

## Lab-5: Aamazon Web Service (AWS)

### Objectives

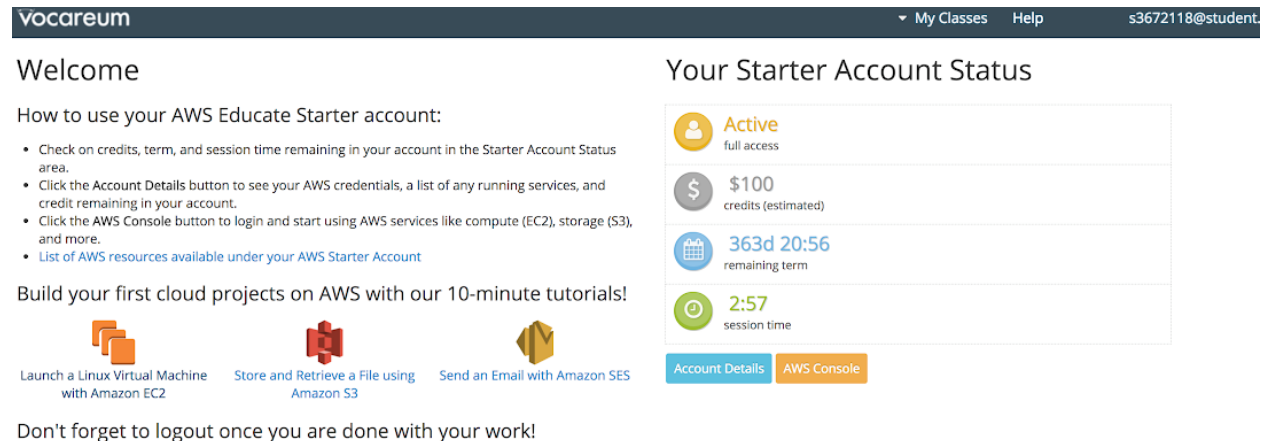
- Learn how to create and configure a virtual machine in AWS
- Learn how to communicate with virtual machines
- Learn how to use S3 and DynamoDB
- Learn how you can build your own platform by installing services.

### 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.




Vocareum My Classes Help s3672118@student

### Welcome


How to use your AWS Educate Starter account:

- Check on credits, term, and session time remaining in your account in the Starter Account Status area.
- Click the **Account Details** button to see your AWS credentials, a list of any running services, and credit remaining in your account.
- Click the **AWS Console** button to login and start using AWS services like compute (EC2), storage (S3), and more.
- [List of AWS resources available under your AWS Starter Account](#)


Build your first cloud projects on AWS with our 10-minute tutorials!



Launch a Linux Virtual Machine with Amazon EC2







Store and Retrieve a File using Amazon S3



Send an Email with Amazon SES

### Your Starter Account Status

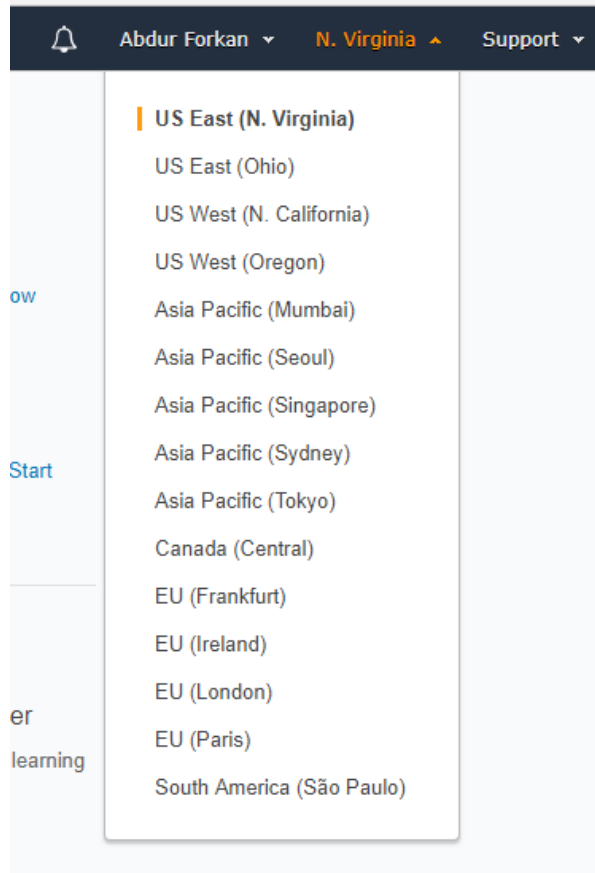
	<b>Active</b> full access
	<b>\$100</b> credits (estimated)
	<b>363d 20:56</b> remaining term
	<b>2:57</b> session time

[Account Details](#) [AWS Console](#)

Don't forget to logout once you are done with your work!

