AWS Lab Exercise

# Lab 1 – Setting up AWS Account

## *Objectives*

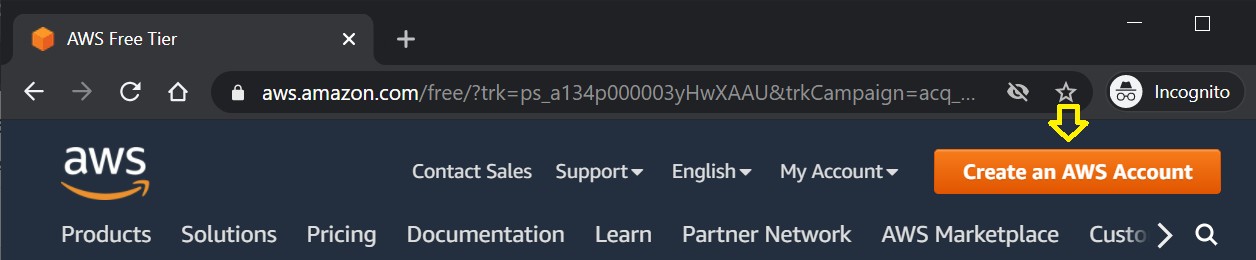
In this exercise, you will register for an AWS account with AWS Educate Account

## *Introduction*

Most system in current environment require to be “Always On” and “Accessible Anywhere”. This would mean that your project is not able to be run from your computer. Your production application for this project will deployed in AWS and accessible using Public IP address and with Firewall rules configured.

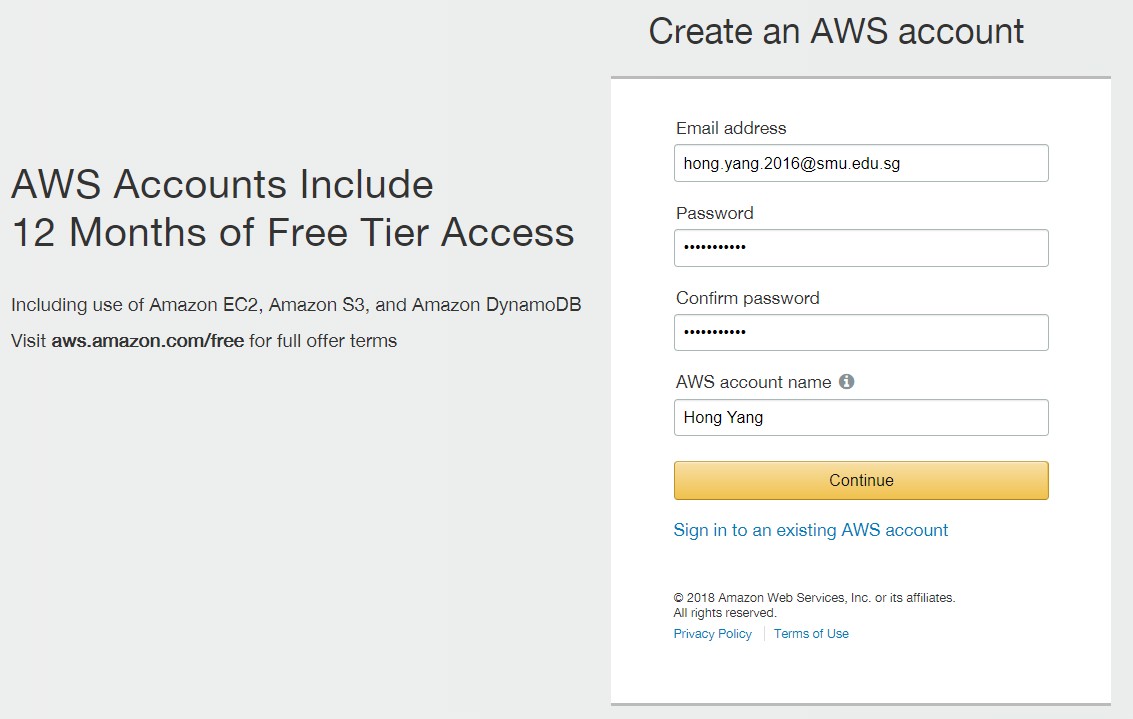
## *Create an AWS account using SMU email*

1. Go to: [https://aws.amazon.com](https://aws.amazon.com/) > Click “Create an AWS Account”

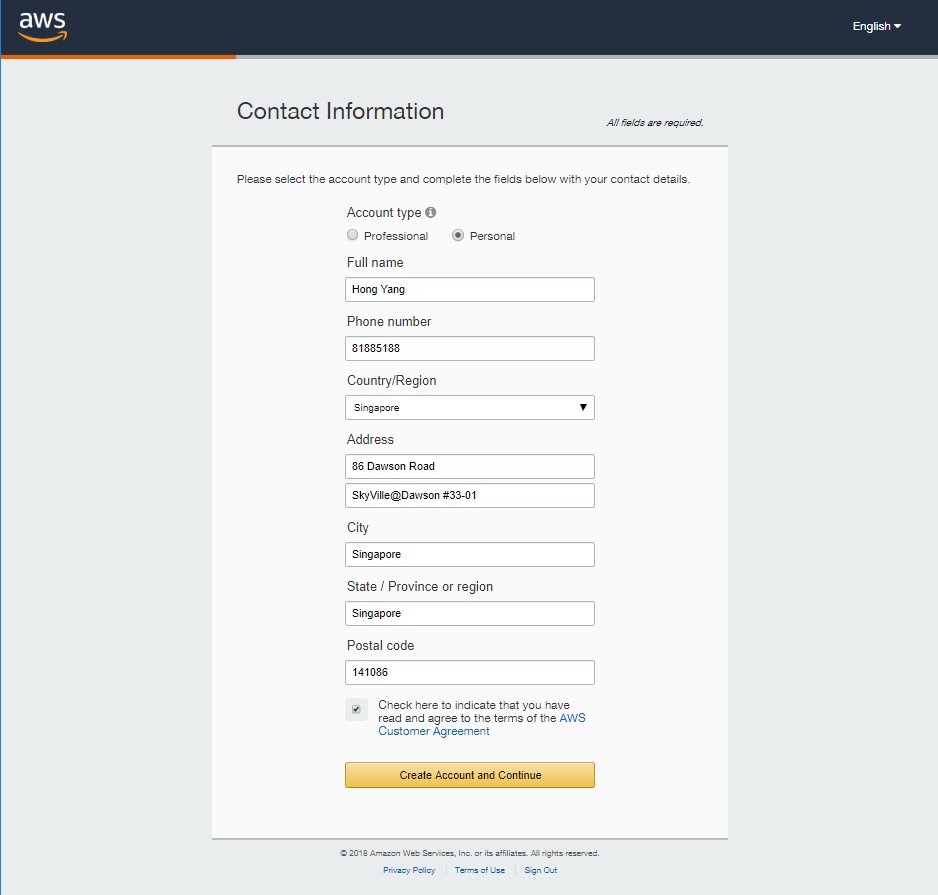


1. Enter your **smu email**: <your-id>@smu.edu.sg, password and account name. (Don’t worry too much

about the account name, it can be change again later)

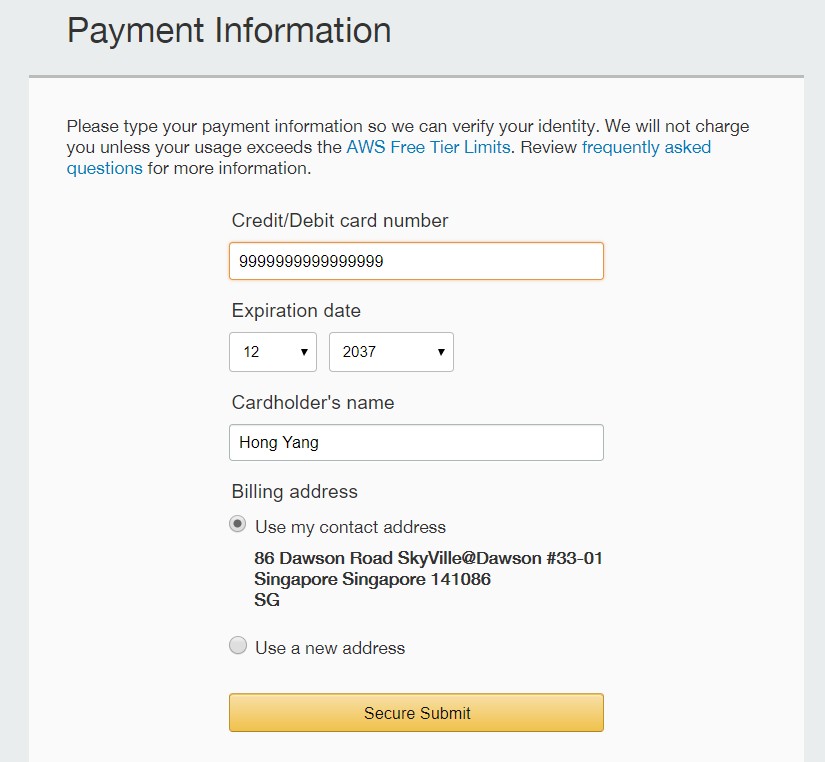


1. Select **personal** for account type and fill up login credentials



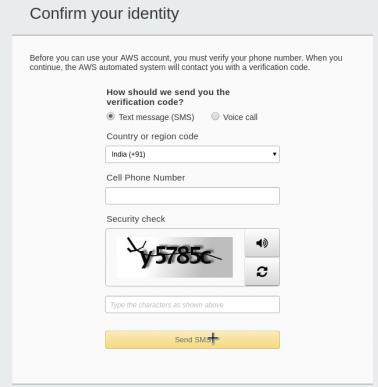
1. Fill up payment information (Don’t worry you will be receiving free credits and will not be billed a

single cent for this course)

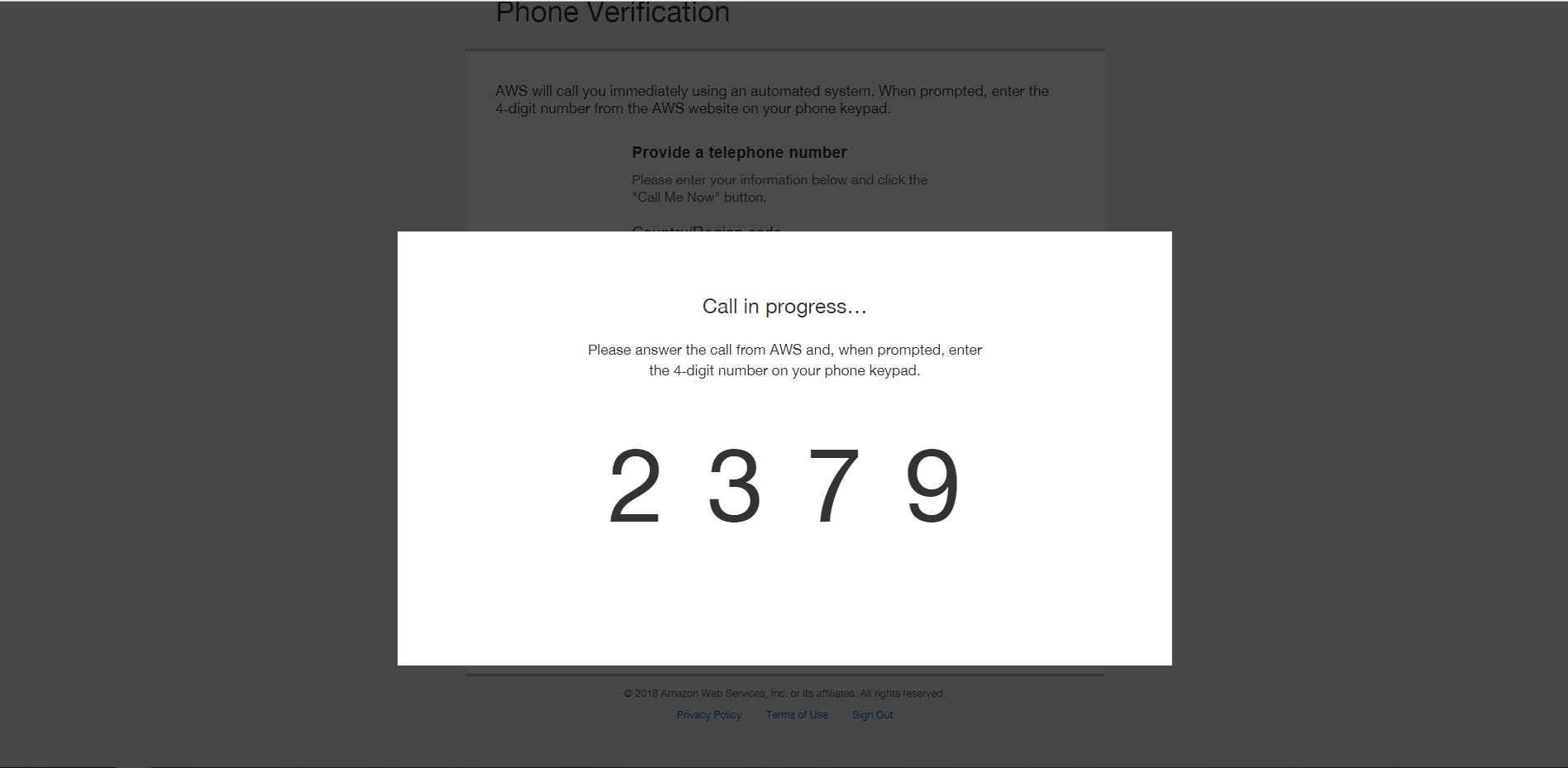


1. If your number has already been filled in, double check and make sure it is correct. You can leave

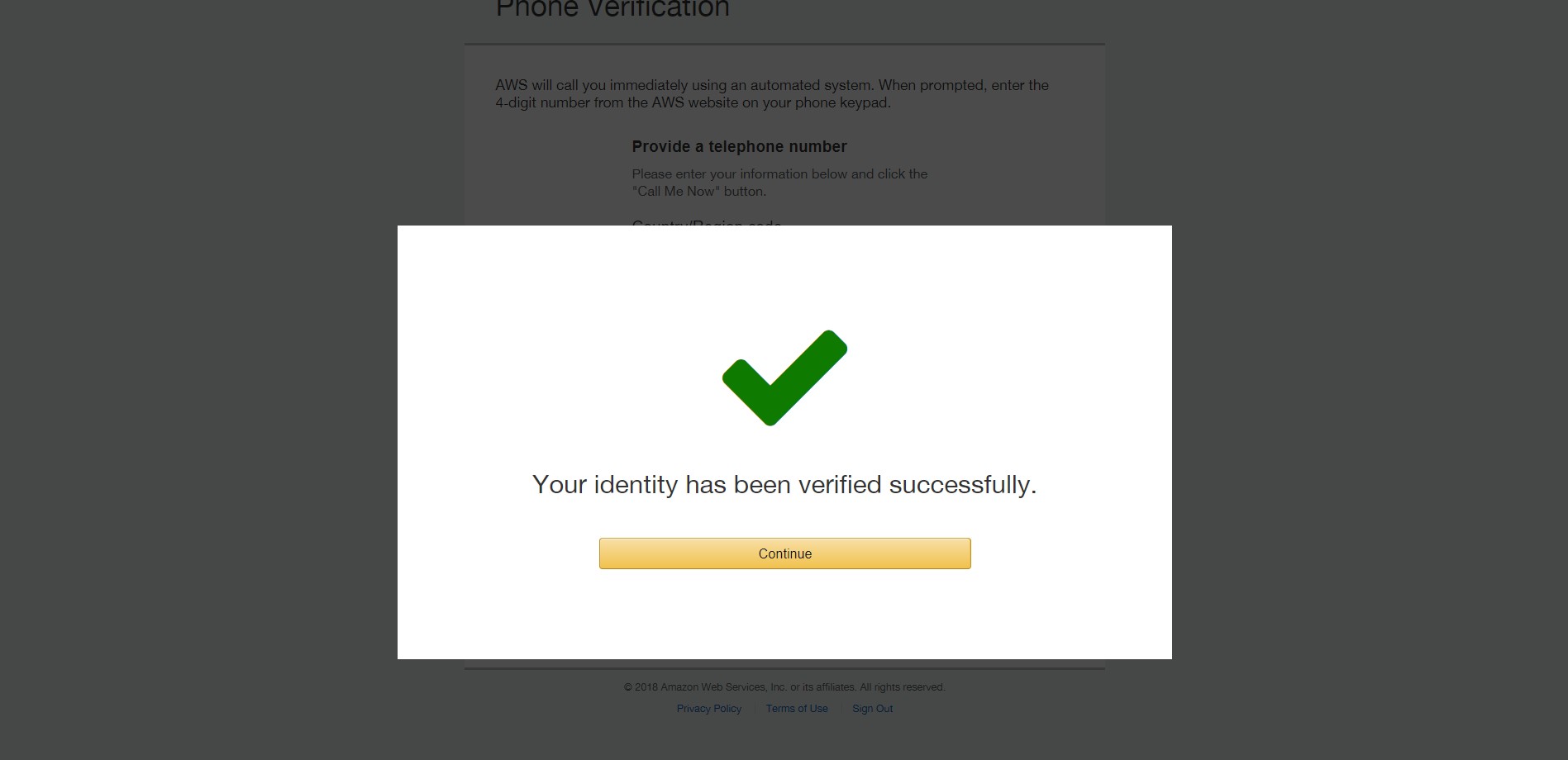
the “Ext” field empty. If not, please fill in your mobile number. Once done, click “Call me now “.



