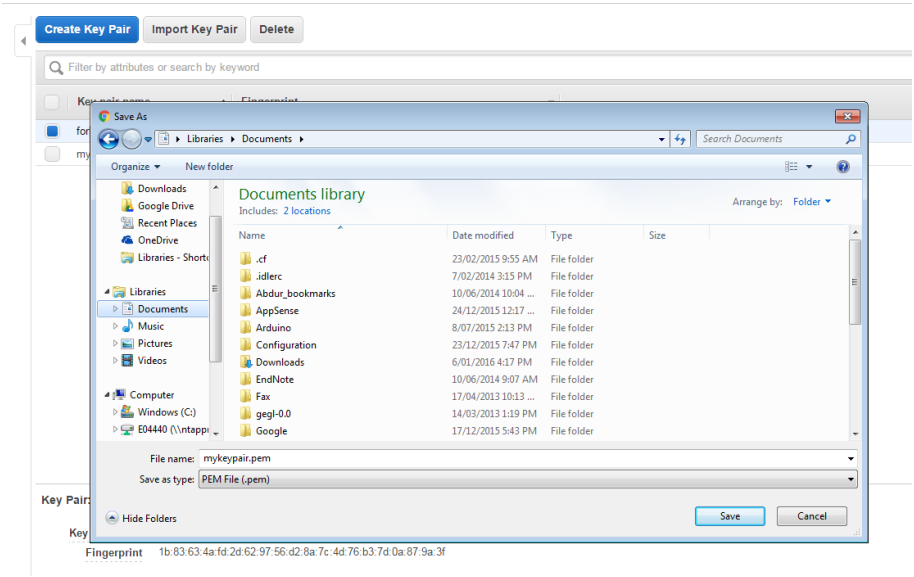
If you need to create a key pair follow the steps, (else skip these steps and go to **step 9**).

I. In the Key pair name textbox type a suitable name for your keypair



II. Click on Create

III. Key pair will be created and popup window will appear to save the **.pem** file for the keypair.



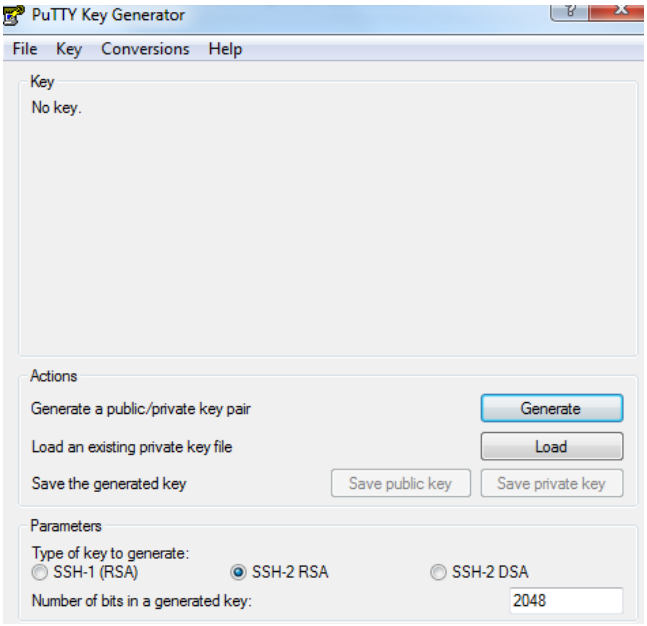
IV. Save the **.pem** file in a known location.

V. (This step is for MAC/Linux users only). Open command line console and change the permission of your **.pem** file to 400 by executing the command **chmod 400 [your_pem_file]**

VI. (This step is for window users only) If you are a windows user you need to convert this .pem file to .ppk file. You will need PuTTYgen to convert .pem file to .ppk file. If you are using personal laptop and PuTTYgen is not there then download the PuTTYgen first.

- Start PuTTYgen.

