

# Step 1 – Preparing prerequisites

---

In order to complete this project you will need an AWS account and a virtual server with Ubuntu Server OS.

AWS is the biggest Cloud Service Provider and it offers a free tier account that we are going to leverage for our projects.

Do not focus too much on AWS itself right now, there will be a proper Cloud introduction and configuration projects later in our course.

Right now, all we need to know is that AWS can provide us with a free virtual server called EC2 (Elastic Compute Cloud) for our needs.

Spinning up a new EC2 instance (an instance of a virtual server) is only a matter of a few clicks.

You can either Watch the videos below to get yourself set up.

- 1.AWS account setup and Provisioning an Ubuntu Server
- 2.Connecting to your EC2 Instance

Or follow the instructions below.

- 1.Register a new AWS account following this instruction.
- 2.Select your preferred region (the closest to you) and launch a new EC2 instance of t2.micro family with Ubuntu Server 20.04 LTS (HVM) launch EC2

IMPORTANT – save your private key (.pem file) securely and do not share it with anyone! If you lose it, you will not be able to connect to your server ever again!

For Windows users, you will need a tool called putty to connect to your EC2 Instance. Download Putty [Here](#). For Mac users, you can simply open up Terminal and use the ssh command to get into the server.

IMPORTANT NOTICE Both Putty and ssh use the SSH protocol to establish connectivity between computers. It is the most secure protocol because it uses crypto algorithms to encrypt the data that is transmitted – it uses TCP port 22 which is open for all newly created EC2 instances in AWS by default. Most of these terminologies will make more sense to you as you proceed. So for now, if nothing makes sense, just ignore. But be assured that the information is already registered in your sub-conscious mind. So it will become useful to you soon.

The process to connect to the virtual server is different between Windows and Mac. So lets take a quick tour.

# Connecting to EC2 terminal

## Using the terminal on MAC/Linux

---

- The terminal is already installed by default. You just need to open it up.
- You do not need to convert to a .ppk file. Just use the same key as downloaded from AWS.
- Change directory into the location where your PEM file is. Most likely will be in the Downloads folder

```
cd ~/Downloads
```

IMPORTANT - Anywhere you see these anchor tags < > , going forward, it means you will need to replace the content in there with values specific to your situation. For example, if we need you to replace the name you have saved the private key on your machine, we will write something like < private-key-name >.

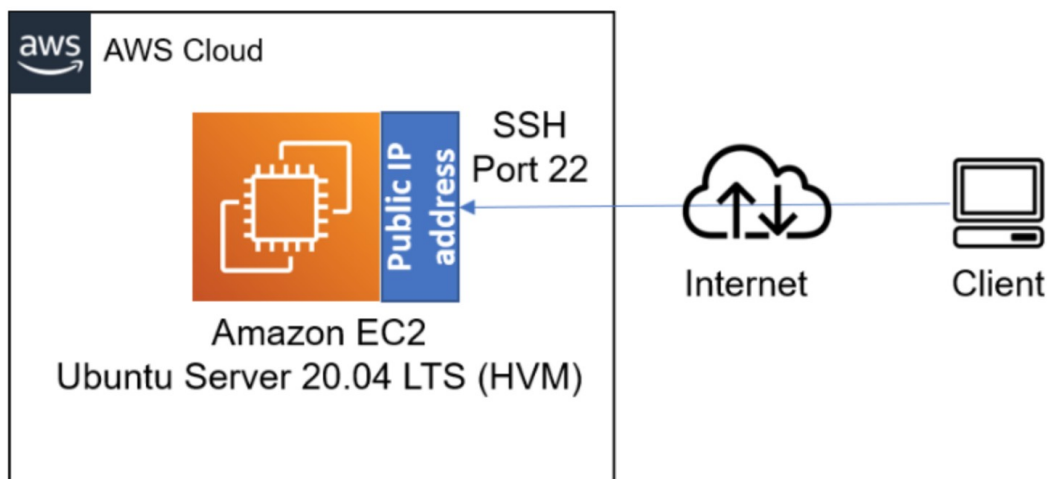
If the private key you downloaded was named my-private-key.pem simply remove the anchor tags and insert my-private-key.pem in the command you are required to execute. Lets try this and follow the instructions below to get some work done.

- Change premissions for the private key file (.pem), otherwise you can get an error "Bad permissions"

```
sudo chmod 0400 <private-key-name>.pem
```

- Connect to the instance by running

```
ssh -i <private-key-name>.pem ubuntu@<Public-IP-address>
```



Congratulations! You have just created your very first Linux Server in the Cloud and our set up looks like this now: (You are the client)

Please read information about AWS free tier limits and make sure that you STOP your EC2 instance when you are not using it.

Stop EC2

All we need to know right now is that we can use 750 hours (31.25 days) of t2.micro server per month for the first 12 months FOR FREE.

You can launch and stop new instances when you need to, but by default there is a soft-limit of maximum 5 running instances at the same time. In our first projects we will be using only 1 running instance at a time. When you stop an instance – it stops consuming available hours.

Note that every time you stop and start your EC2 instance – you will have a new IP address, it is normal behavior, so do not forget to update your SSH credentials when you try to connect to your EC2 server.

Let us move on and configure our EC2 machine to serve a Web server!