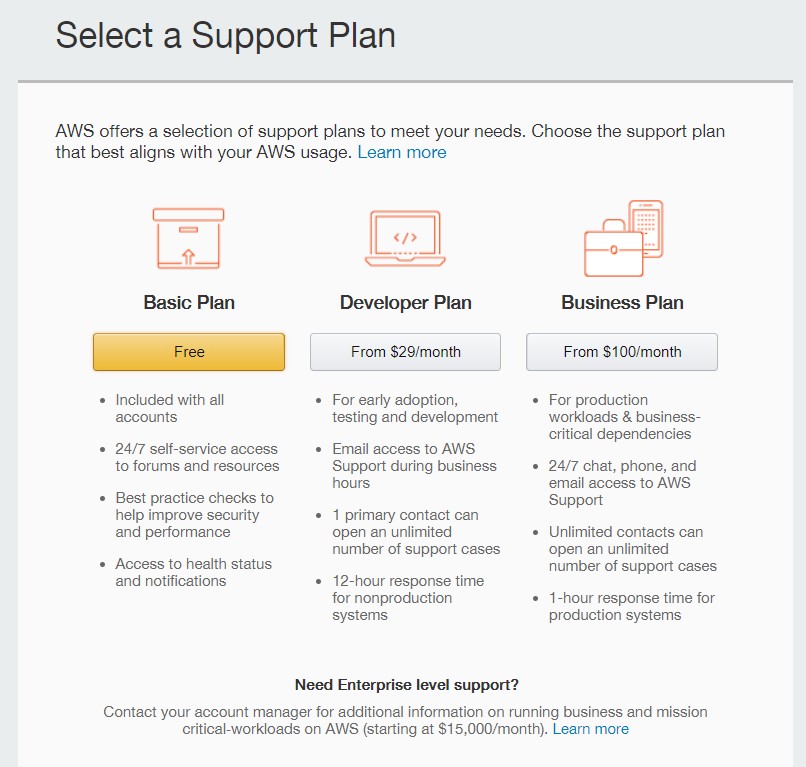
1. Wait patiently for a call. You will hear an instruction to enter your code. Enter the code once prompted. Give it a short wait while it verifies the pin you enter on your phone.



1. Once verification is completed, click on “Continue”



1. Select “Free”.



1. Once you see this page, your account registration has been completed.

\*\*Please Stop your Instance when not IN USE to avoid additional cost”. You can also opt

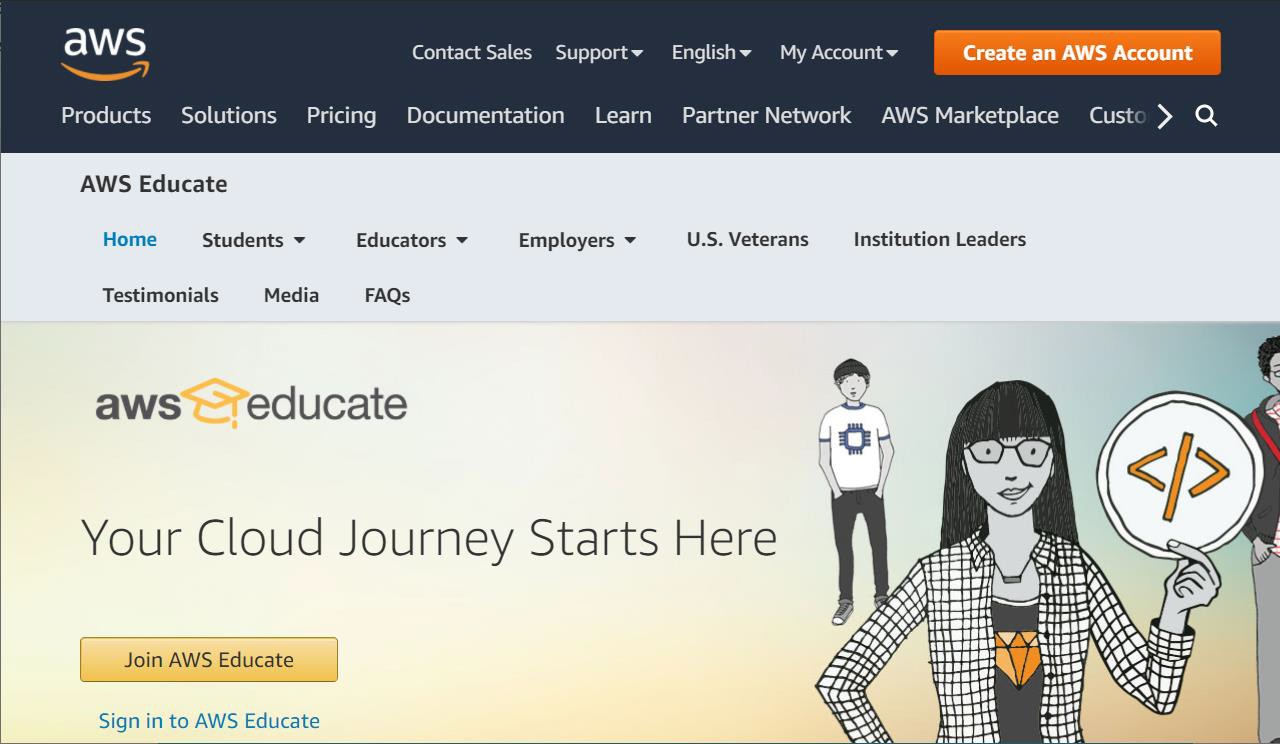
for the Free Tier options”

## *(Optional) Register for an AWS Educate account*

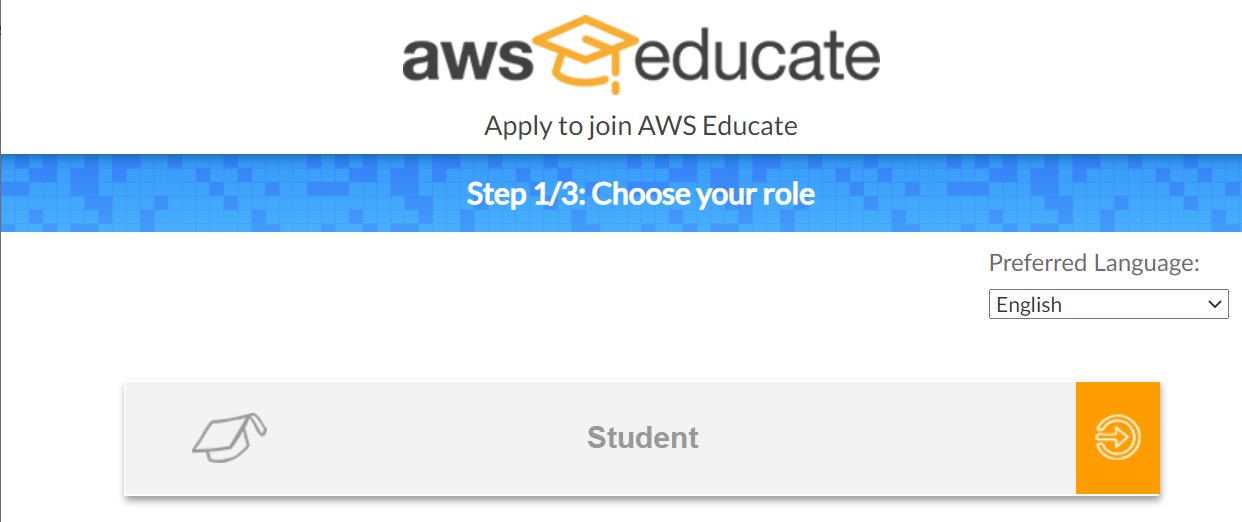
On top of the free services that AWS provides, you can get free AWS credits when you sign up with AWS Educate as a student of SMU. [https://aws.amazon.com/education/awseducate/aws-educate-faqs](https://aws.amazon.com/education/awseducate/aws-educate-faqs/)/

If you intend to use AWS for your projects in the coming 12 months, complete this section to get the free credits. Otherwise, you can skip this section and can sign up with AWS Educate only when you need to use AWS.

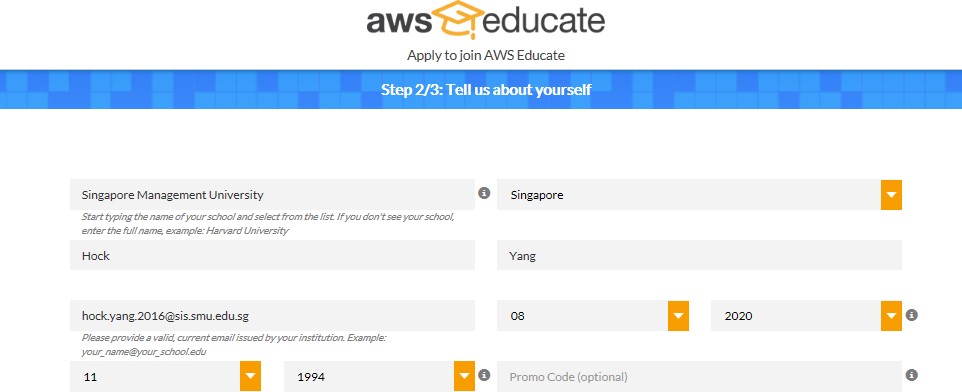
1. Go to: <http://aws.amazon.com/education/awseducate/>
2. Click “Join AWS Educate”