Under Type of key to generate, select SSH-2 RSA

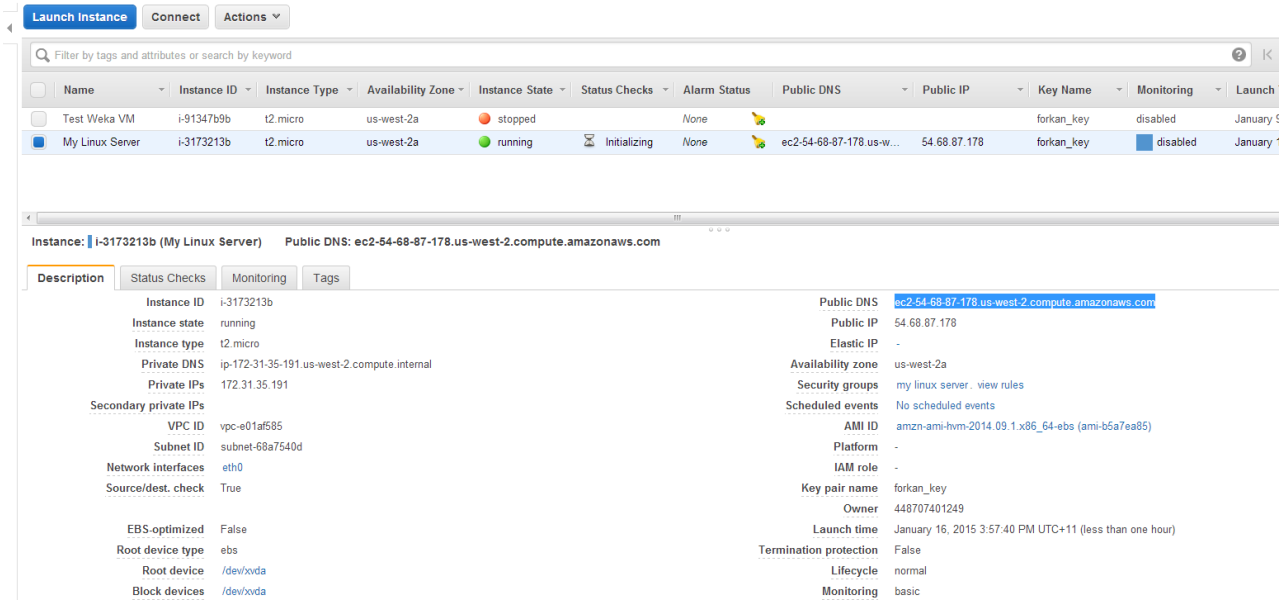


- Click **Load**. By default, PuTTYgen displays only files with the extension .ppk. To locate your .pem file, select the option to display files of **all types**.
- Select your .pem file and click **Open**. Click **OK** to dismiss the confirmation dialog box.
- Click **Save private key** to save the key in the format that PuTTY can use.
- Specify the same name for the key that you used for the **.pem** key pair. PuTTY automatically adds the **.ppk** file extension.

9- Next in your AWS console you will see a status page, notifying you that your instances are launching. Click "**View Instances**" to continue.

10- Your instance will be displayed in the list of running EC2 instances. If the instance state is not running, wait a few moments and press the double-arrow icon to refresh the list. When it is **running**, continue on to the next step.

11- Select the instance to display the instance details. Identify the **Public DNS** value and copy it to the clipboard.



2.2 Connect to your Virtual machine Instance

Option 1 is for MAC/Linux user and Optissh on 2 is for Windows

Option 1:

If you are using linux or mac OS **ssh** command should work directly from terminal/command window.

To connect with you virtual server, open the terminal window and type the following command.

ssh -i [your_key_file_location]/[yourkeyfilename].pem ubuntu@[yourinstancepublicDNS]

(**example**: ssh -i /usr/home/myLinuxKey.pem ubuntu@ec2-54-68-87-178.us-west-2.compute.amazonaws.com)

Please provide the location and name of your key file (.pem file) correctly have saved before.

If any yes/no question appears type yes then enter. Now you are connected to your Amazon instance.

In MAC/Linux if you see error something like Key is too open the change the permission of the key by chmod command: chmod 400 [your_key_file_location]/[yourkeyfilename].pem

(example: chmod 400 /usr/home/myLinuxKey.pem)

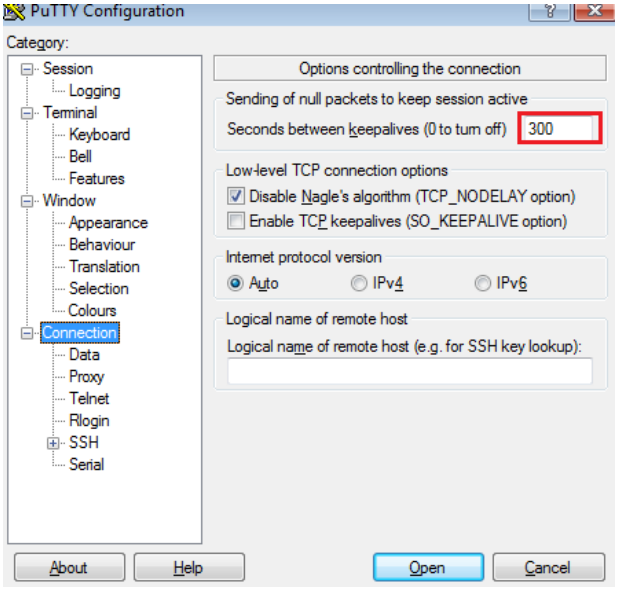
And then try again to connect. If you still see any error please contact with your tutor.