### 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.

### AWS services

**Find Services**  
You can enter names, keywords or acronyms.

**Recently visited services**

EC2
 S3
 Billing

Amazon Machine Learning
 IAM

**All services**

**Compute**  
 EC2  
 Lightsail [↗](#)  
 Lambda  
 Batch  
 Elastic Beanstalk  
 Serverless Application Repository  
 AWS Outposts  
 EC2 Image Builder

**Blockchain**  
 Amazon Managed Blockchain

**Satellite**  
 Ground Station

**Quantum Technologies**  
 Amazon Braket [↗](#)

**Management & Governance**  
 AWS Organizations  
 CloudWatch  
 AWS Auto Scaling  
 CloudFormation

**Security, Identity, & Compliance**  
 IAM  
 Resource Access Manager  
 Cognito  
 Secrets Manager  
 GuardDuty  
 Inspector  
 Amazon Macie [↗](#)  
 AWS Single Sign-On  
 Certificate Manager  
 Key Management Service  
 CloudHSM  
 Directory Service  
 WAF & Shield  
 AWS Firewall Manager

**Access resources on the go**

 Access the Management Console using the AWS Console Mobile App. [Learn more](#) [↗](#)

**Explore AWS**

**Free Digital Training**

Get access to 350+ self-paced online courses covering AWS products and services. [Learn more](#) [↗](#)

**AWS IQ**

Connect with AWS Certified third-party experts for on-demand consultations and project help. [Get started](#) [↗](#)

**Amazon EFS Infrequent Access**

Reduce cloud file storage costs by up to 92%. [Learn more](#) [↗](#)

**Amazon EMR**

The EMR runtime for Apache Spark is up to 32X faster, delivering improved performance and lowering costs. [Learn more](#) [↗](#)

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)**.

The screenshot shows the AWS Management Console interface for Step 1: Choose an Amazon Machine Image (AMI). The top navigation bar includes the AWS logo, Services, Resource Groups, and user information. The breadcrumb trail shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. The main content area is titled 'Step 1: Choose an Amazon Machine Image (AMI)' and includes a 'Quick Start' sidebar with filters for 'My AMIs', 'AWS Marketplace', and 'Community AMIs'. The 'Free tier only' filter is selected. The main list displays several AMIs, including Amazon Linux 2 AMI, Amazon Linux AMI 2018.03.0, Red Hat Enterprise Linux 8, SUSE Linux Enterprise Server 15 SP1, and Ubuntu Server 18.04 LTS. The Ubuntu Server 18.04 LTS AMI is highlighted, showing its details and a 'Select' button.

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”.

The screenshot shows the AWS Management Console interface for Step 5: Add Tags. The top navigation bar includes the AWS logo, Services, Resource Groups, and user information. The breadcrumb trail shows the steps: 1. Choose AMI, 2. Choose Instance Type, 3. Configure Instance, 4. Add Storage, 5. Add Tags, 6. Configure Security Group, 7. Review. The main content area is titled 'Step 5: Add Tags' and includes a description of tags. Below the description is a table with columns for 'Key', 'Value', 'Instances', and 'Volumes'. The 'Key' field is labeled 'Name' and the 'Value' field is labeled 'My Linux Server'. The 'Instances' checkbox is checked.

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

Type ⓘ	Protocol ⓘ	Port Range ⓘ	Source ⓘ	Description ⓘ	
SSH ▾	TCP	22	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
All TCP ▾	TCP	0 - 65535	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
All ICMP - IPv4 ▾	ICMP	0 - 65535	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
All ICMP - IPv6 ▾	IPv6 ICMP	All	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
HTTP ▾	TCP	80	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
HTTPS ▾	TCP	443	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
MYSQL/Auro ▾	TCP	3306	Anywhere ▾ 0.0.0.0/0, ::/0	e.g. SSH for Admin Desktop	✕
Add Rule					