1. Click Student

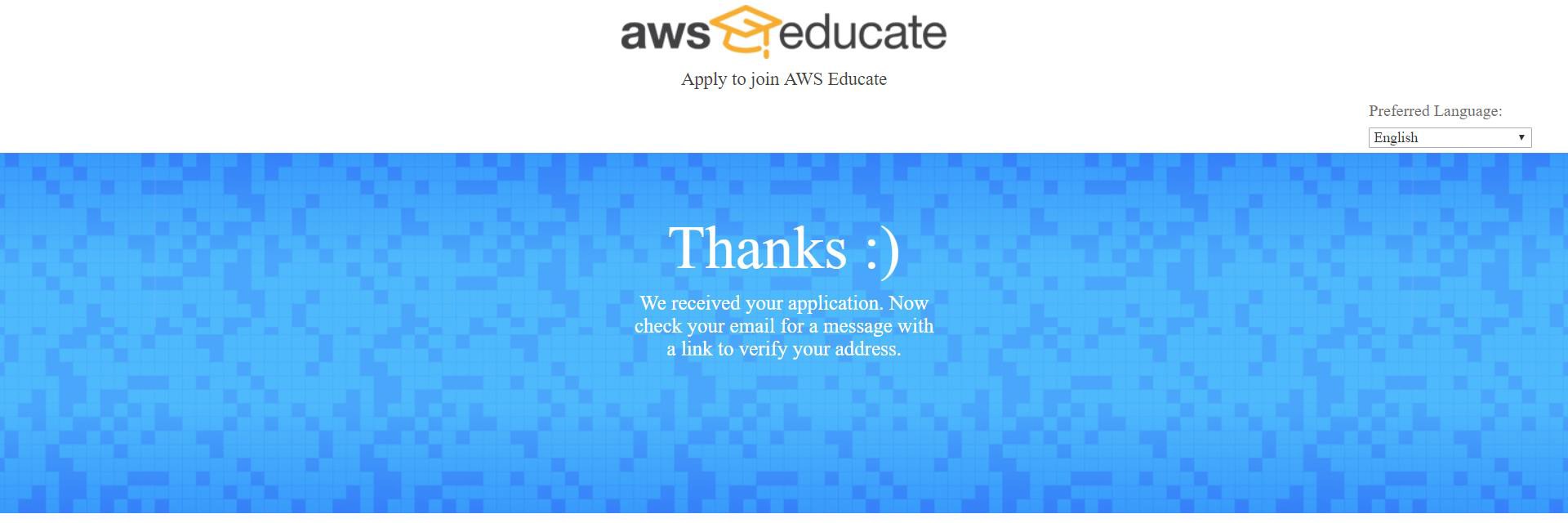


1. Fill out the application form as follows.

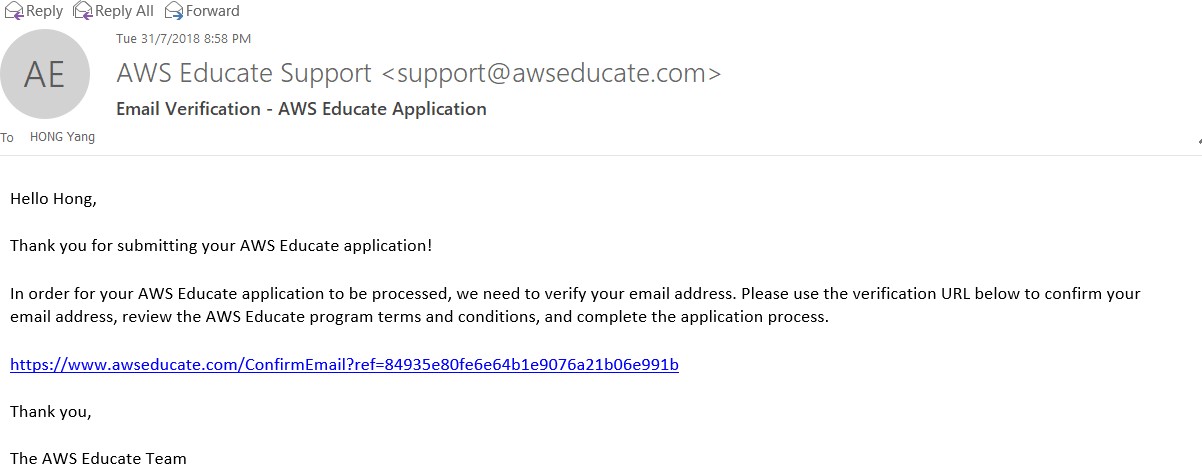


1. Accept the Terms and Conditions.

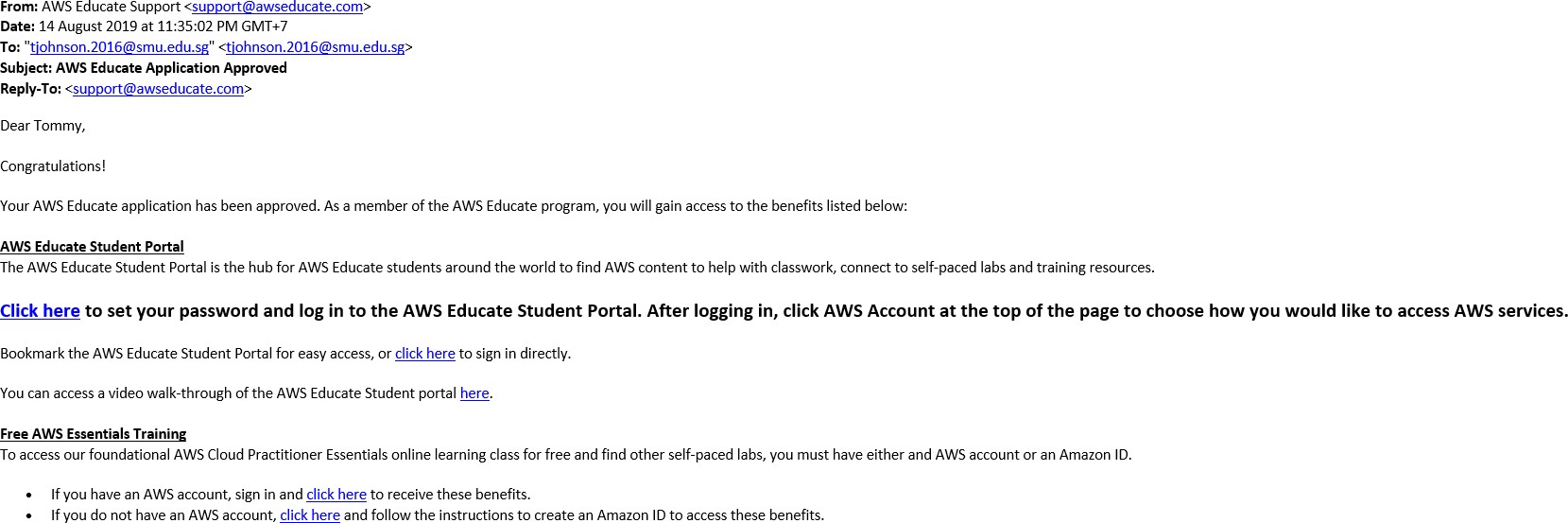




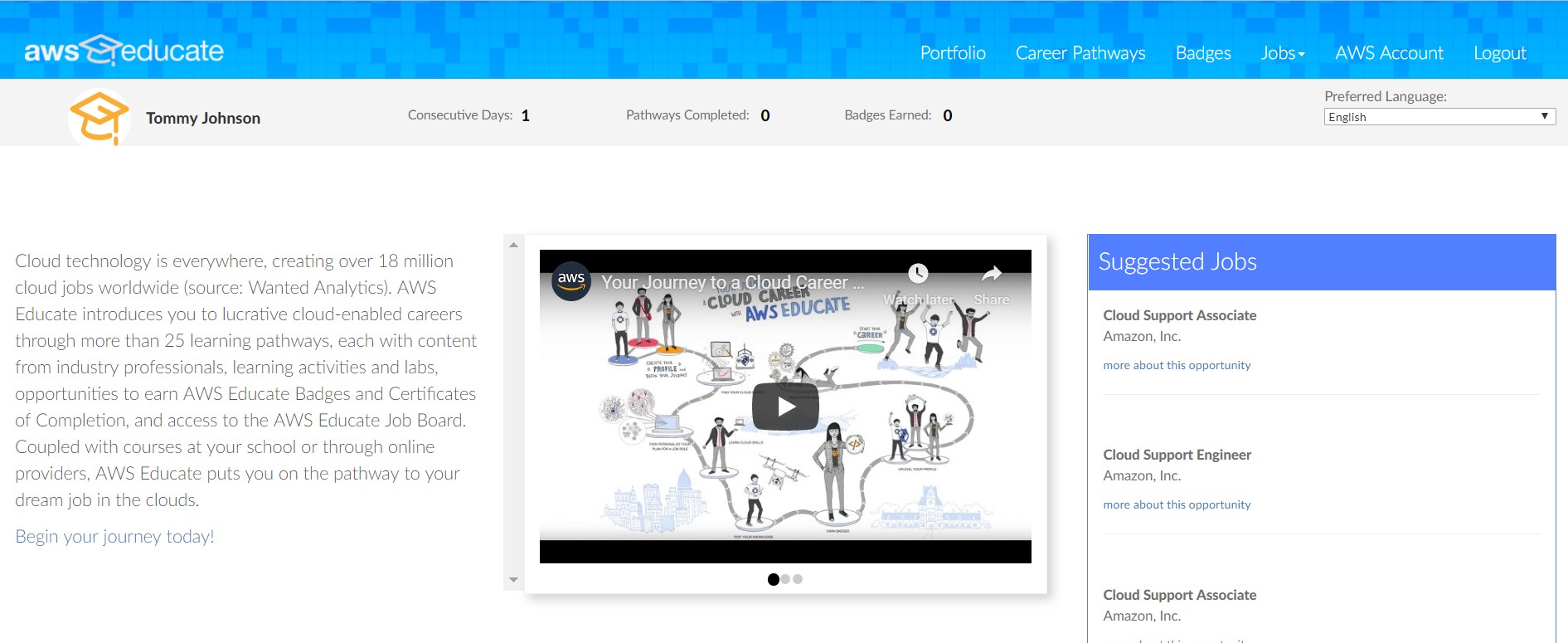
1. Go to your SMU email account and click on the link



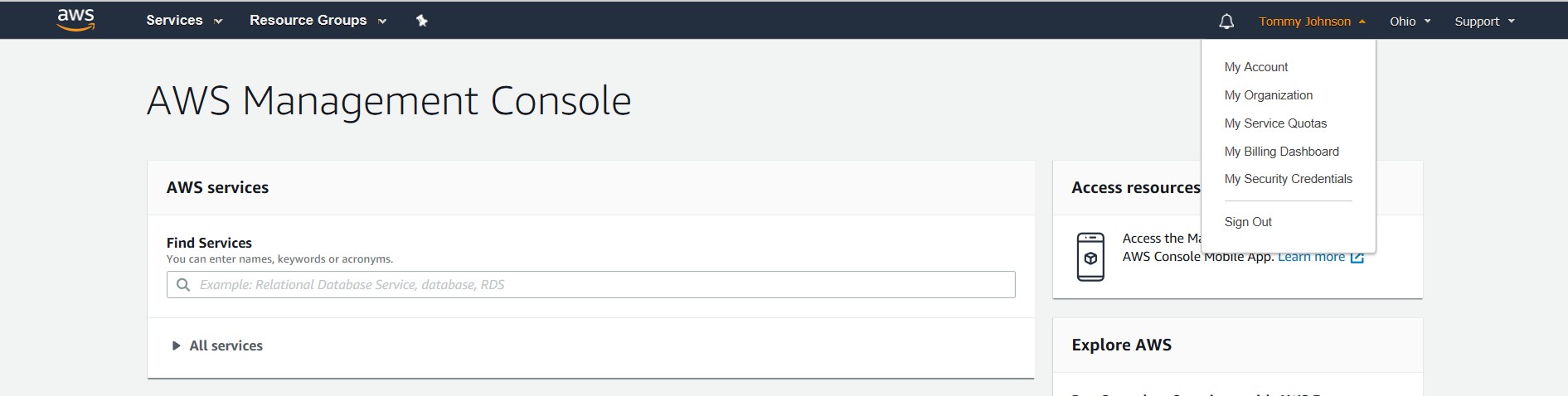
1. When your application is approved, you will receive an email notification as follows. Click on the link to set your password and log in to the AWS Educate Student Portal.

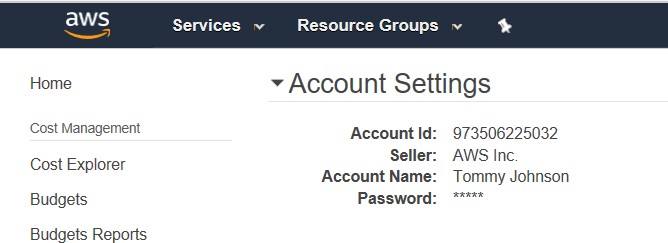


1. After you are logged in, click on “AWS Account” and choose “personal AWS Account ID”.



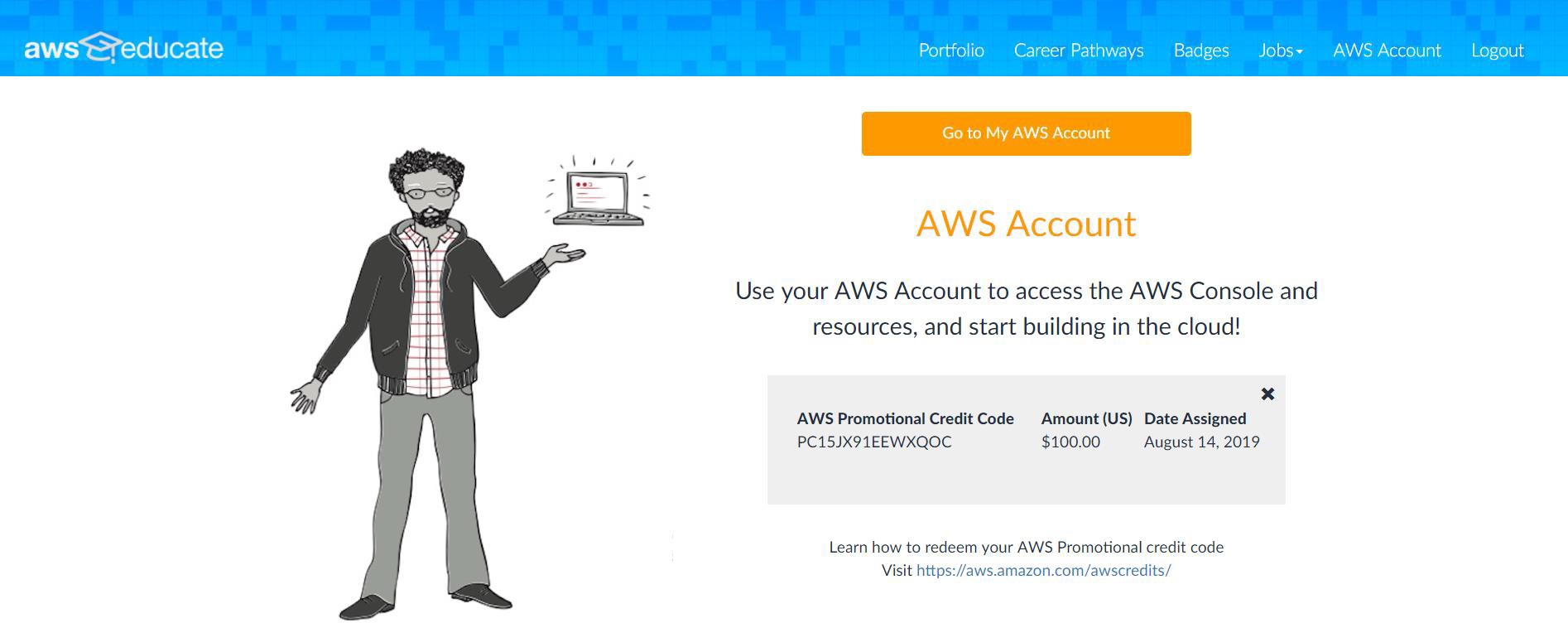
1. Log in to your AWS Account and click on “My Account” under your profile.



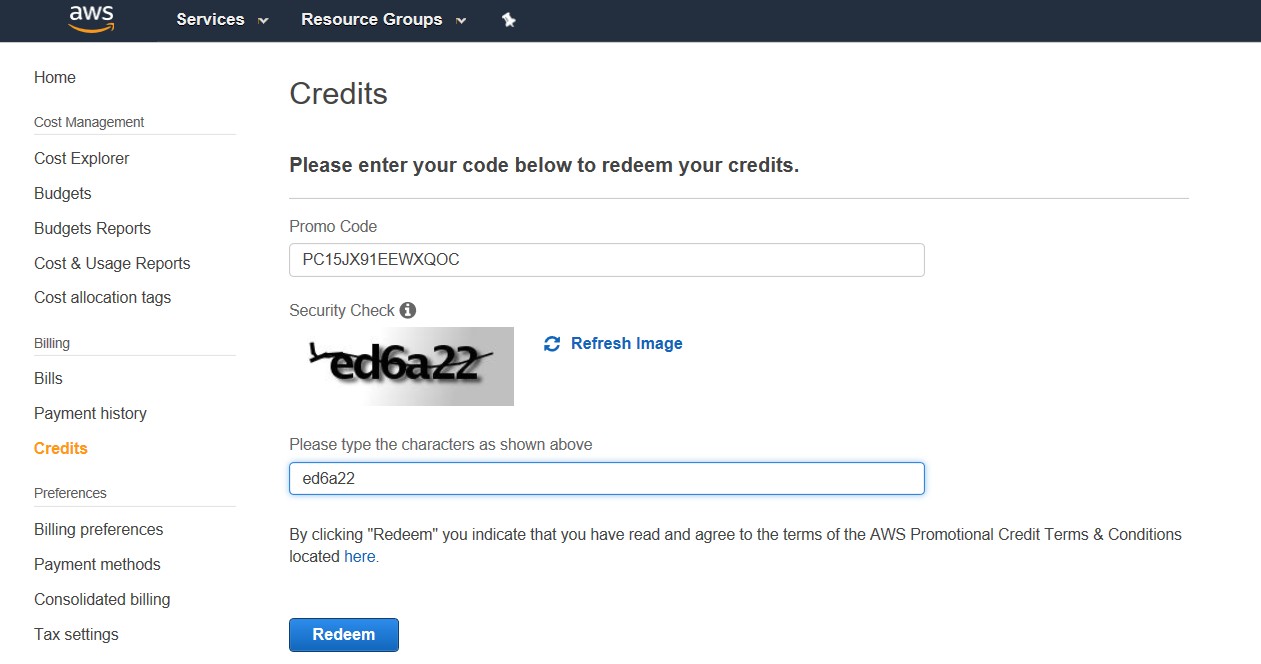


1. Copy your “Account Id” onto the AWS educate portal to get a AWS Promotional Credit Code.

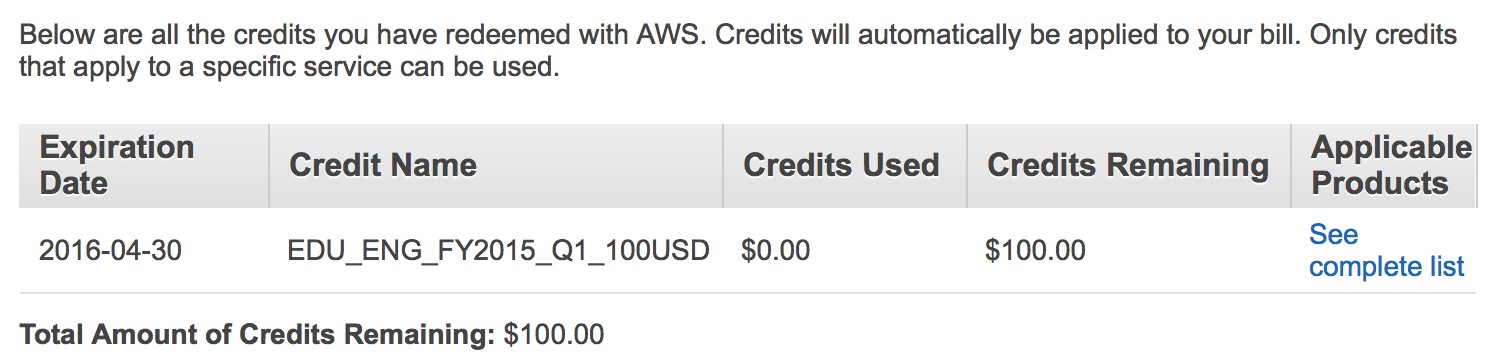
Copy this Code.