Option 2:

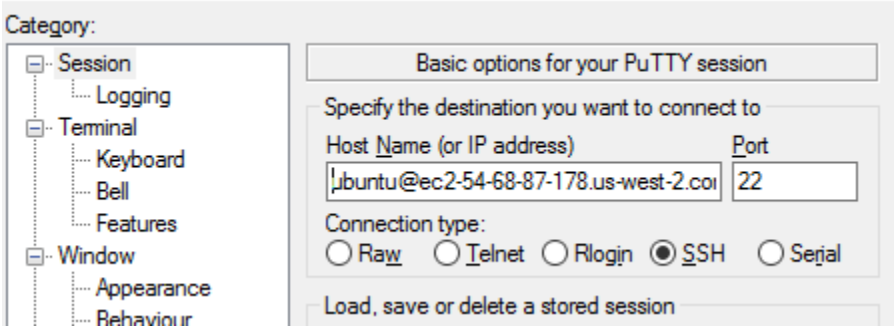
If you are using windows you will need PuTTY to ssh your virtual machine. If you don't have it already please download it.

Your private key is the **.ppk file** you created before to use with PuTTY. You can now connect to your instance using PuTTY's SSH client.

2- Click on connection and put 300 in Seconds between keep alive as in the following screen.



3- Now click on Session. In the Host Name box, enter **ubuntu@[yourinstancepublicDNS] (**example: ubuntu@ec2-54-68-87-178.us-west-2.compute.amazonaws.com**).** Ensure that Port is **22** and Connection type is **SSH**



4- Now in the **Category** pane, expand **Connection**, expand **SSH**, and then select **Auth**. Complete the following:

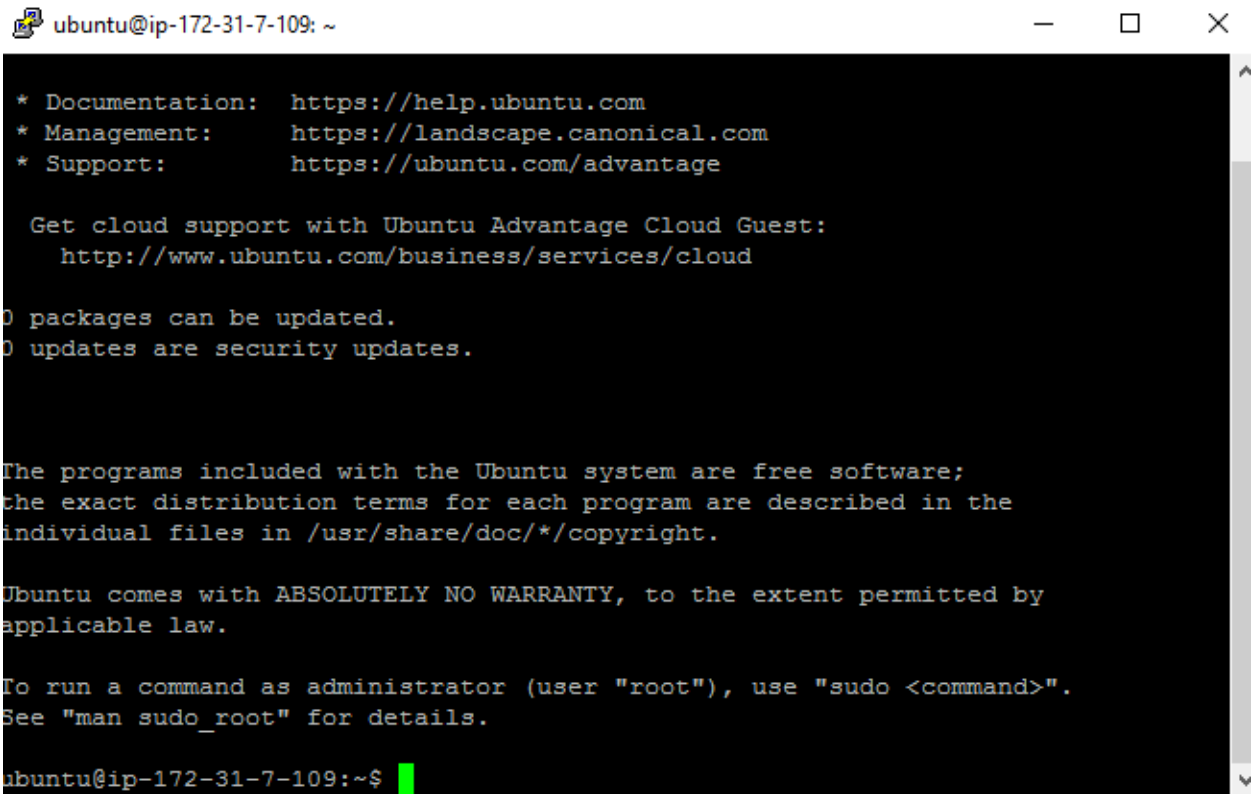
- Click **Browse**.
- Select the **.ppk** file that you generated for your key pair, and then click Open.

5- If you plan to start this session again later, you can save the session information for future use. Select **Session** in the Category tree, enter a name for the session in Saved Sessions, and then click **Save**.

6- Click **Open** to start the PuTTY session.

7- PuTTY displays a security alert dialog box that asks whether you trust the host you are connecting to.

8- Click **Yes**. A window opens and you are connected to your instance.