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

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

Create a new key pair

Key pair name

mykeypair

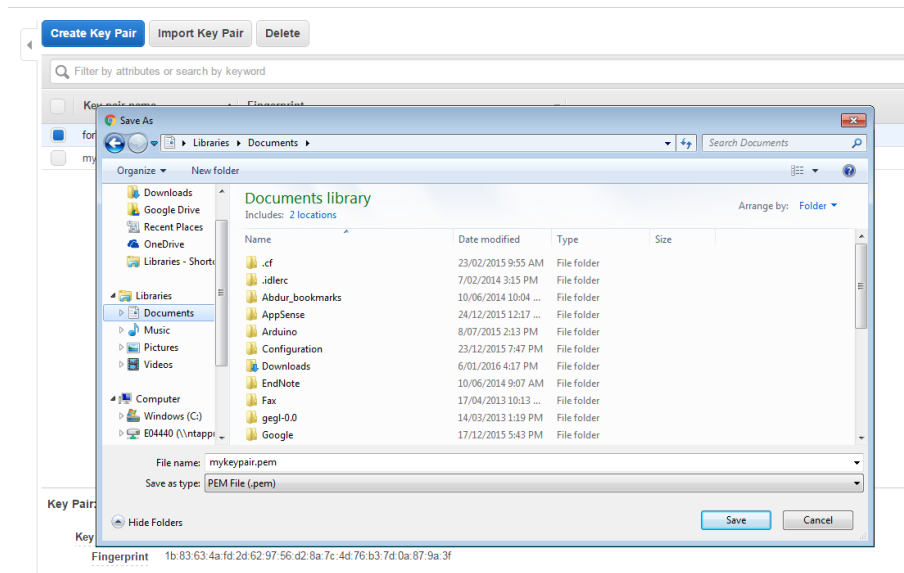
Download Key Pair

You have to download the **private key file** (\*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

Cancel

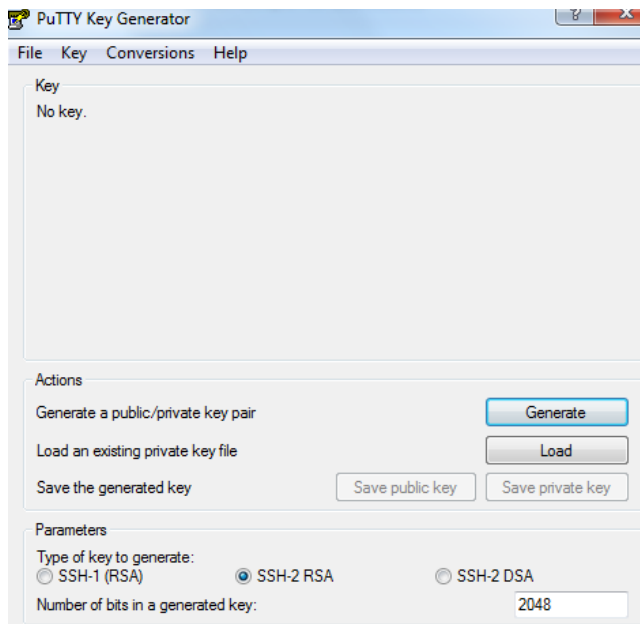
Launch Instances

- 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.

Launch Instance Connect Actions

Filter by tags and attributes or search by keyword

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP	Key Name	Monitoring	Launch
Test Weka VM	i-91347b9b	t2.micro	us-west-2a	stopped		None			forkan_key	disabled	January 5
My Linux Server	i-3173213b	t2.micro	us-west-2a	running	Initializing	None	ec2-54-68-87-178.us-w...	54.68.87.178	forkan_key	disabled	January 5

Instance: i-3173213b (My Linux Server) Public DNS: ec2-54-68-87-178.us-west-2.compute.amazonaws.com

Description	Status Checks	Monitoring	Tags
Instance ID	i-3173213b		
Instance state	running		
Instance type	t2.micro		
Private DNS	ip-172-31-35-191.us-west-2.compute.internal		
Private IPs	172.31.35.191		
Secondary private IPs			
VPC ID	vpc-e01af585		
Subnet ID	subnet-68a7540d		
Network interfaces	eth0		
Source/dest. check	True		
EBS-optimized	False		
Root device type	ebs		
Root device	/dev/xvda		
Block devices	/dev/xvda		
Public DNS	ec2-54-68-87-178.us-west-2.compute.amazonaws.com		
Public IP	54.68.87.178		
Elastic IP	-		
Availability zone	us-west-2a		
Security groups	my linux server, view rules		
Scheduled events	No scheduled events		
AMI ID	amzn-ami-hvm-2014.09.1.x86_64-eks (ami-b5a7ea85)		
Platform	-		
IAM role	-		
Key pair name	forkan_key		
Owner	448707401249		
Launch time	January 16, 2015 3:57:40 PM UTC+11 (less than one hour)		
Termination protection	False		
Lifecycle	normal		
Monitoring	basic		