1. In your AWS account, navigate to “Credits” and paste the Code to redeem credits as follows.



1. And you’re done! Enjoy your credits!



# Lab 2 – Creating an AWS EC 2 Instance

## *Introduction*

As mentioned earlier, in real projects, the final products will be deployed in a production environment.

**Production environment**1 is where we will deploy to once our web application passes user testing. It is another computer your customer will use for real.

* It will have its own copies of the application server (Apache), and database (MySQL).
* It has a **public IP Address (X.X.X.X)** that allows our users to access the application via the Internet at any time and any place.

Below is a diagram for illustration.

|  |  |  |
| --- | --- | --- |
| **Development Environment** |  | **Production Environment** |
| Development Environment | Deploy to | Production Environment |
| E.g. Your laptop   * Apache 8 * MySQL * phpMyAdmin |  | **EC2 Instance**: ***X.X.X.X***   * Apache 8 * MySQL * phpMyAdmin |

In this lab exercise, We will create a **virtual machine** on the cloud with a public I.P. address (called an instance on Elastic Compute Cloud or EC2).

1 For real life projects, depending on the project’s requirements andbudget, we may have

1. Development environment – where developers do their coding,
2. QA environment – for developers or a separate QA team to test the integrated application,
3. Staging environment – closely mimics the production environment, andfor customers to conduct their acceptance testing before deployment to the production environment,
4. Production environment – the real environment where customers will use your web application.

## *Pre-Setup*

***Download resources***

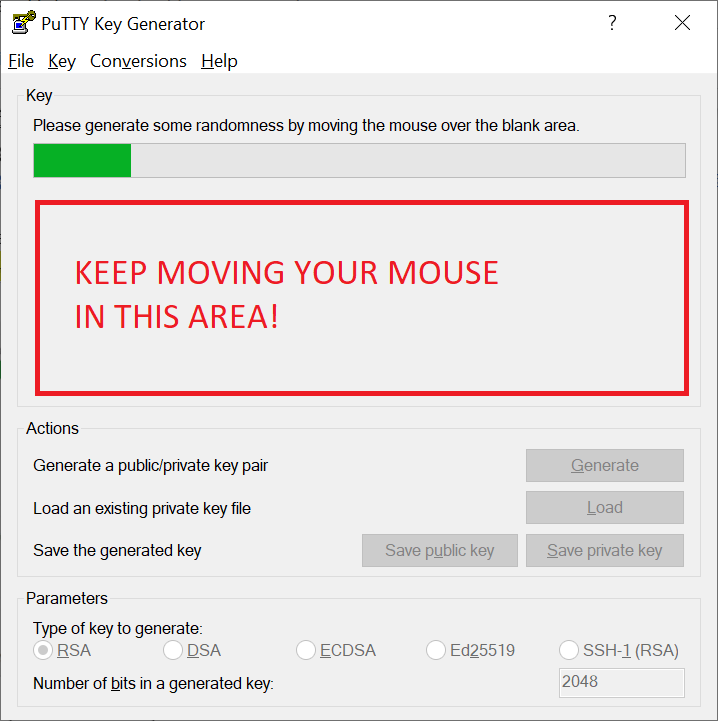
1. Download **puttygen.exe** and **WinSCP.zip** from this Google drive shared folder: <https://drive.google.com/drive/folders/1r0BythgEpI_4Wrb9zeueHunACfG2cuSe?usp=sharing>
2. Unzip the WinSCP.zip.

## *Generating a SSH key*

AWS requires us to use SSH keys2 for secure connections. Hence, we are creating a SSH key first.

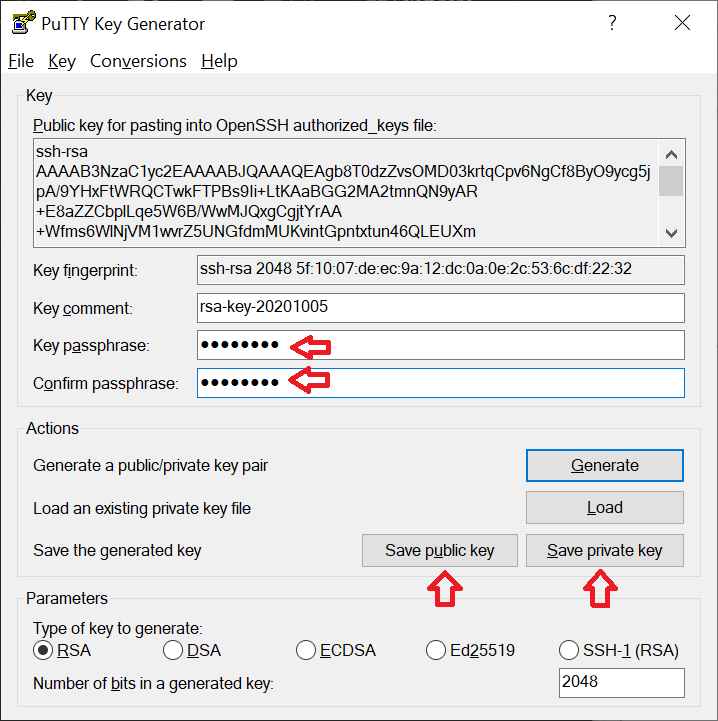
**1.** Run **Puttygen**. Select **Key** > **SSH-2 RSA**. Click **Generate.**

**KEEP MOVING YOUR MOUSE** over the blank area in the middle of the puttygen! If done correctly, the progress bar will increase.



2For more details on SSH keys: https://dev.to/risafj/ssh-key-authentication-for-absolute-beginners-in-plain-english- 2m3f

1. Once the key has been generated,
   1. Enter a Key passphrase and confirm passphrase
   2. Save the **public** key (extension .pub), e.g. **spm.pub**
   3. Save the **private** key (extension .ppk) ), e.g. **spm.ppk**



### IMPORTANT:

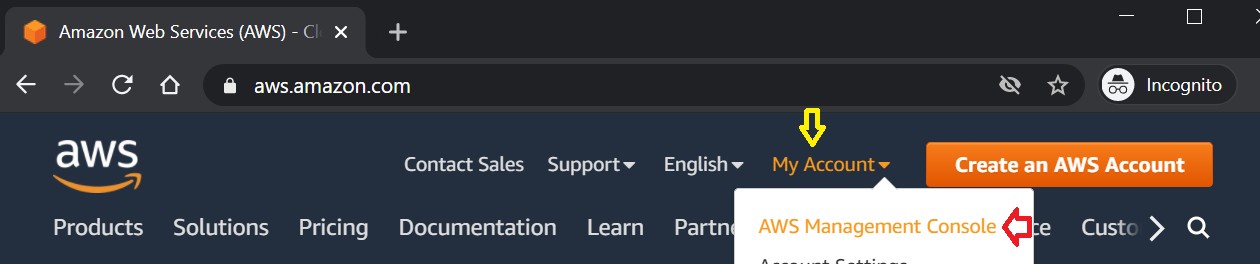
* Remember your passphrase.
* Remember where you save your **public** and **private** keys.

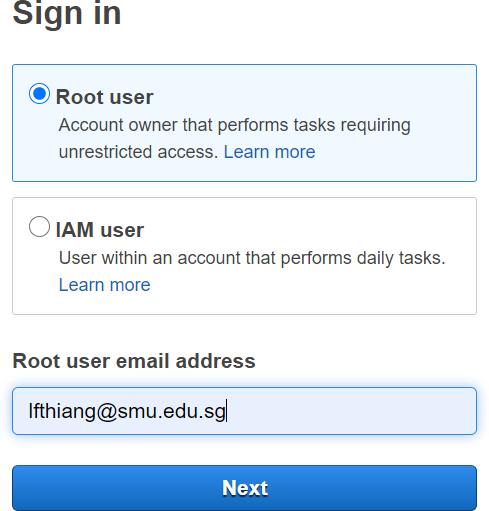
Note: If you work in a team, you can generate a separate key pair for the team and use it for all members to

access the team’s AWS virtual machine.

## *Logging into EC2*

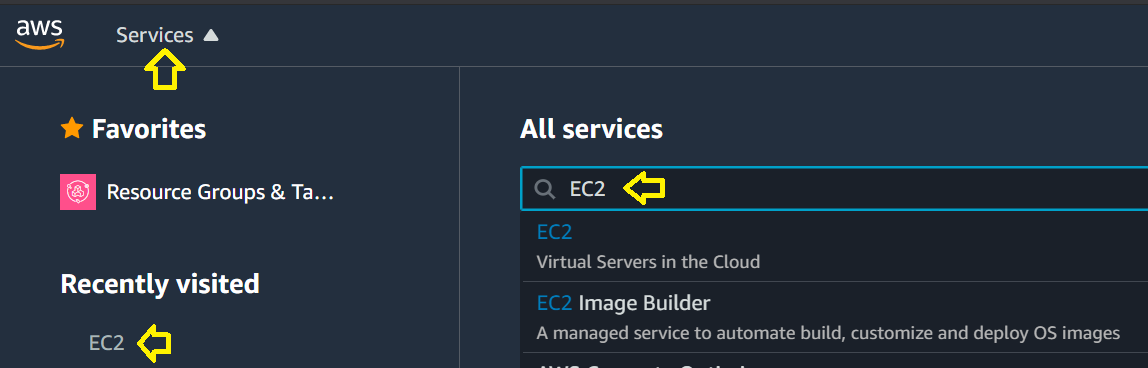
1. Go to: <https://aws.amazon.com/>>My Account > AWS Management Console To log into your AWS account.



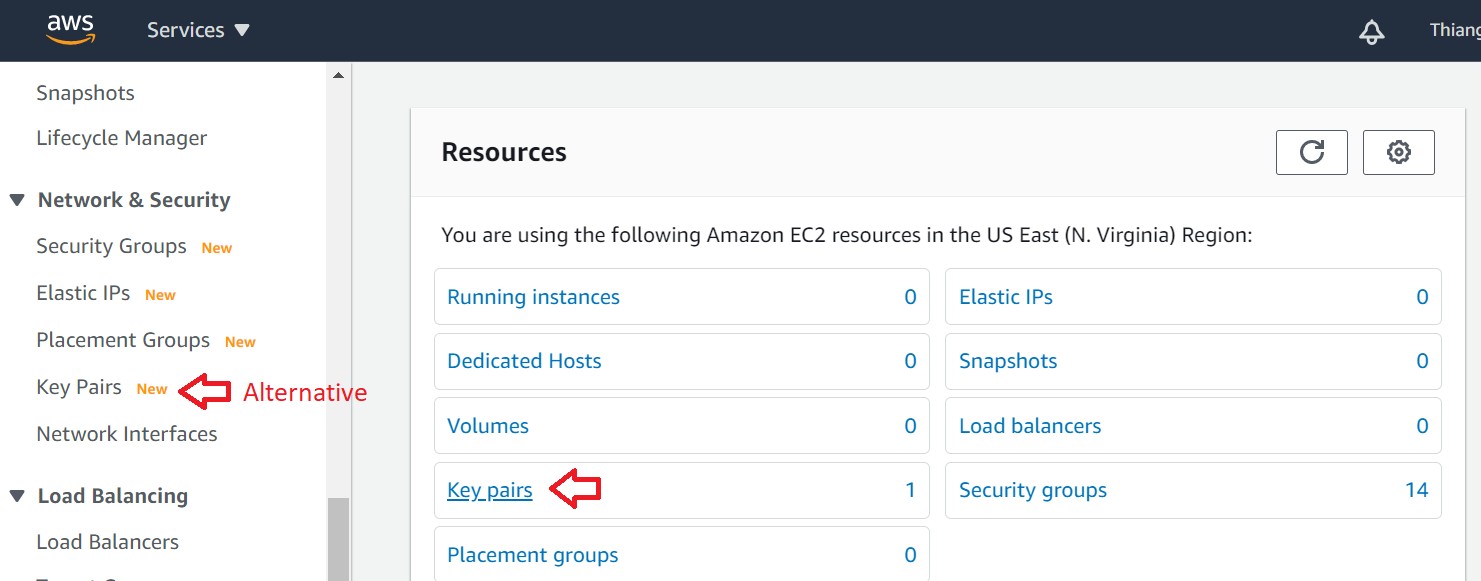
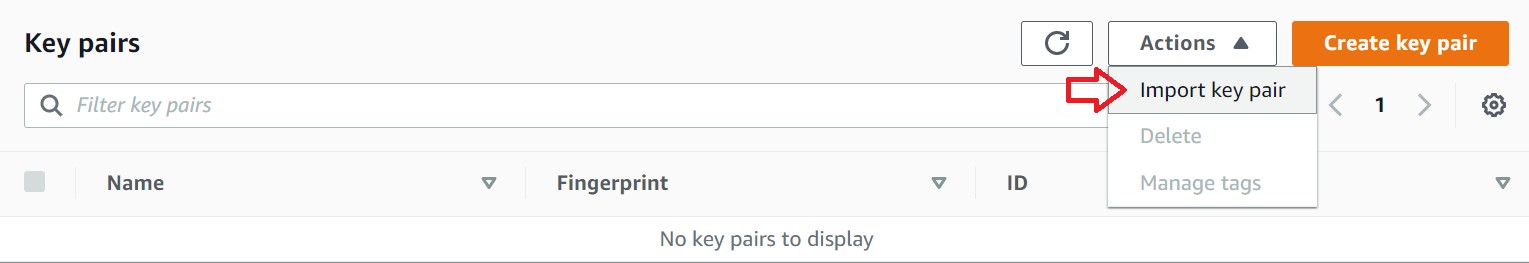