Now think what you have done. Here AWS is providing you infrastructure. That is a Virtual machine is a specific hardware configuration. Now for example, you need to build platform on it. For example, you want to have java runtime environment in this VM so you can run your java code here. Also you want to use this VM as a hosting server and you want to use apache. Now we will see how to do this.

3 Customize your VM instance

3.1 Install JDK

1- Install latest JDK using the following commands (click yes/ enter key in each screen you get in this installation process).

```
sudo add-apt-repository ppa:linuxuprising/java
```

2- Enter the following command to update your Ubuntu packages

```
sudo apt update
```

3- Enter the following command to install JDK 14 (click yes/ enter key in each screen you get in this installation process)

```
sudo apt install oracle-java14-installer
```

Note: Click “Esc” button in your keyboard when you see the agreement page, and then click “Yes” button on the next page to accept the agreement.

4- Now check the java version using the command

```
java -version
```

5- It should show java version 14 as default version.

6- Now let’s setup java home environment variable by using the following commands

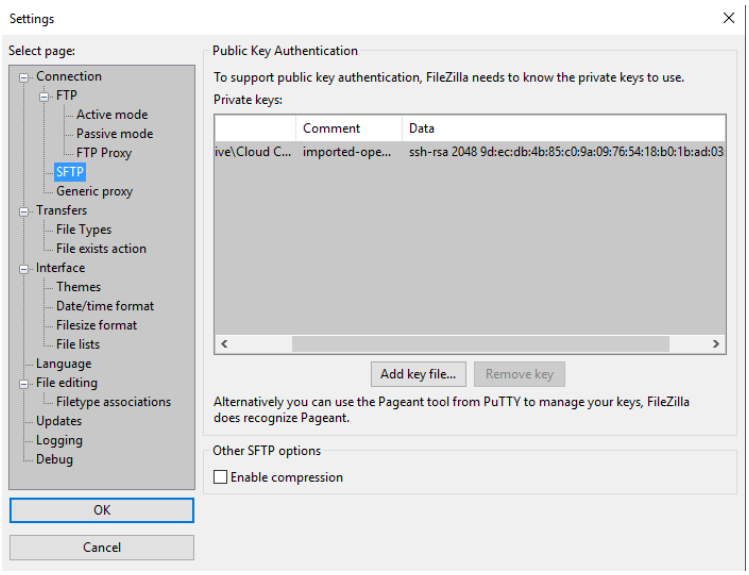
```
export JAVA_HOME=/usr/bin/java
export PATH=$JAVA_HOME/bin:$PATH
```

7- Now you can run any executable .jar file or .class file in the virtual machine

3.2 Upload files at your VM

1- Open FileZilla (if you already have it in your computer otherwise download or install it). Alternatively, you can use WinFCP if you are already familiar with it.

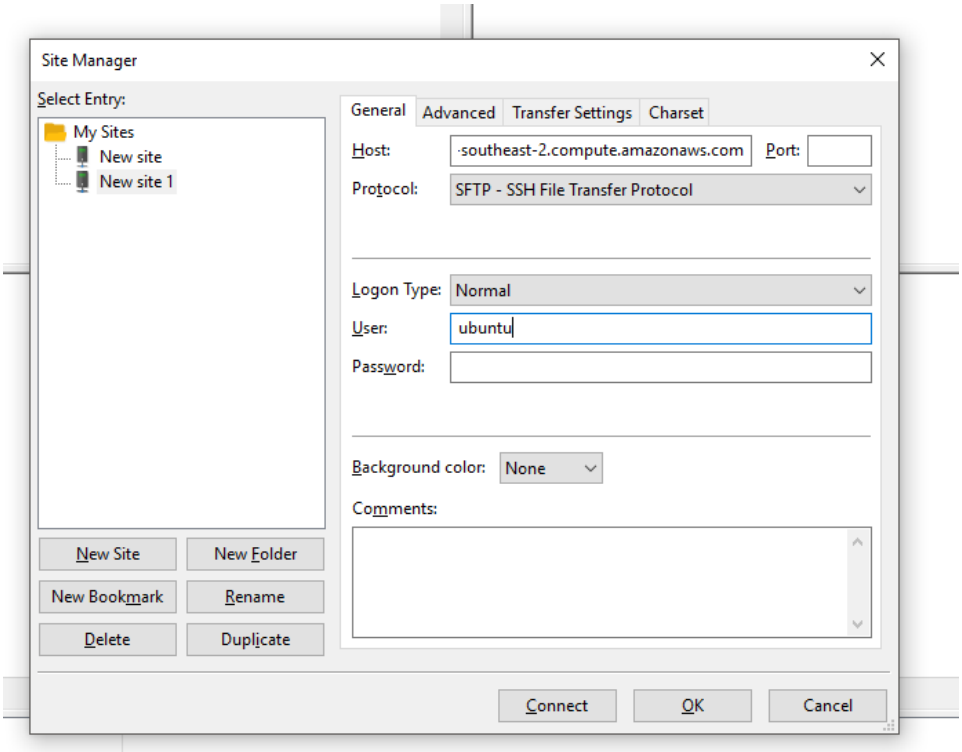
2- In Filezilla click on Edit>Settings. Then Select SFTP under connection. Click on Add Key file and browse to your .pem file. The key will be added. Then Click OK.



