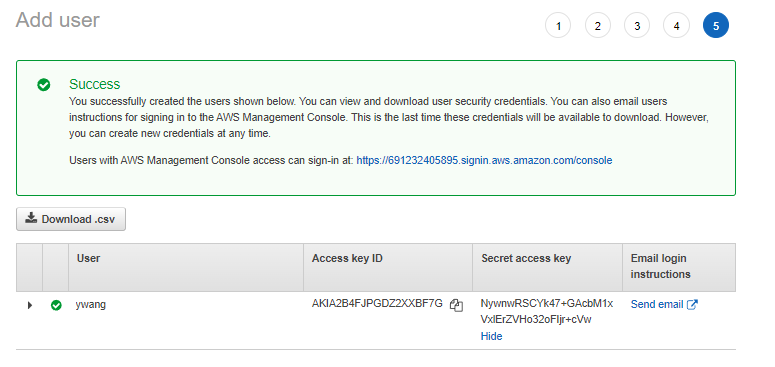
Negin Amou

Link to webpage : <http://ec2-107-22-104-204.compute-1.amazonaws.com>

Or got to this ip : <http://107.22.104.204/>

Make sure it is http not https

|  |
| --- |
| Account ID: 6912-3240-5895  IAM user: ywang  Password: Negin0325817 |
| Console login: https://691232405895.signin.aws.amazon.com/console |
| Access key ID : AKIA2B4FJPGDZ2XXBF7G |
|  |
| Secret access key |
| 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 |



1-Create EC2 account . This step is straightforward. The documentation to follow for account set up is provided here:

[Create and activate an AWS account (amazon.com)](https://aws.amazon.com/premiumsupport/knowledge-center/create-and-activate-aws-account/)

Graphical user interface, application, website

Description automatically generatedGraphical user interface, text, application

Description automatically generated2-Next step is creating an instance . go to services. Select Ec2. Select a region for server and launch instance .

Graphical user interface, text, application, email

Description automatically generated

This is how the page looks like for creating an instance. Select an OS of choice. Make sure to either create a key-pair or upload something you have created previously .

This is the last chance to download it.

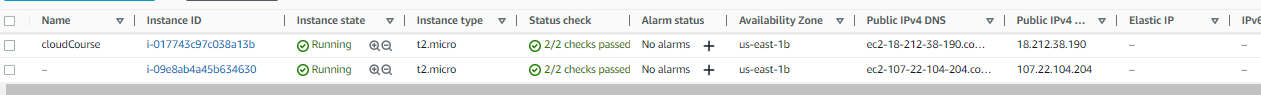
At the bottom of the same page it is better to tick the options to allow http and https traffic if we plan to make a web server on that machine. This setting can also be set in the security groups where we allow any traffic from HTTP and HTTPS.

Graphical user interface, text, application, email

Description automatically generatedI kept the memory information as default since I wasn’t going to use this instance for anything too heavy .

Graphical user interface, text, application, email

Description automatically generated

If the steps are done successfully we should be able to see the machines in running state. 

Click on any of the instances we want to see further details. Information to take note are public ip, elastic ip and Public IPv4 DNS. These are the addresses we can use to navigate to a running web page using apache.Graphical user interface, application, email

Description automatically generated

To login to Ubuntu machine we can either use putting or PowerShell. If we are using putting make sure to first load your key pair in the authentication section then add your host name which can be found from the instance details (Picture above).

Graphical user interface, text, application

Description automatically generated

Similarly to upload files in the machine we can use WinScp and similar to putty we have to provide our key pair file.

If we want to use PowerShell we can use ssh to connect . Make sure you include the path to your key pair in the command. (note : we can convert ppk file to pem file using puttygen in case we forgot to download both format while creating the instance )

ssh -i "[pathToFile]/MyEC2Instance.pem" [ubuntu@ec2-107-22-104-204.compute-1.amazonaws.com](mailto:ubuntu@ec2-107-22-104-204.compute-1.amazonaws.com)

Once we are in the machine we can treat it like any ubuntu computer when we want to host a website. To host my website I installed Lampp first. Next make sure to run apache and mysql services. I followed this tutorial to install them :

[How to install LAMP stack web server on Ubuntu 20.04 - UpCloud](https://upcloud.com/resources/tutorials/installing-lamp-stack-ubuntu)

Other extra steps to take is make sure error reporting in php.ini is on as it turns out it can be off by default in Lamp. Other steps to take is to uncomment extension=mysqli or PDO in php.ini to be able to use them in your project for database connection. To make sure mysql and apache are running test them by trying to login to mysql from machine running the following command:

Mysql -u root -p

To check if apache2 is running correctly , try navigating to your public ip or elastic ip. If the page doesn’t load, it could be either because apache2 isnt running or in your security rules you are not allowing traffic through http or https . To allow it go to security group in side bar. Find the security group that your machine uses. In my case it is launch-wizard-3.

Text

Description automatically generated

Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, application

Description automatically generated

Click on edit in bound rules and add http and https traffic to be allowed. (it can be allowed from anywhere or specific ips)

To add your website files , either git clone or use winscp. Load them in /var/www/html/ directly.

The basic of creating a a connection can be seen below. Note that we can use either localhost or public ip of the machine. I personally used localhost . One small details to note is that it is recommended to create a sperate database user instead of root to use in web application.

$servername = "localhost";

$username = "ngn";

$password = "passwordHard!";

$dbname = "ass2\_cl";

$conn = new mysqli($servername, $username, $password, $dbname);

Command to create a new user in mysql :

CREATE USER 'username'@'host' IDENTIFIED WITH authentication\_plugin BY 'password';

Make sure to give privilege of using the database we plan to use in project to the user.

GRANT PRIVILEGE ON database.table TO 'username'@'host';

The rest of the application are web development related and doesn’t have much to do with cloud computing set up. I included the sql tables as a separate file in case grader wants to take a look at. The project is also cloneable from github in case it was to be expanded or replicated.