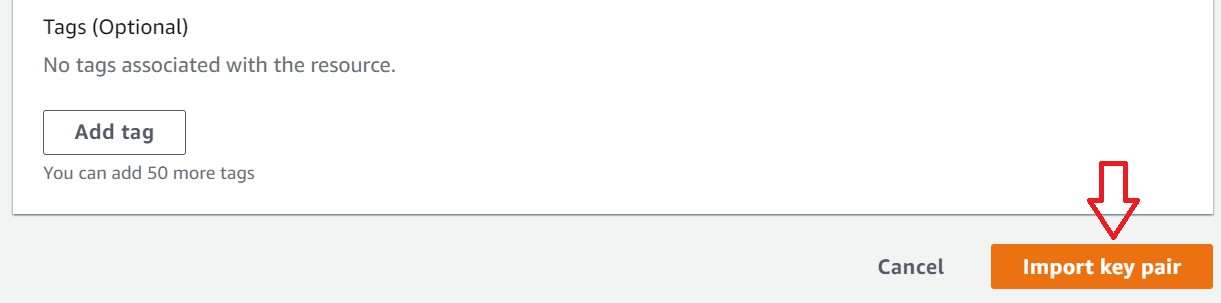
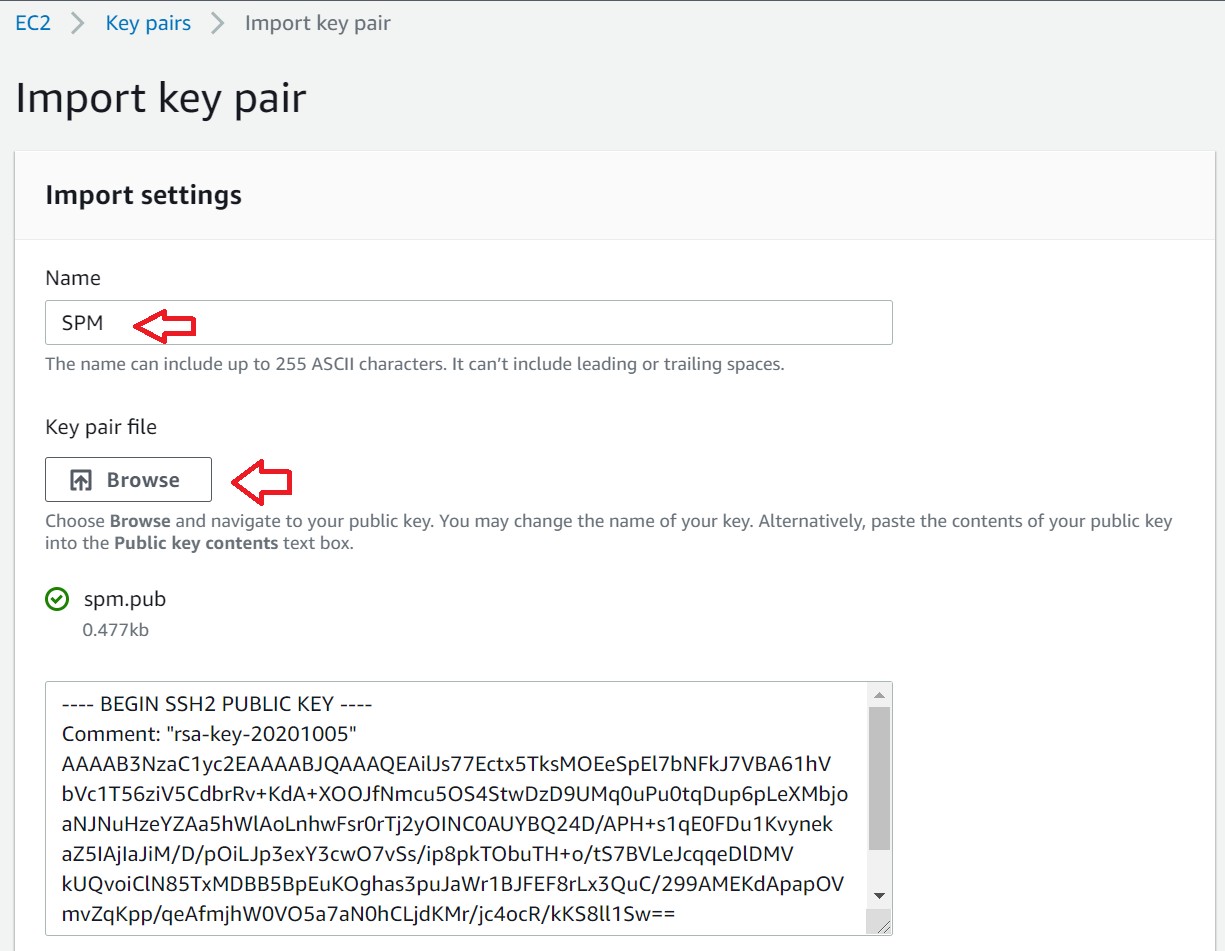
1. Top menu bar > click “Services” > Search and/or select “EC2” from the list shown.

We will be using EC2 to host our Apache server.

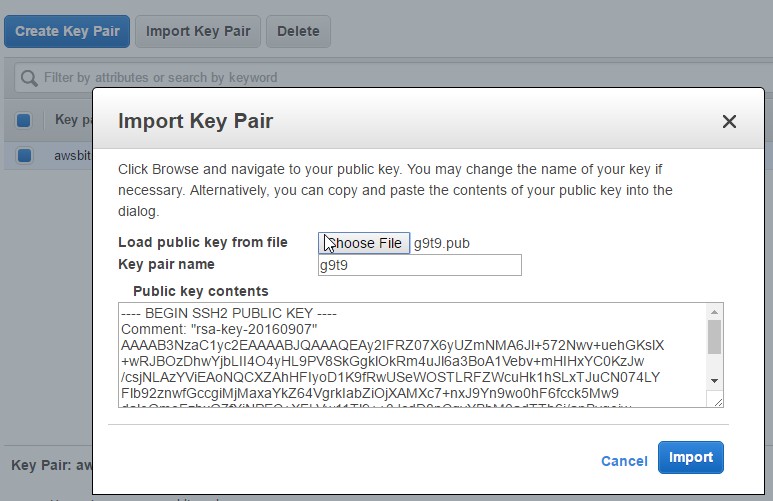


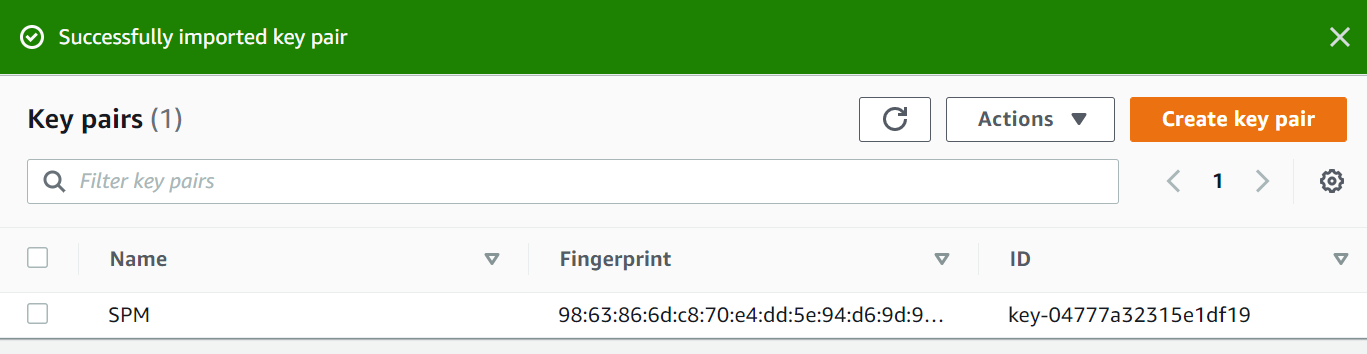
## *Set up SSH Keys*

1. After selecting EC2, you will see the EC2 Dashboard page. On the main window, click on “Key Pairs” to set up your SSH key.
   1. Alternatively, left menu > scroll down > Key Pairs
2. Drop down list “Actions” > Import key pair
3. The “Import Key Pair” dialog box will appear. Within the dialog box,
   1. Give an intuitive name; e.g. SPM
   2. Browse for your “Key pair file”, choose the public key file (e.g. spm.pub) that you have created and saved earlier for your team,
   3. Then click the “Import key pair” button.



If you have your private key file but not your public key file, launch PuttyGen. Load your private key file, then save the public key file.

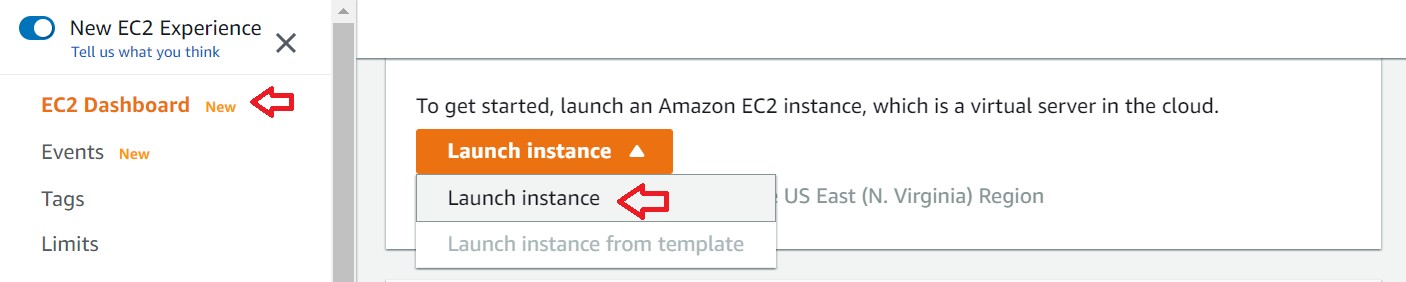


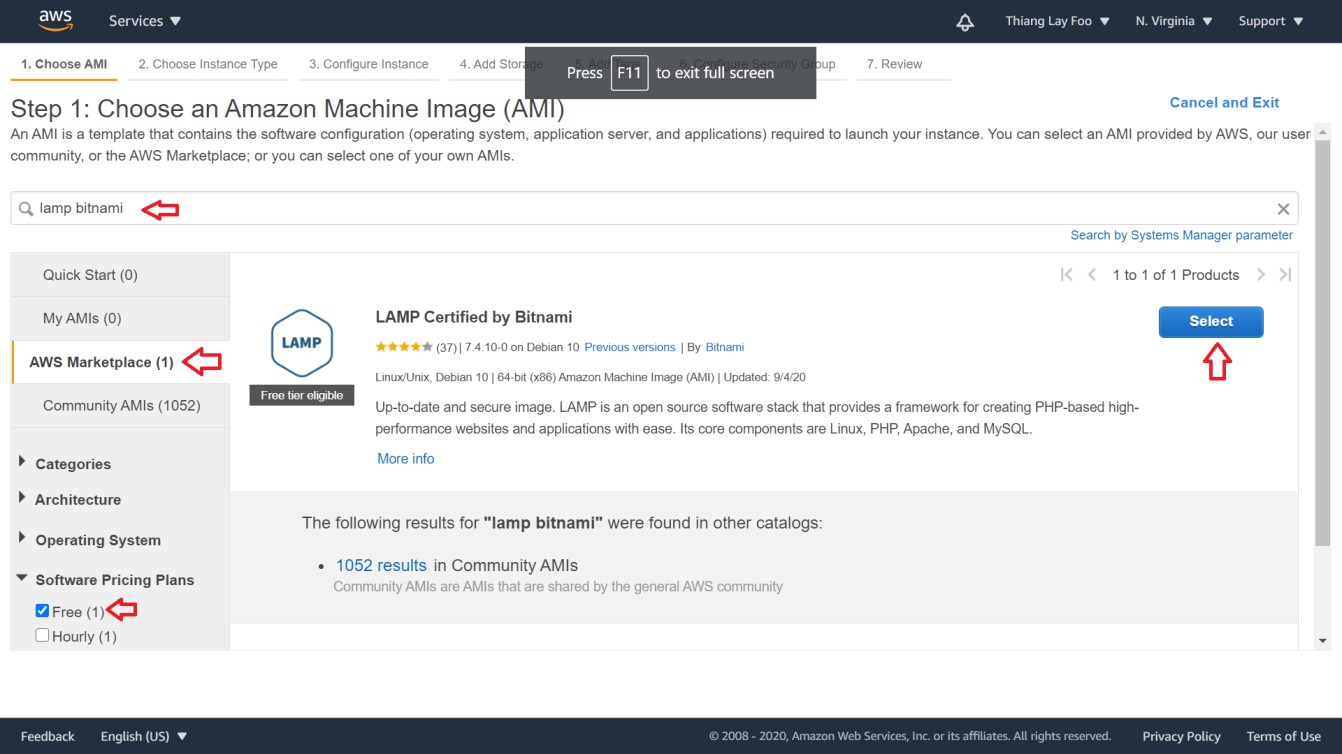
10. You will return to the Key Pair page and you should see your imported key pair listed.

Now, AWS has your public key. In the later part of this lab exercise, you will use your private key to connect to AWS securely.

## *Create an instance*

1. Left Menu > “EC2 Dashboard” to go back to EC2 Dashboard page
2. On the EC2 Dashboard page, scroll down and click the “Launch Instance” to choose an Amazon Machine **Image** (configuration of your instance, such as the operating system, application server, applications).



1. In the search box, type **‘lamp bitnami’** and press Enter. On the left menu
   1. Select the ‘**AWS Marketplace’**
   2. Scroll down to “Software Pricing Plans” > Check “Free”

You should see one result: “**LAMP Certified by Bitnami**”. This is a preconfigured image (or

operating system setup) containing Apache server, MySQL and phpmyadmin.

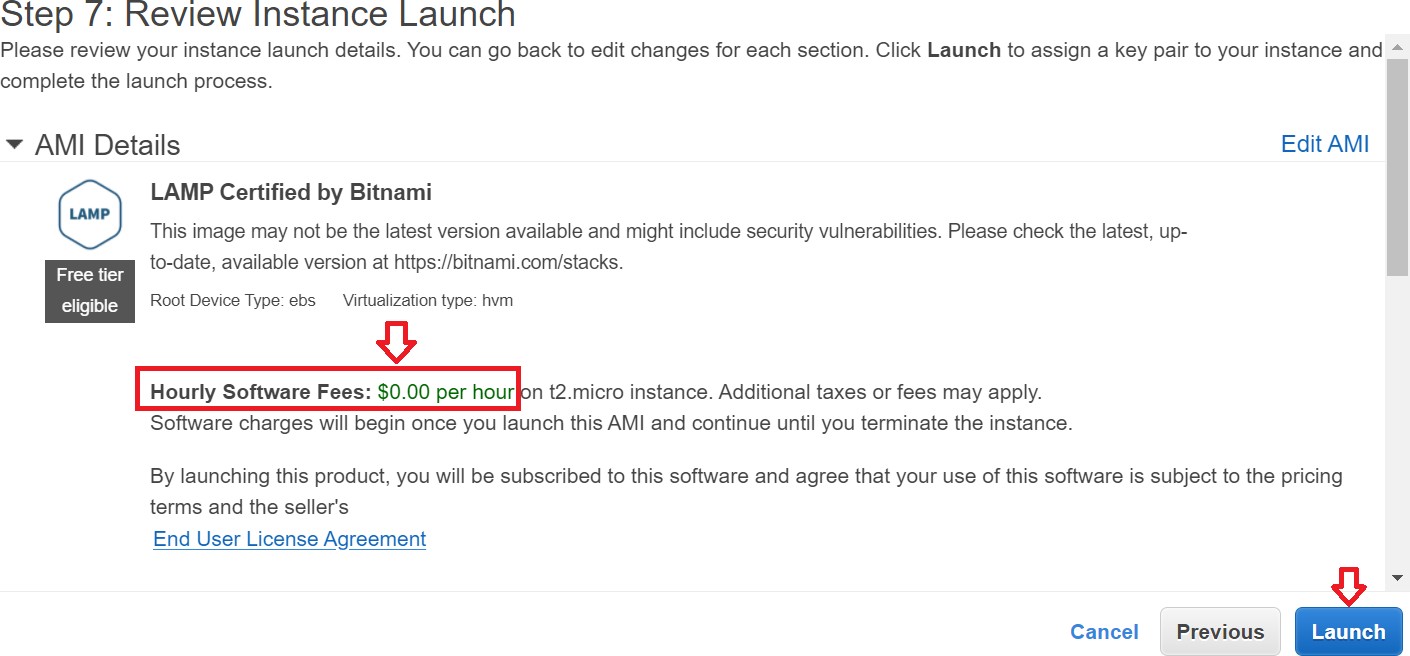
1. Click the “Select” button. A dialog box pops up to show all the pricing plans. Just click “Continue”

to close the dialog box.