3- Click Ok.

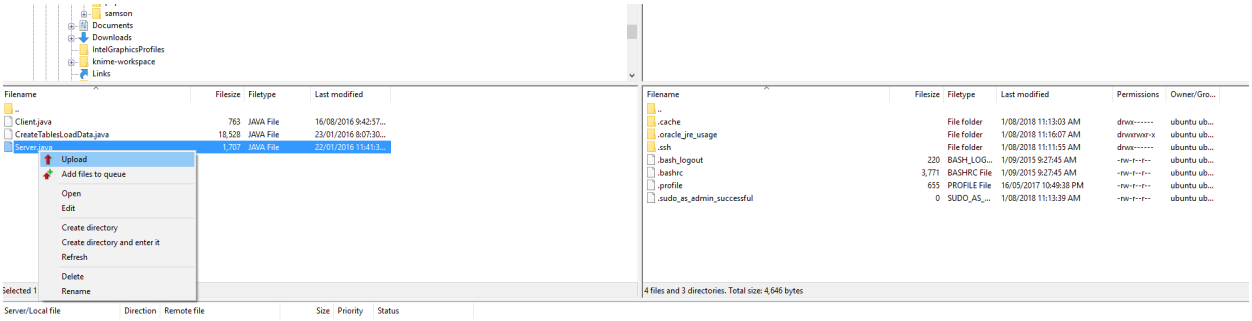
4- Now click File>Site Manager>New Site

5- In host name provide [your_public_dns], Port: 22 and Protocol: SFTP. In Logon type select “Normal” and in user type “ubuntu” as in the screen.



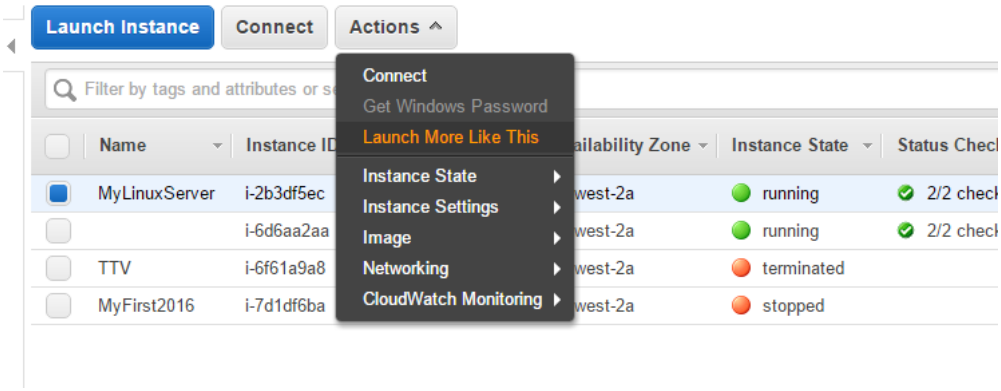
You can now able to see home directory of Ubuntu user.

7- Download Server.java from Canvas and upload this file on your VM using FileZilla.



3.3 Duplicate your instance

1- Select the instance that we just created (e.g. MyLinuxServer) from AWS management console (as in the figure). Then Select Action>Launch More Instance like this



2- In next step go to the Tags section and click on Edit Tags. Give the name of this instance as **MyLinuxClient**

3- Select review and Launch the instance and then **Launch** the instance.

4- Open another PuTTY console/terminal, connect to **MyLinuxClient** using the same process and install jdk 11 by following the exactly the same process.

3.4 Create a client-server communication using 2 VMs

1- Now download Client.java from Canvas. Open the file and modify the **localhost** to **public IP** of your **MyLinuxServer**

2- Upload the code Client.java to MyLinuxClient using FileZilla by following same process described previously. Here you don't need to Add key in filezila again. Just create a new connection.

Then modify the **localhost** to **public IP** of your MyLinuxServer (you can do it using vi or nano editor on linux virtual machine). Alternatively, you can download it to your local machine using WinSCP/Filezilla and re-upload the modified version to your Virtual Machine instance.

3- Now go back to PuTTY console/terminal of MyLinuxServer. Compile the Server.java code using the command **javac Server.java**

4- Similarly move to PuTTY console/terminal of MyLinuxClient and compile the java code using the command **javac Client.java**

5- Go back to PuTTY console/terminal of MyLinuxServer. Now run the Server using the command **java Server**. You will see the message Server is now running at port: 1113

6- Go back to PuTTY Console of **MyLinuxClient** and run Client using command **java Client.** You will find the your two VMs are communicating using TCP protocol.

Now close the console/terminal of your MyLinux Client.