## 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`)

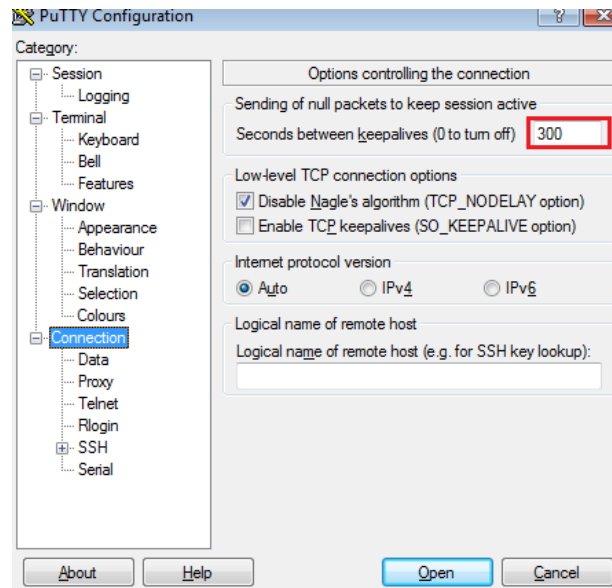
And then try again to connect. If you still see any error please contact with your tutor.

### Option 2:

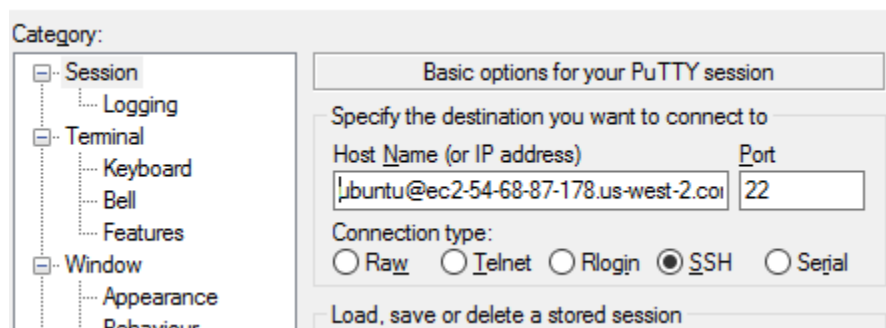
If you are using windows you will need PuTTY to ssh your virtual machine. If you don't have it already please download it.

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.