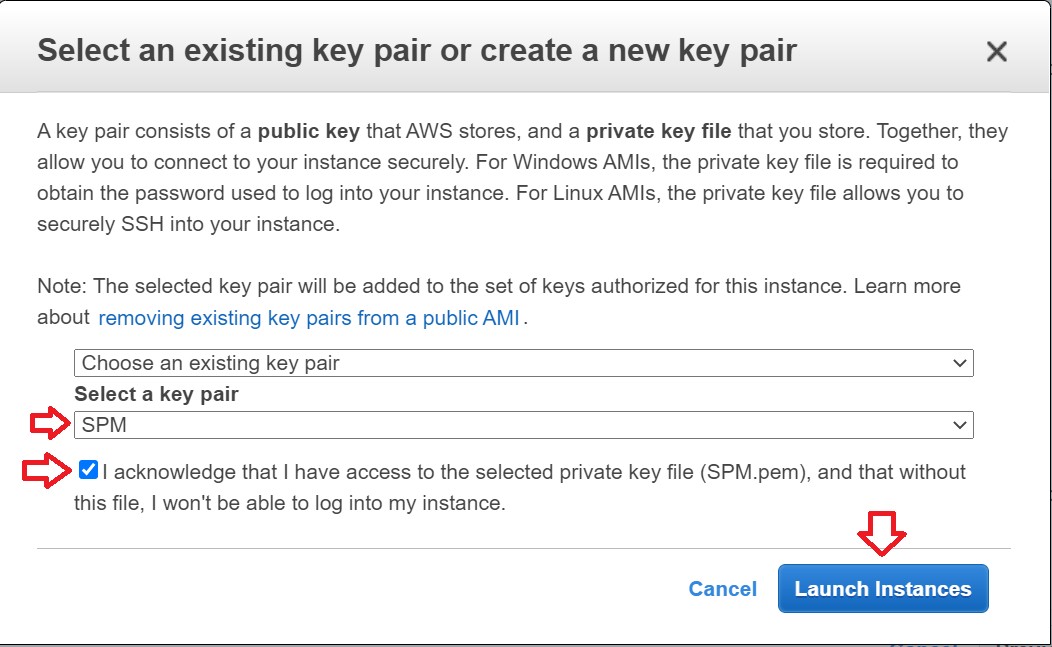
You will choose an instance type (i.e. pricing plan). Select **t2.micro** (free tier eligible!), and click

### Review and Launch.

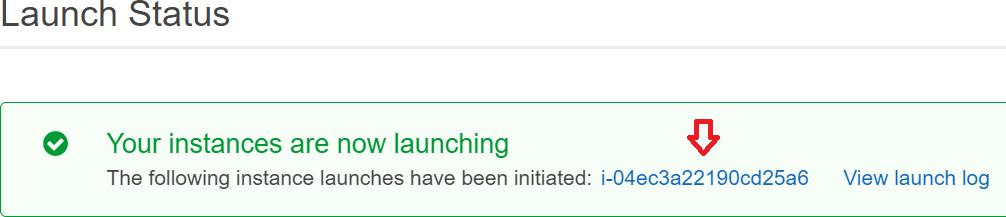
1. Ensure that the fees are “Hourly Software Fees: **$0.00 per hour”** is stated within the loaded page**,** then click **Launch.**



Click “**Choose an existing key pair”** dropdown list, and select **the key pair that you have imported**. Tick the checkbox “I acknowledge that I have access to the selected private key file …” and click the **“Launch instances”** button.

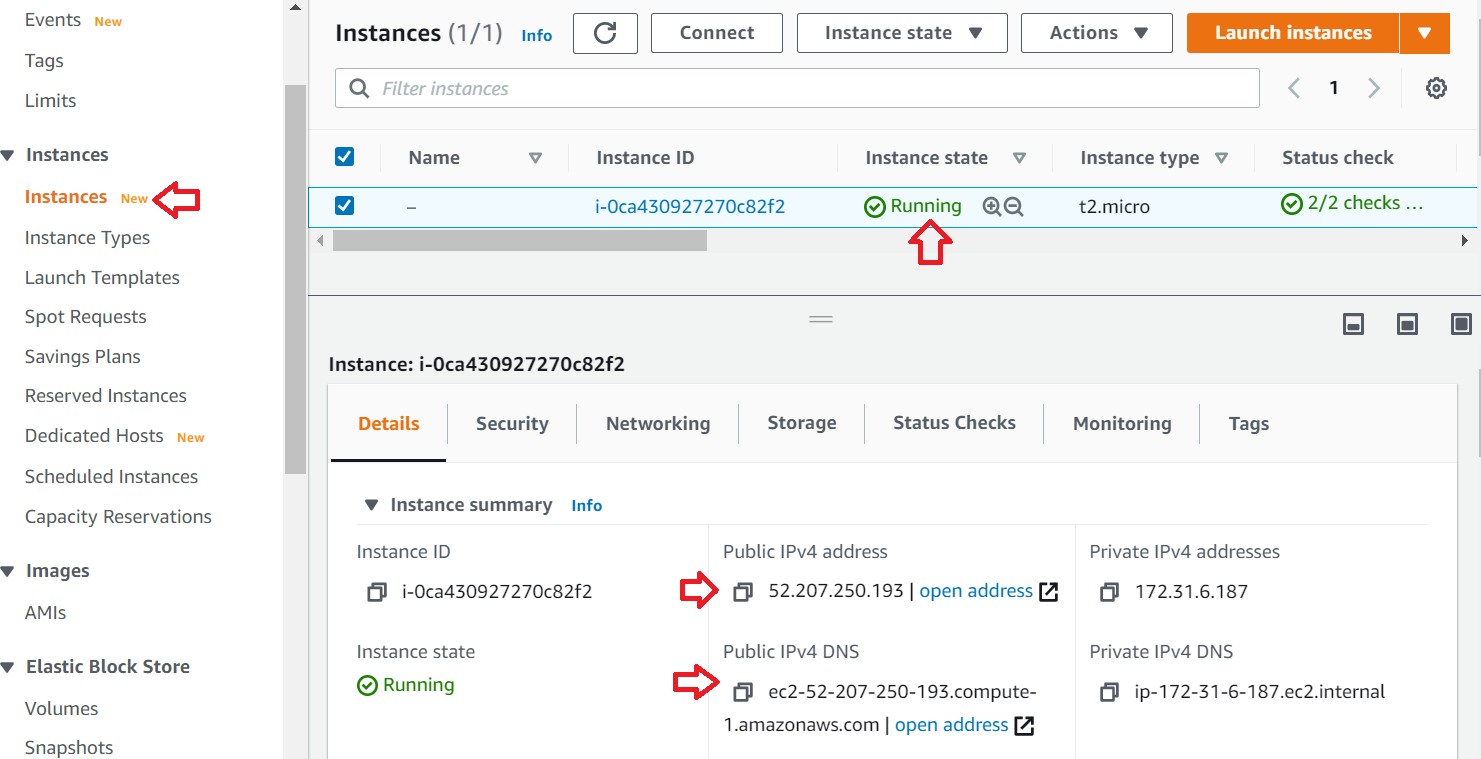


1. Wait for AWS to initiate the instance.



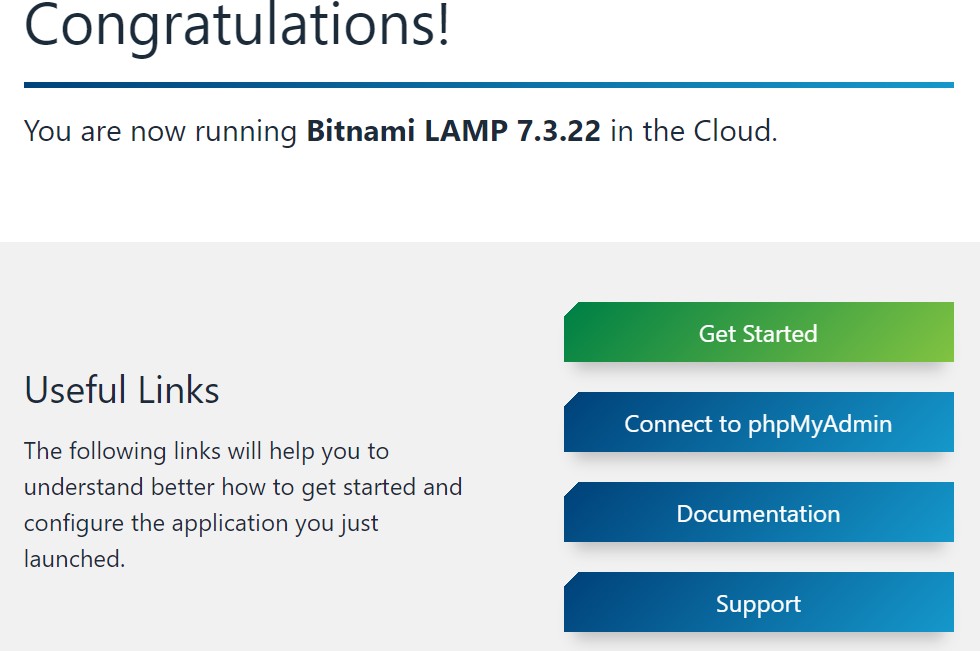
You should see a similar success message “The following instance launches have been initiated: **<ID and link to your instance>**”.

1. Click on the link within the success message to see the status of the virtual machine. The loaded page shows a list of instances that you owned.
   1. This page is also accessible via
      1. Top menu bar > Services > EC2
      2. Left Menu > INSTANCES > Instances
2. Wait till the **Instance State** becomes **running AND status checks** have a **green tick**.
3. Select the instance and you should see more details below the list.



1. Note down the public IP. Do not close this window.
2. Enter the Public DNS or IP address in your browser to see if Apache is running.
   1. In the screenshot, IP is 23.22.140.228, replace with the one on your own screen

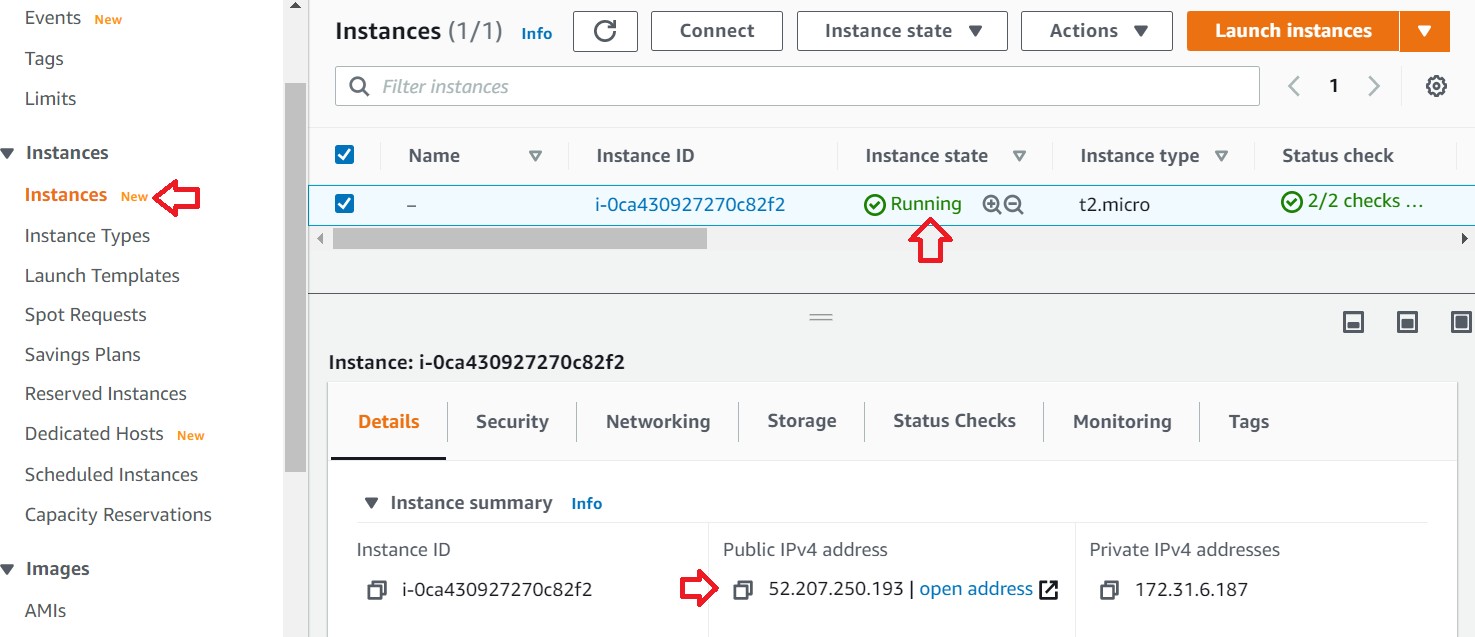
If you see the following screen, it means you are successful in running the Bitnami LAMP instance!



\*\* You can setup Billing Alarms if you would like to limit your cost

# Deploy an application to AWS

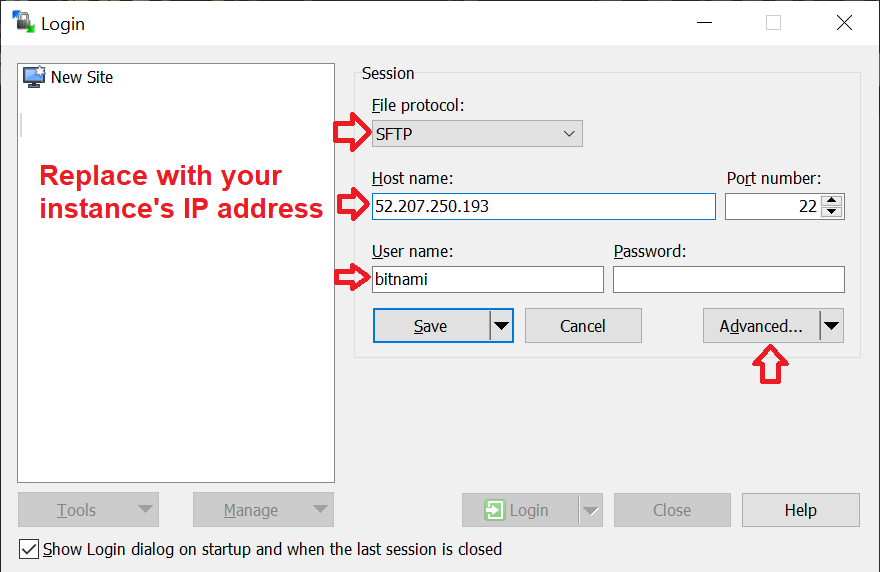
1. Get the IP address of your AWS instance. Select the instance and select the Details Tab. You will find the Public IP address.