3.5 Prepare your apache server

1- Install apache using the following command

```
sudo apt-get install apache2
```

2- Install php using the following commands

```
sudo apt-get install php libapache2-mod-php
sudo a2enmod mpm_prefork && sudo a2enmod php7.0
```

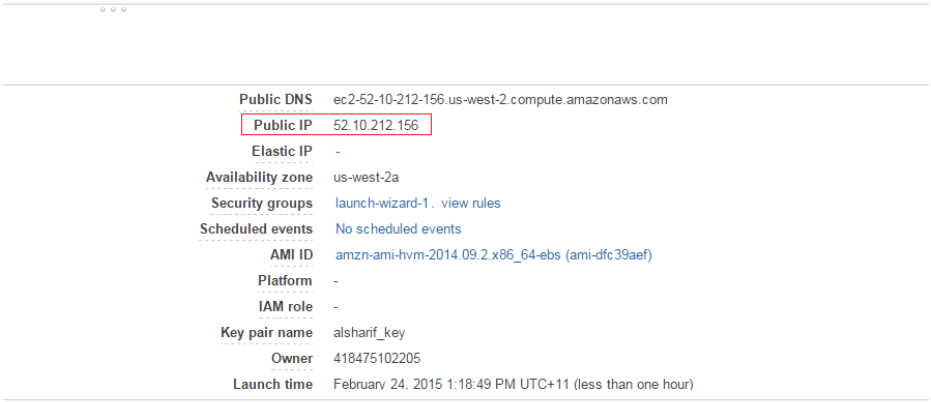
3- Start the Apache web server using the following commands

```
sudo service apache2 start
```

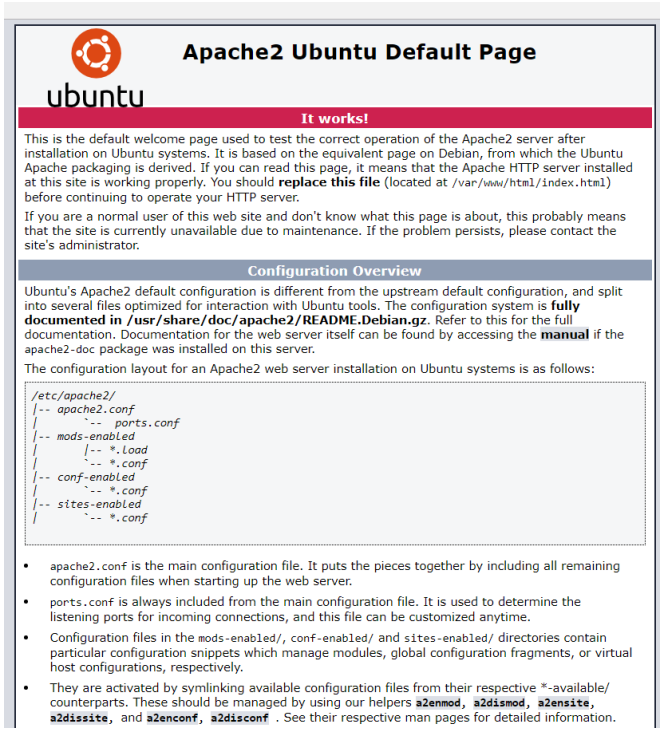
3- If you see apache already started and running before then restart apache

```
sudo service apache2 restart
```

4- Test your web server. In a web browser, enter the public DNS address (or the public IP address) of your instance. You can get the public DNS for your instance using the Amazon EC2 console (check the **Public DNS** column; if this column is hidden, click the **Show/Hide** icon and select **Public**



5- You should see apache test page.



So, here we installed java, php and apache on our first VM instance.

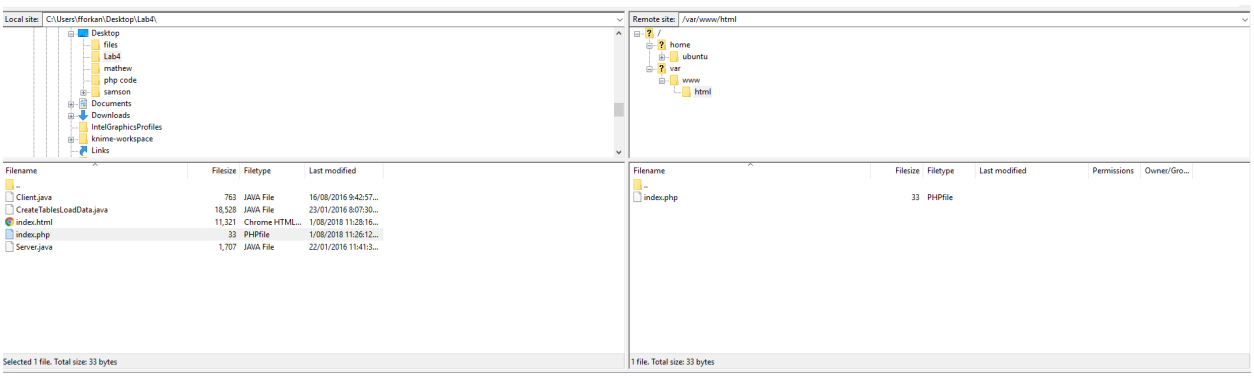
6- Now create a file named **index.php** which prints hello world.

```
<?php

echo "Hello World!";

?>
```

7- Go back to the console of your **MyLinuxServer** and set permission to your webroot folder by the command:

```
sudo chmod -R 777 /var/www/html
```

7- Now upload the **index.php** to **/var/www/html/** of your **MyLinuxServer** Virtual Machine instance using FileZilla. Also remove the index.html file from /var/www/html. It will look like the following.