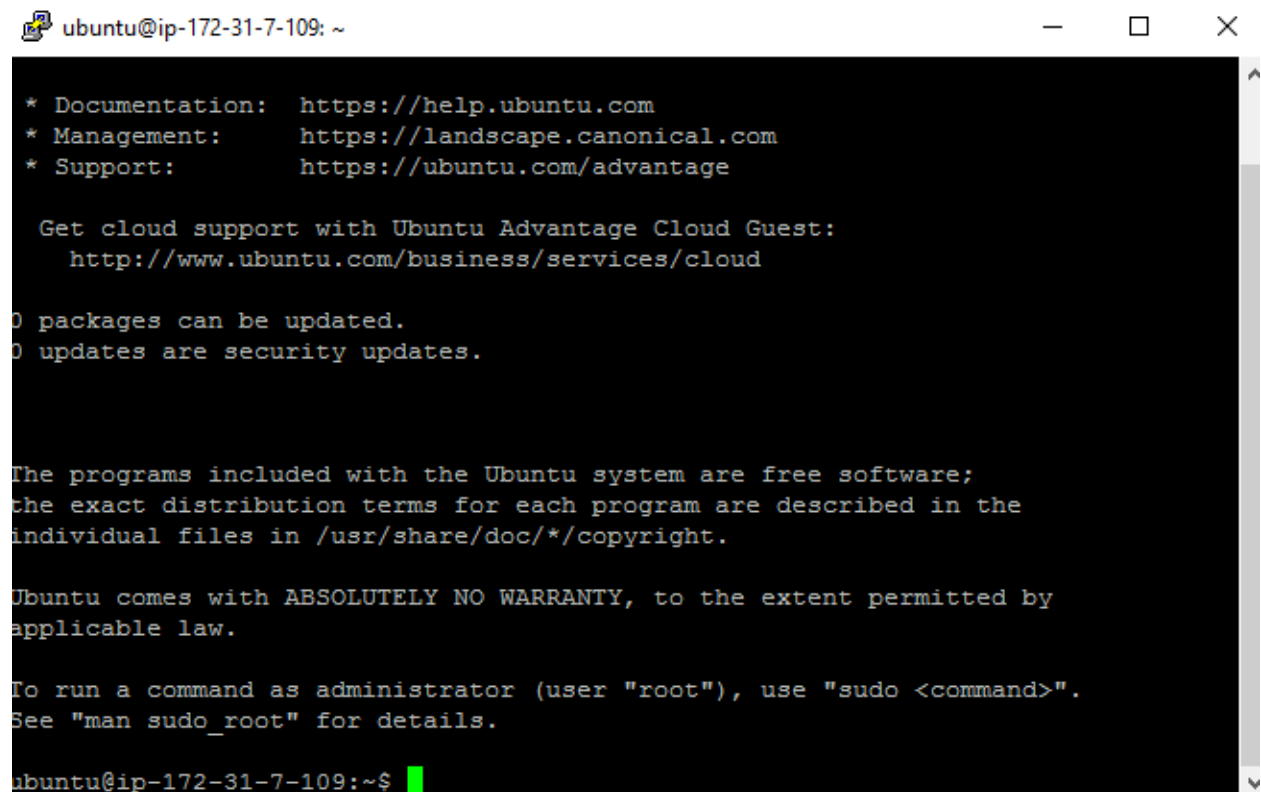


5- If you plan to start this session again later, you can save the session information for future use. Select **Session** in the Category tree, enter a name for the session in Saved Sessions, and then click **Save**.

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.

A screenshot of a terminal window titled 'ubuntu@ip-172-31-7-109: ~'. The terminal displays the following text:

```
* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-7-109:~$
```

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

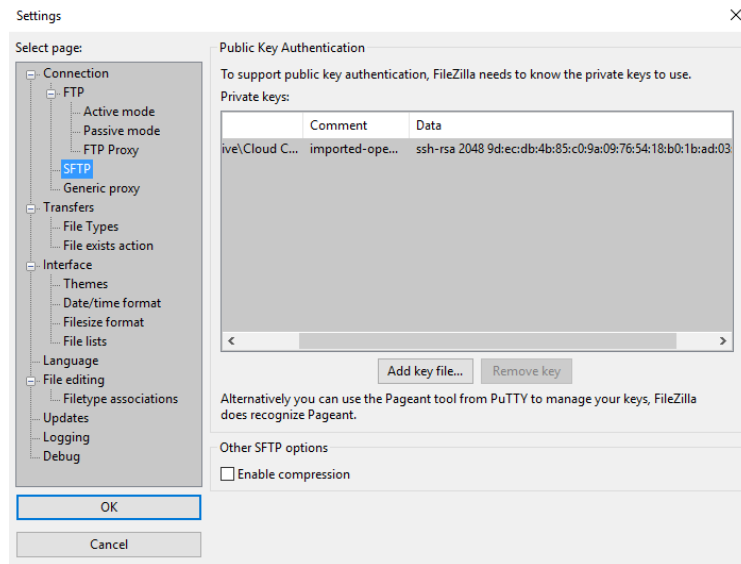
```
export JAVA_HOME=/usr/bin/java  
export PATH=$JAVA_HOME/bin:$PATH
```

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.