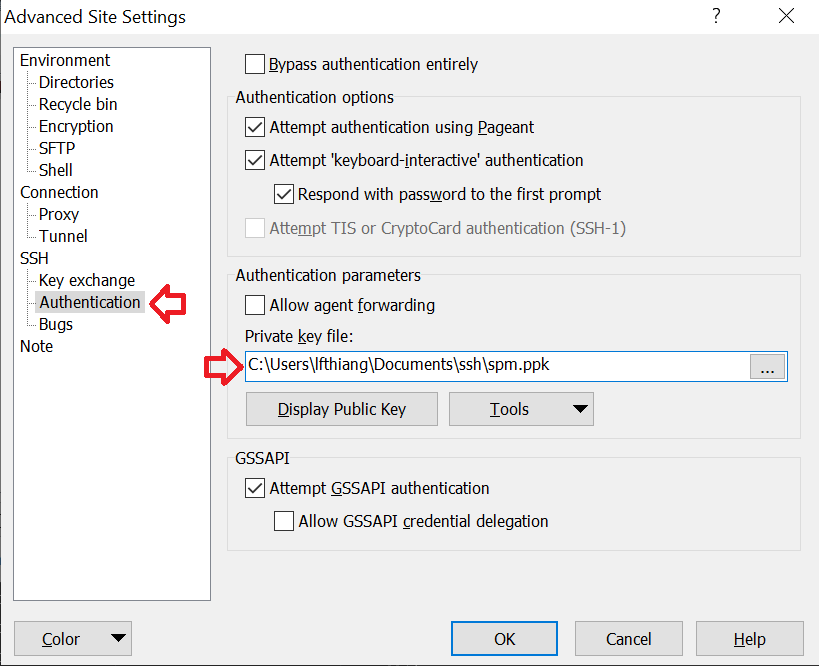
1. Run WinSCP.exe.
   1. Download it if you have not done so. Ref: [Download resources](#_bookmark0)
2. Select ‘New Site’
   1. Hostname is the IP address of your AWS instance.
   2. Username is ‘**bitnami**’ without the quotes.

### Make sure ‘File Protocol’ is ‘SFTP’.

* 1. Click ‘Advanced…’ button.

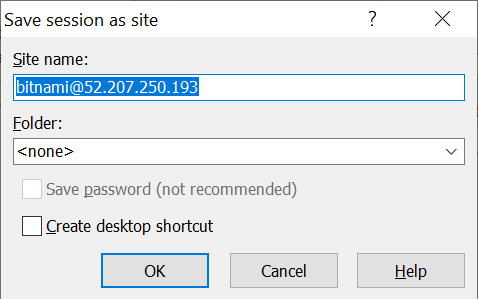


1. In the pop-up window
   1. Left menu > SSH > Authentication
   2. Select your team’s private key file that you have saved
   3. Click ‘OK’



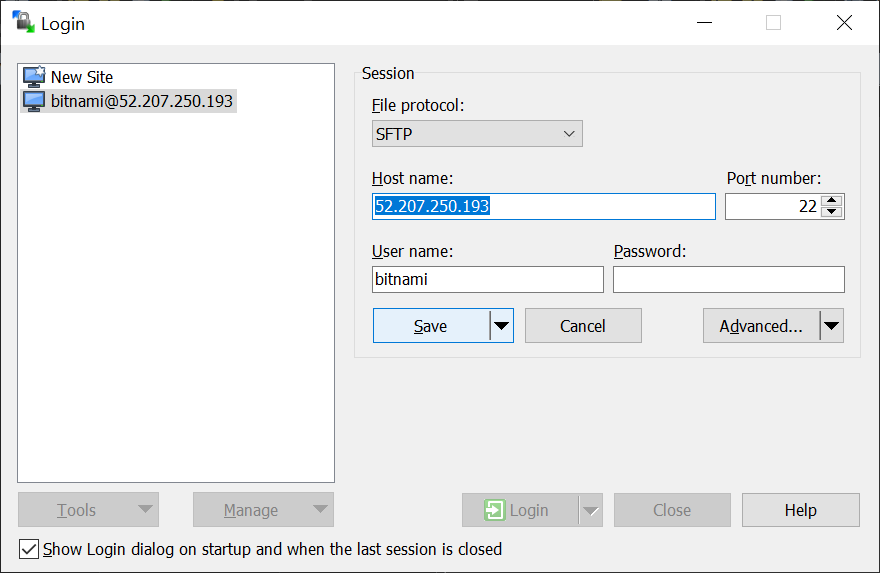
1. You are back to the previous window. Click the ‘Save’ button.

It will prompt you for a site name to save the above details.



1. You may change the ‘Site name’ if you wish. Click the ‘Ok’ button and you will go back to WinSCP

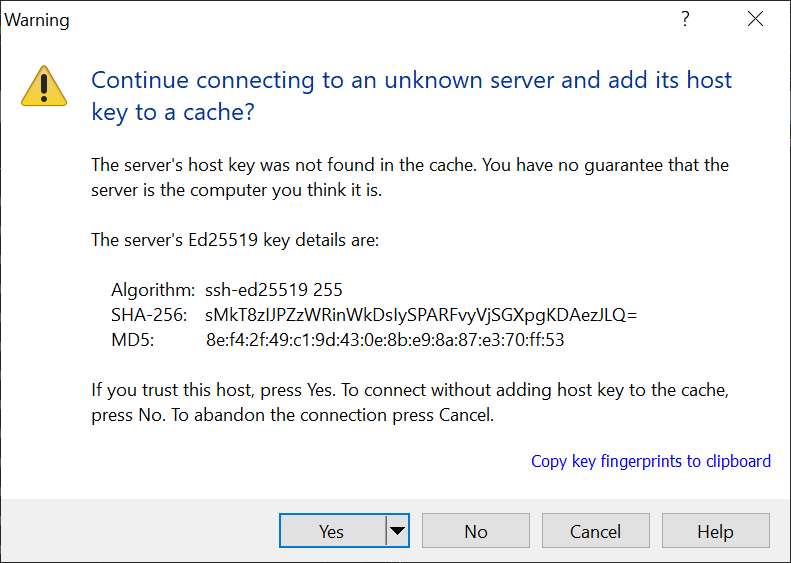
default login window.



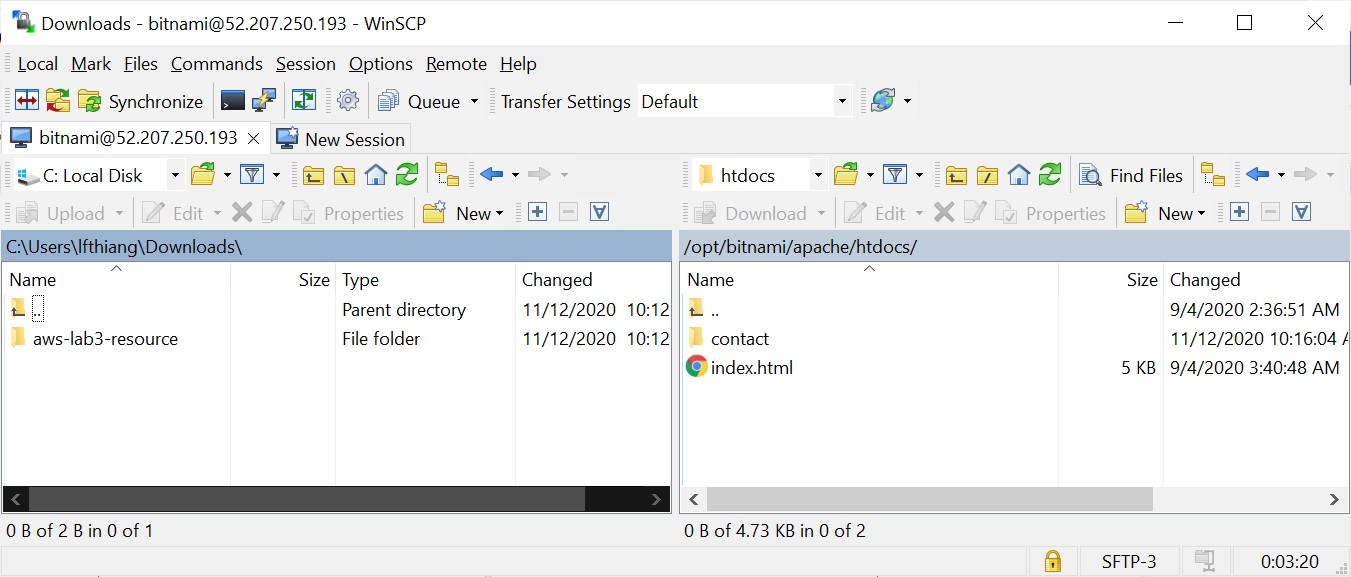
1. Select your site (from the list on the left) and click the ‘Login’ button.

If you see a warning “Continue connecting to an unknown server and add its host key to a cache?”,

click the “Yes” button.



1. Once login successfully, you will see
   1. Left side is your computer/laptop.
   2. Right side is the AWS instance.



1. Create hello.php.

<html>

<body>

<h1>Hello</h1>

<?php

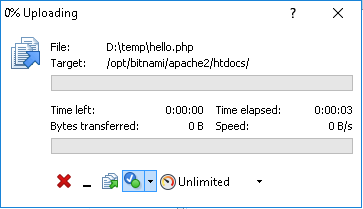
$now = date('Y-m-d H:i:s'); echo $now;

?>

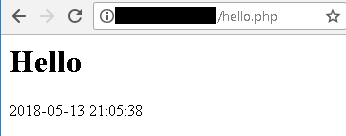
</body>

</html>

1. In WinSCP,
   1. On the left side, navigate to your hello.php
   2. On the right, double click the folder ‘**htdocs**’. This is the **web root folder** on your AWS instance.
      1. Alternatively, menu bar > Remote > Go to > Open Directory/Bookmark
      2. Enter /opt/bitnami/apache2/htdocs
      3. Click the “OK” button.
   3. Drag and drop hello.php from the left to the right.



1. When upload is done, go to your browser, enter your AWS instance’s IP address and access hello.php.



Stop your instance to avoid incurring cost

# Lab 3 – Creating an AWS EC 2 Instance

## *Objectives*

In this exercise, you will learn to

* 1. Setup MySQL & phpMyAdmin on AWS
  2. Deploy another application with database to AWS

## *Introduction*

**Secure Shell (SSH) tunneling** is a security method that maps a port on your local machine to a remote

machine’s port (see diagram below) such that data sent/received is encrypte d.



Port

8888

**Your Machine**

127.0.0.1

**Remote**

**Bitnami LAMP EC2 Instance** 52.X.Y.Z

**SSH Tunnel (Port 22)**

Port 80

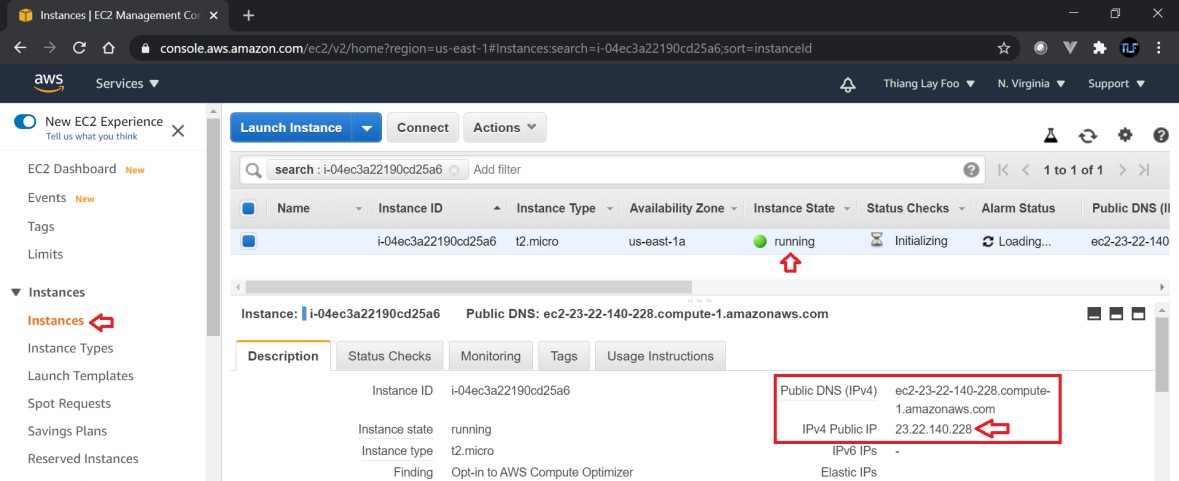
To access **phpmyadmin** on your AWS instance, AWS Cloud mandates the use of SSH tunneling to ensure secure transmission of information over the Internet.

In this lab exercise, we will setup SSH tunneling to access phpmyadmin by mapping remote AWS instance’s

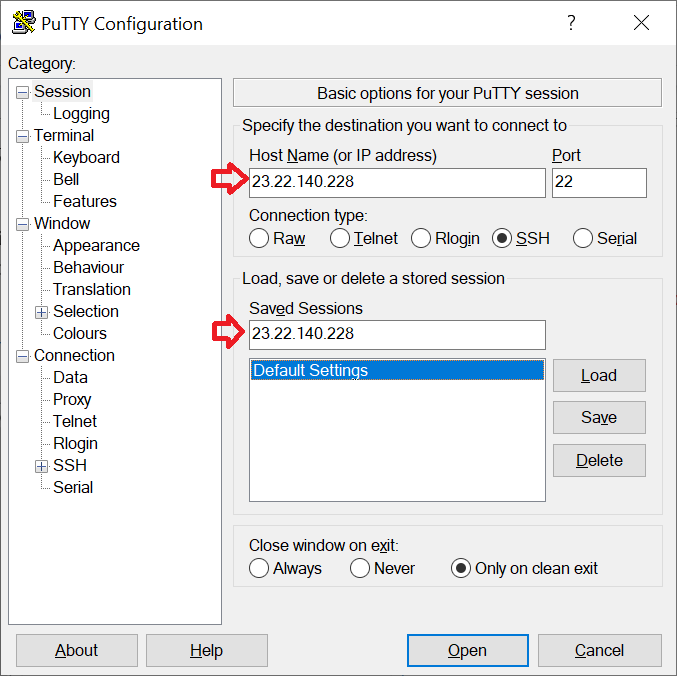
port 80 to local machine’s port 8888 (aka **reverse port forwarding**).

## *Connect to the remote AWS instance using Secure Shell (SSH) protocol*

1. Go to your Amazon EC2 Dashboard and note the IP Address of your EC2 instance.



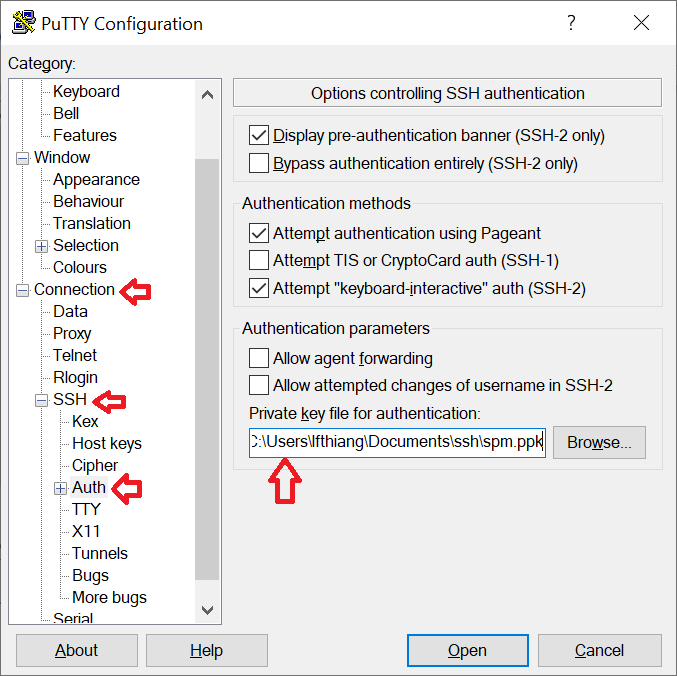
1. Run PuTTy.
2. Enter public IP as the host name in the Putty window. Name your session.