8- Browse you public ip using web browser again. Now you should see a php hello world page instead of Apache test page.

3.6 Amazon S3

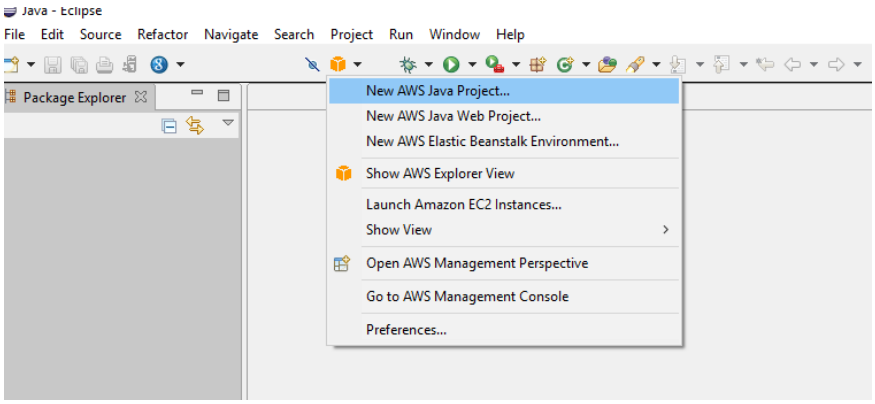
From AWS console dashboard “Services”> "Storage" select S3.

This is a similar feature for Amazon what we like GCS in Google. Try to create a bucket and some folders inside the bucket. Also upload some image files in one of your folder.

4 Amazon DynamoDB: Create Tables and Upload Data Using the AWS SDK

1- Open Eclipse

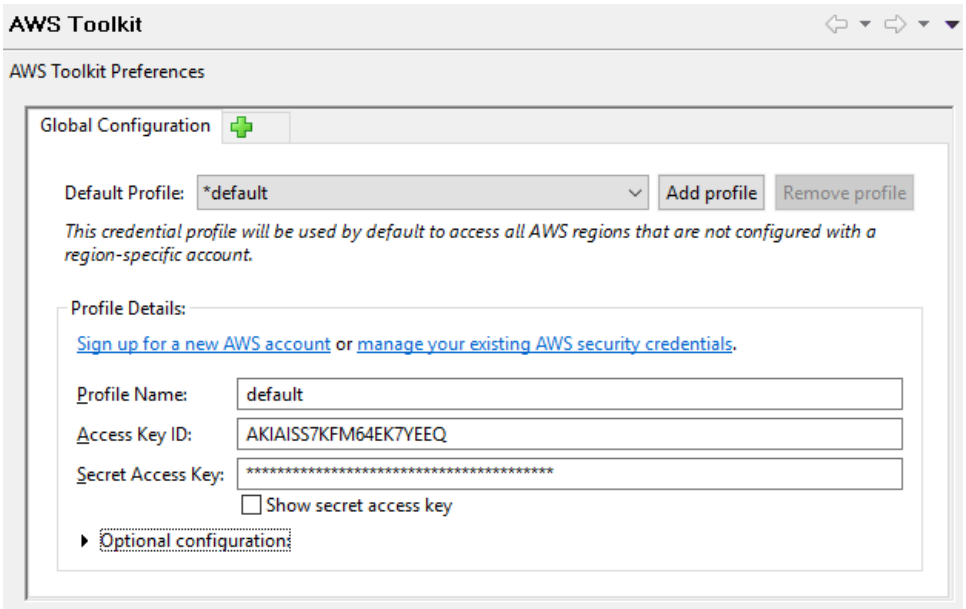
2- From Amazon icon select **New AWS Java project** as in the screen



3- In the Project name field, type a name for your project (e.g. DynamodbTest). Click Finish to create the project. Note that the project is pre-configured, and includes the AWS SDK for Java .jar files.