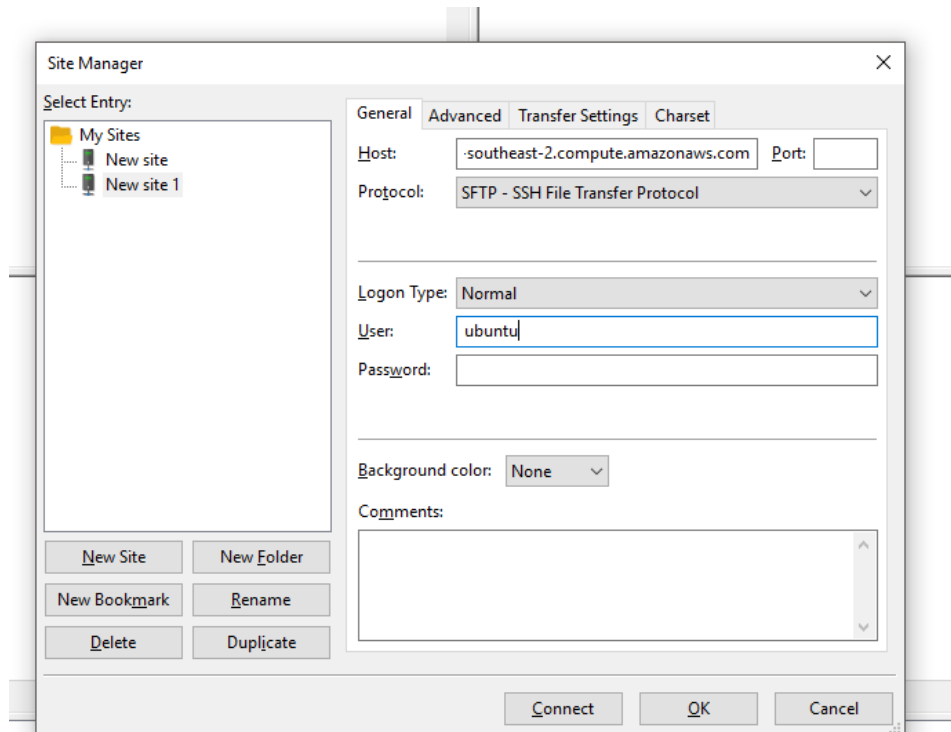
2- In Filezilla click on Edit>Settings. Then Select SFTP under connection. Click on Add Key file and browse to your .pem file. The key will be added. Then Click OK.



3- Click Ok.

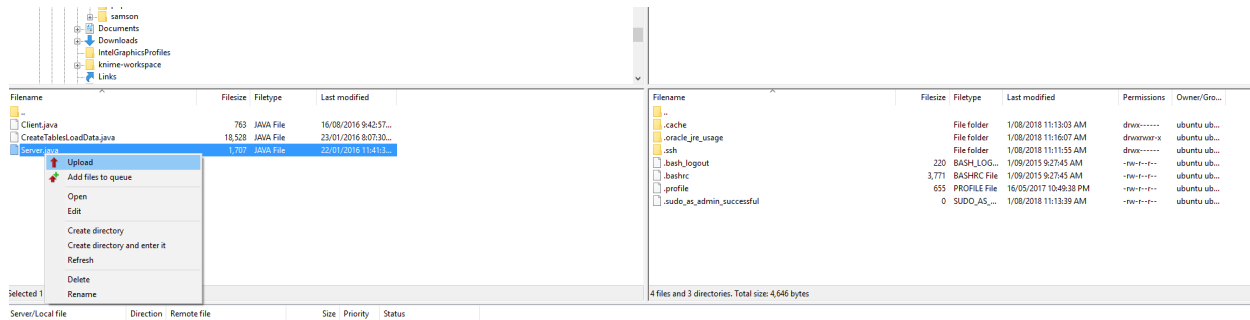
4- Now click File>Site Manager>New Site

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.



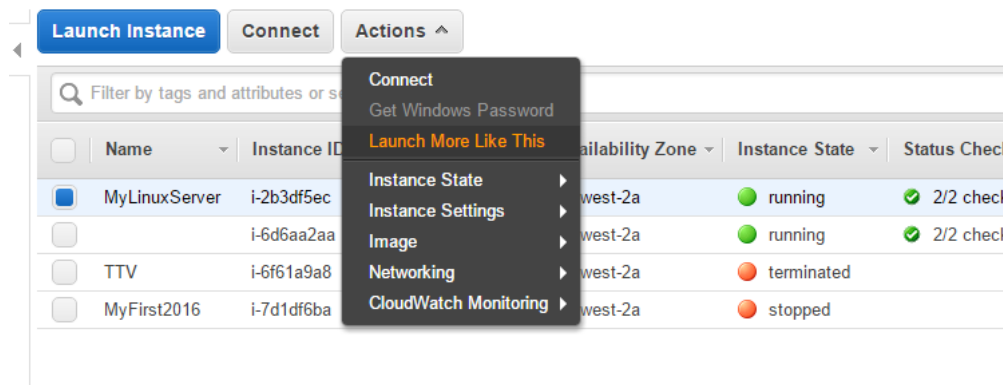
You can now able to see home directory of Ubuntu user.