1. Set the authentication key

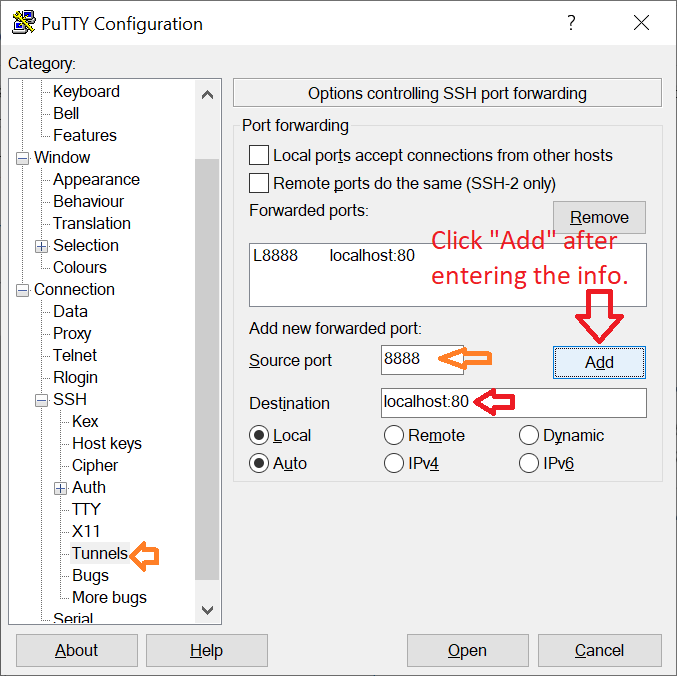
### Left Menu > Connection > SSH > Auth.

* 1. Select **your team’s private key** file that you have saved **from your previous deployment lab**.

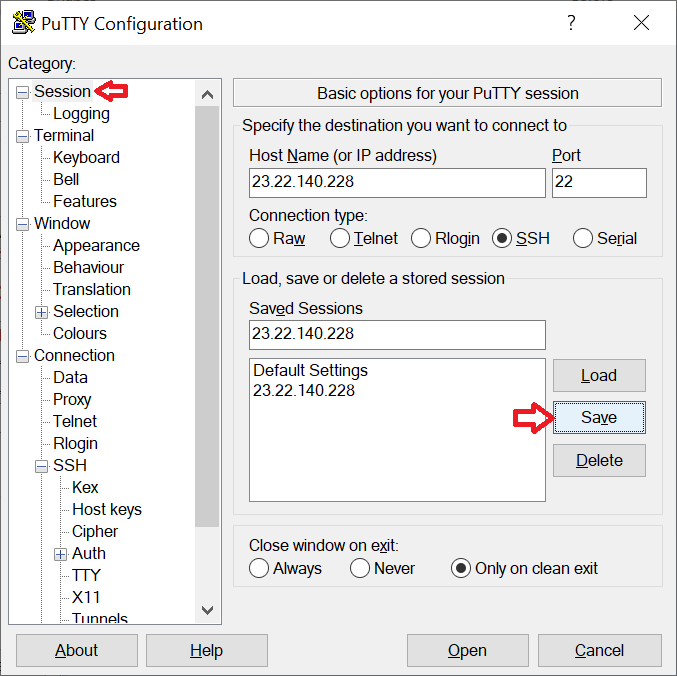


1. To setup the reverse port forwarding for phpmyadmin, go Left Menu> Connection > SSH > Tunnels
   1. Source port "88883"
   2. Destination port "localhost:80"
   3. Click “**Add”** button

You should see the “L8888 localhost:80” in the list of forwarded ports.

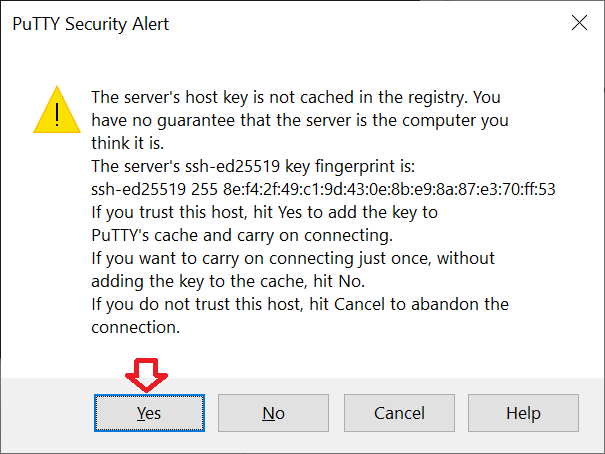


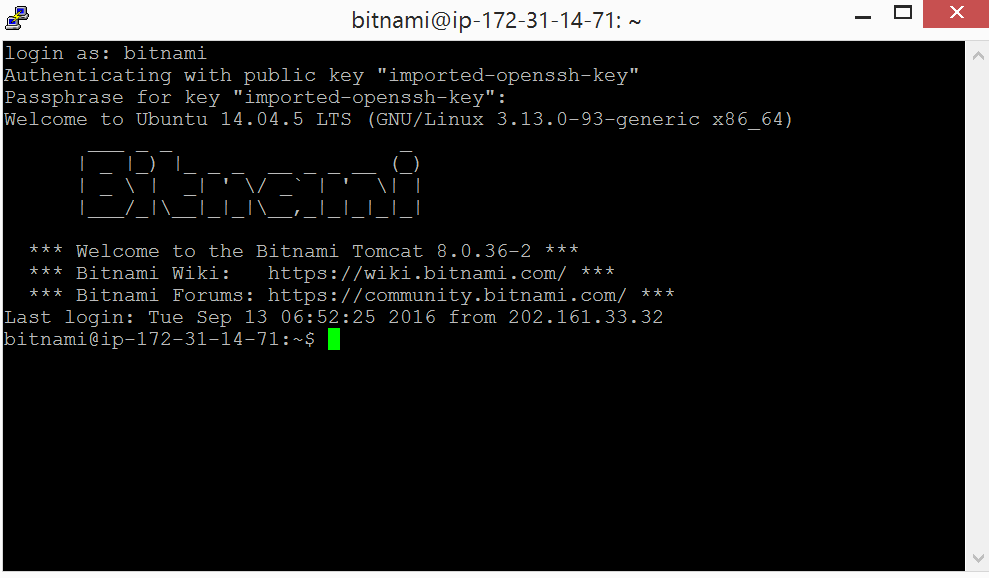
1. Left Menu > Session > Click “**Save”** button to save the session.



3 You may use other port number (>= 1024) if you know 8888 is in use.

1. Click the “**Open**” button to connect to AWS Instance using SSH protocol**.**
2. The following prompt “The server’s host key is not cached…” appears. Click **‘Yes’** to continue.



1. Login as ‘***bitnami***’ without the quotes.
   1. If you had specified a passphrase for your SSH key, key in the passphrase when prompted.
   2. When you key in your passphrase, you **will NOT see** what you typed! This ensures no one can peek at your password from behind you.
2. Done! You have connected to your AWS instance using SSH protocol.

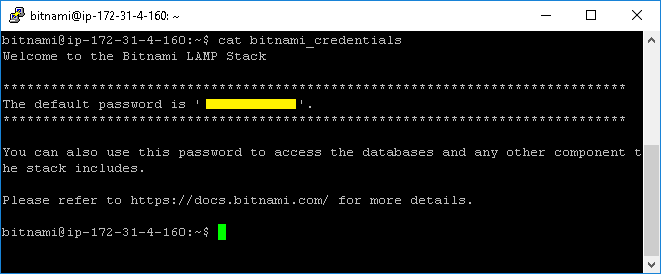
## *Access AWS instance’s phpmyadmin*

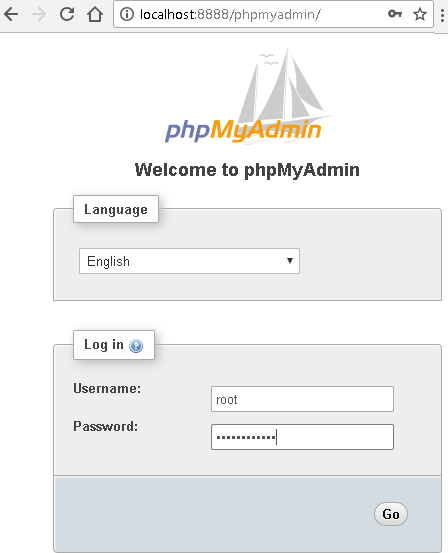
1. In putty, type the following command

cat bitnami\_credentials

and you should see the default **password** for your **AWS** instance’s **MySQL and phpmyadmin**. **Username** is ‘**root**’ without the quotes.

Note down the default password. Highlight the password text in PuTTy will copy it to the clipboard.



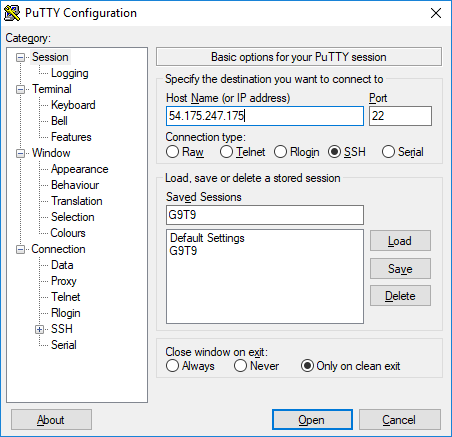
1. Open your browser and go to http://localhost:8888/phpmyadmin/.
2. Login and you can access AWS instance’s **phpmyadmin** using SSH tunneling.

**Note:** When you are using AWS phpmyadmin, **PuTTy must remain open** to keep the SSH tunnel running.

## *Subsequent access to AWS instance’s phpmyadmin*

From now onwards, whenever you want to access phpmyadmin on your AWS instance,

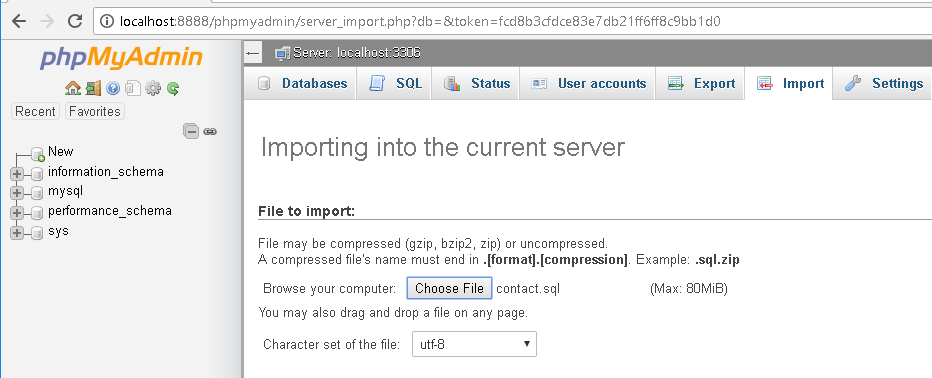
1. Run putty. Load the saved putty session by
   1. Selecting the saved session name and
   2. Click the “Load” button.



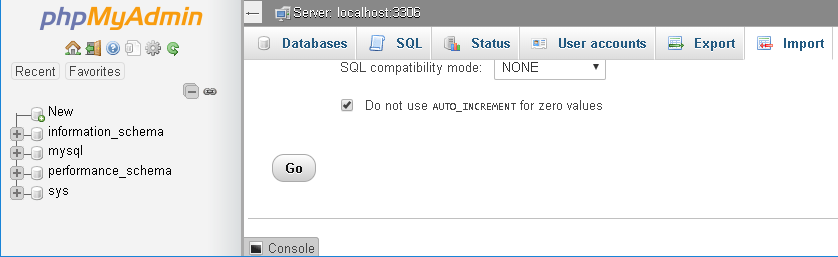
1. Click the “Open” button and continue to login as described to open the SSH tunnel.
2. Open your browser to access phpmyadmin as described above. Keep putty running while you access AWS phpmyadmin.

## *Create a database schema and deploy application*

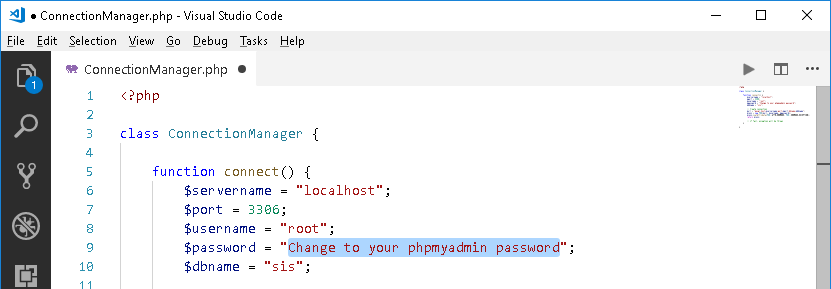
1. Go to AWS phpmyadmin,
   1. Tab “Import”
   2. Click “Choose File” button
   3. Select the contact.sql from the given lab resource (aws-lab3-resource.zip).



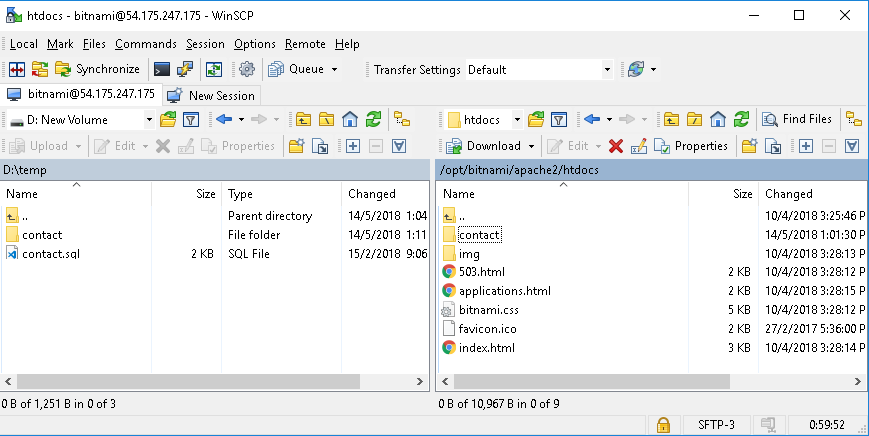
1. Scroll all the way down, and click the “Go” button.



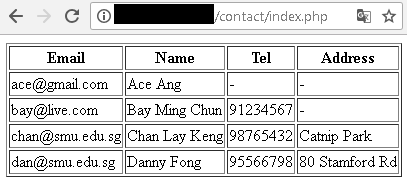
1. Edit ConnectionManager.php in the given lab resource.



1. Using WinSCP to upload the whole “contact” folder to your AWS instance.



1. Go to your browser and access the application. Remember to use your AWS instance’s IP address. The web page should display a list of contacts’ details.



## *Stop/terminate your instance to avoid incurring cost*

### Important!

EC2 > right click on the instance > Instance state

1. “Stop instance” to shut down the virtual machine.
   1. Quote from <https://aws.amazon.com/ec2/pricing/>

*EC2 usage are billed on one second increments, with a minimum of 60 seconds.*

Read the AWS web pages for more information.

* 1. Hence, stop your instance when you are not going to use it for some time to avoid incurring unnecessary charges.

1. “Start instance” to power up the virtual machine.
2. “Terminate instance” to delete the virtual machine.
   1. Do this if you are not going to use the instance any more.