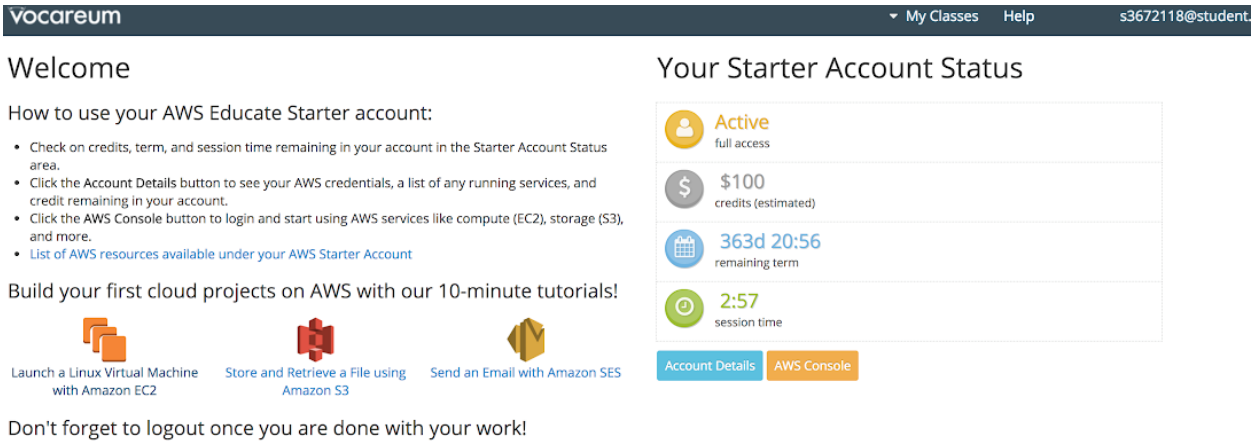
4- You will now need to create a default credential profiles file. This file enhances security by storing your credentials separately from your project directories, so that they cannot be unintentionally committed to a public repository. Click on AWS icon again and Select **Preferences**

5- Provide your Access Key Id and Secret Key. Then click ok.



If you have not created your keys before then follow the steps to create your keys:

- Go to AWS Educate Account. Go into AWS Account section



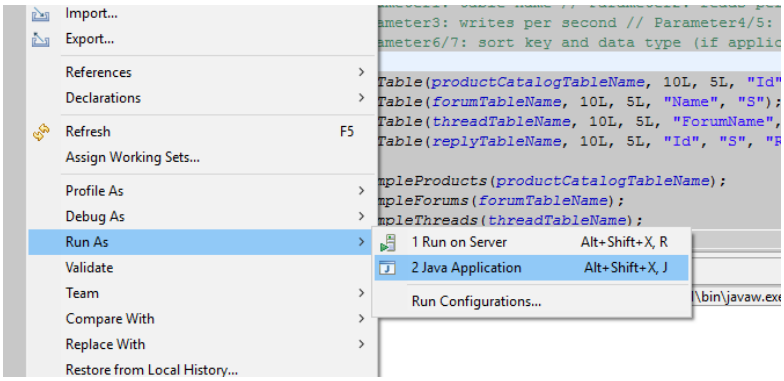
- In the popup menu select **Access Keys / Secret Keys**.

6- Now click on your project > src and create a new package named com.amazonaws.codesamples

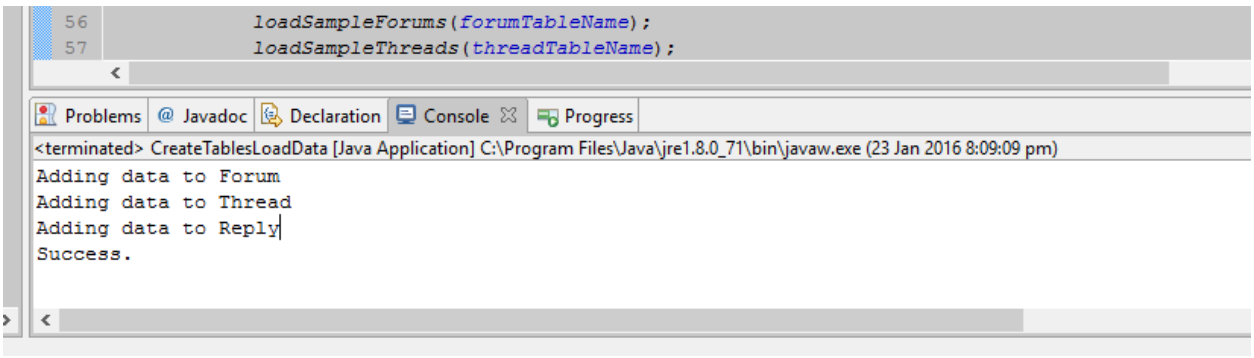
7- Right click on the package name and create a new class named CreateTablesLoadData

8- Now get the file CreateTablesLoadData.java from Canvas and copy paste the content to your CreateTablesLoadData.java class file. Then save it.

9- Right click on CreateTablesLoadData.java file. Then Select **Run as>Java application**



10- The application will start to run. Wait until you see success message in Eclipse console.



So entries are created in your dynamoDB. Try to understand what is inside the code and how it works. Can you find out in your AWS console that where these data are going?

Cleaning up!!! Don't leave your lab before doing these

1. Go to your AWS console
2. Click on instances
3. Select all of your instances
4. Go to action then "Stop" it.
5. Also delete your DynamoDB tables