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 filezilla 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**

Public DNS	ec2-52-10-212-156.us-west-2.compute.amazonaws.com
Public IP	52.10.212.156
Elastic IP	-
Availability zone	us-west-2a
Security groups	launch-wizard-1. <a href="#">view rules</a>
Scheduled events	No scheduled events
AMI ID	amzn-ami-hvm-2014.09.2.x86_64-eks (ami-dfc39aef)
Platform	-
IAM role	-
Key pair name	alsharif_key
Owner	418475102205
Launch time	February 24, 2015 1:18:49 PM UTC+11 (less than one hour)

5- You should see apache test page.



## Apache2 Ubuntu Default Page

### ubuntu

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

#### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.

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
```

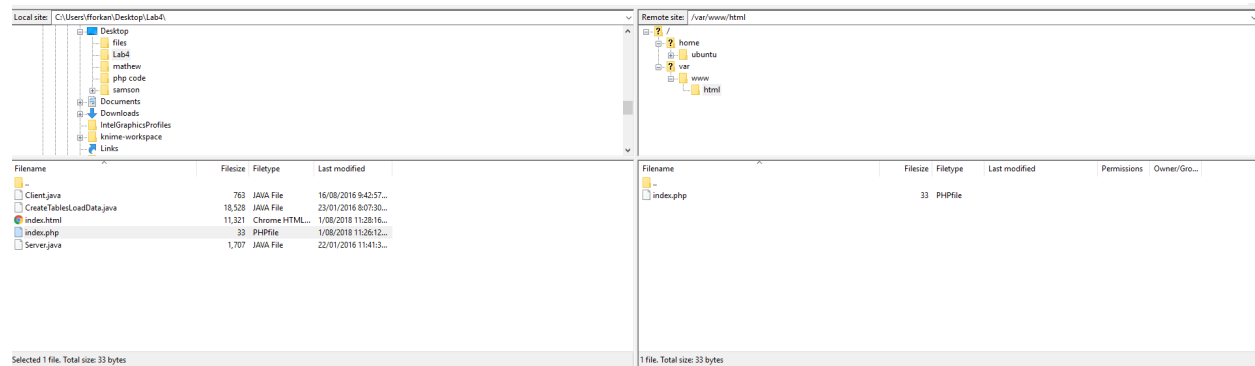
```
echo "Hello World!";
```

```
?>
```

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

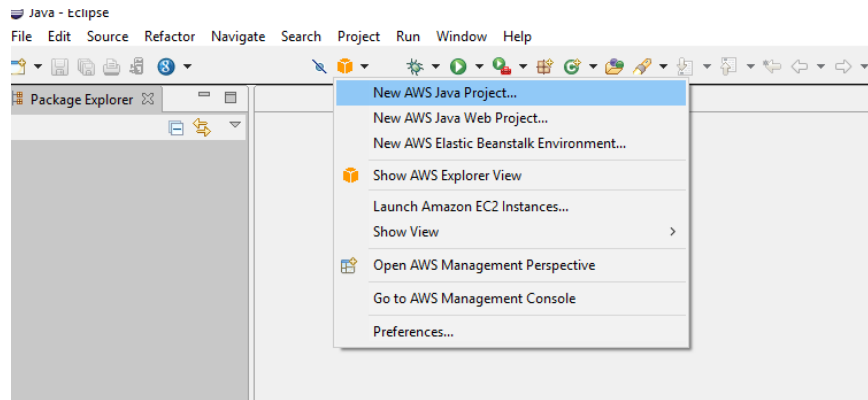
From AWS console dashboard "Services"> "Storage" select S3.

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

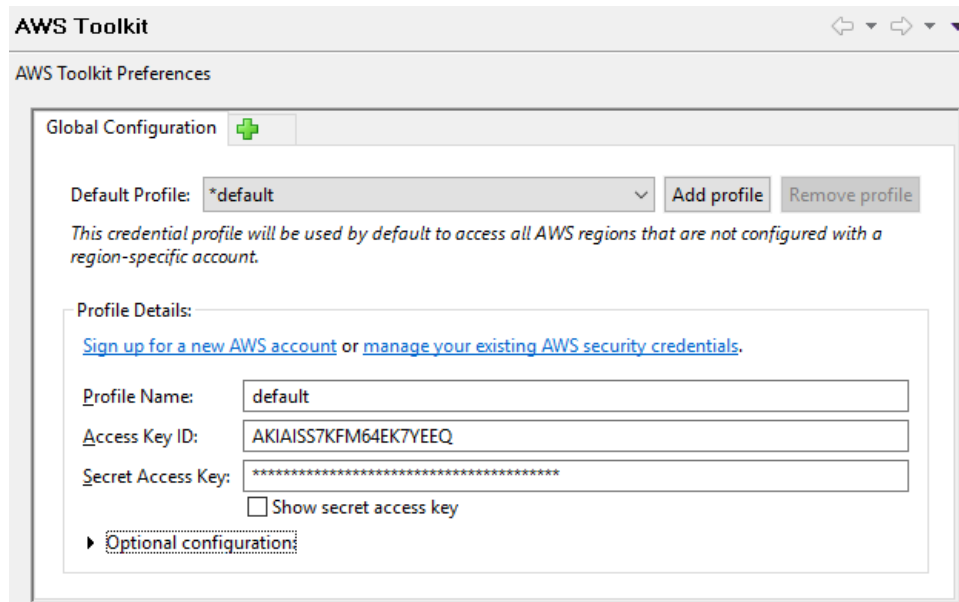
2- From Amazon icon select **New AWS Java project** as in the screen



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



Vocareum

My Classes

Help

s3672118@student

Welcome

How to use your AWS Educate Starter account:

- Check on credits, term, and session time remaining in your account in the Starter Account Status area.
- Click the Account Details button to see your AWS credentials, a list of any running services, and credit remaining in your account.
- Click the AWS Console button to login and start using AWS services like compute (EC2), storage (S3), and more.
- List of AWS resources available under your AWS Starter Account

Build your first cloud projects on AWS with our 10-minute tutorials!

Launch a Linux Virtual Machine with Amazon EC2

Store and Retrieve a File using Amazon S3

Send an Email with Amazon SES

Don't forget to logout once you are done with your work!

Your Starter Account Status

Active

full access

\$100

credits (estimated)

363d 20:56

remaining term

2:57

session time

Account Details

AWS Console

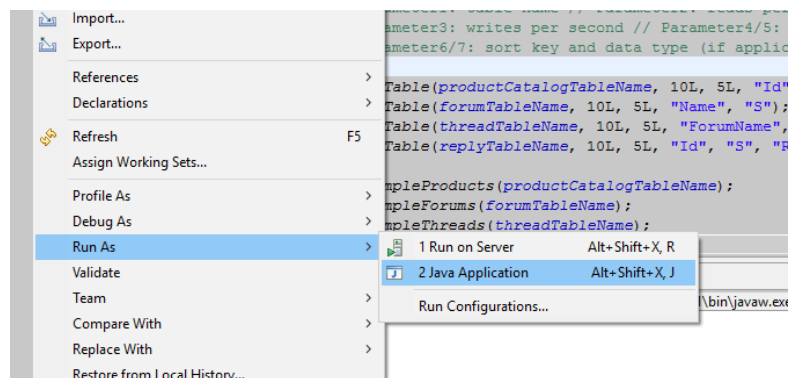
- In the popup menu select **Access Keys / Secret Keys**.

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

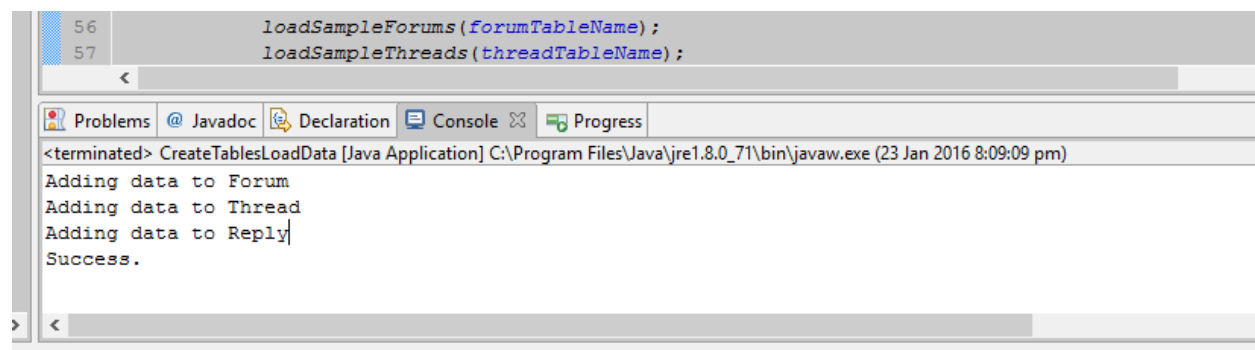
7- Right click on the package name and create a new class named **CreateTablesLoadData**

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