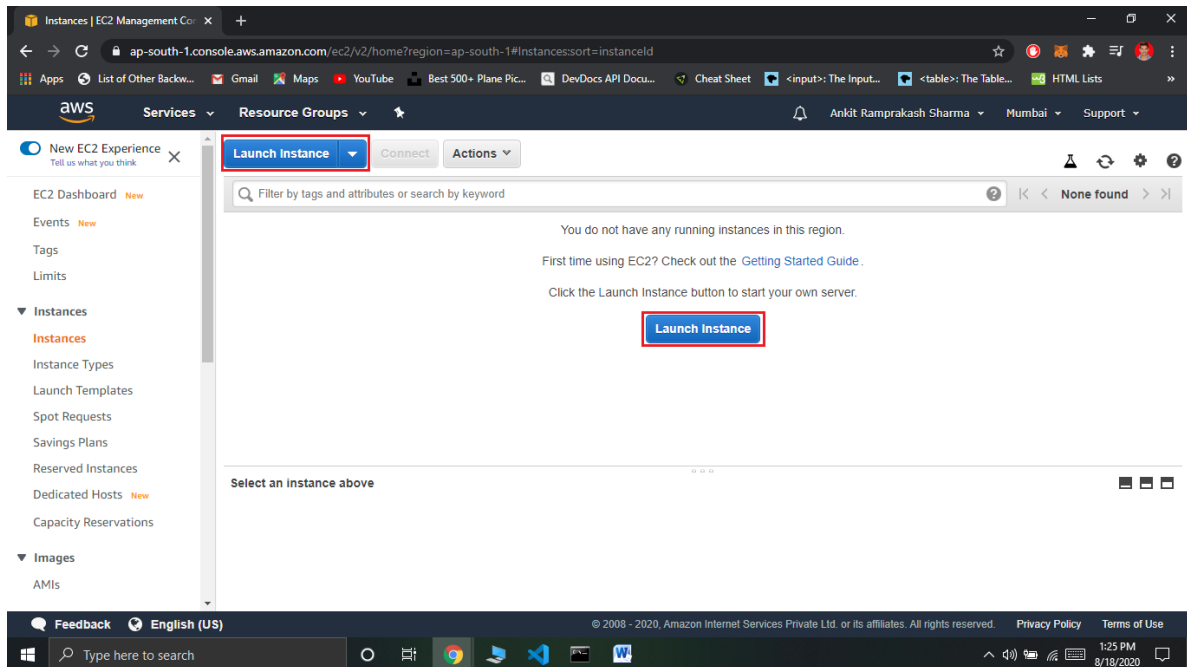


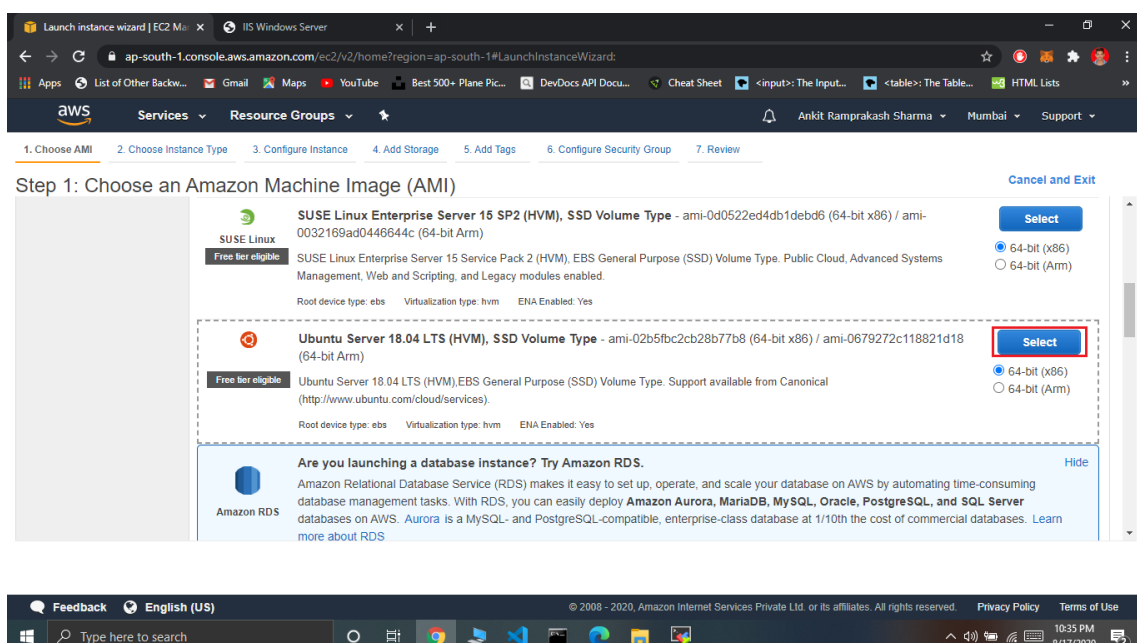
# Project 2:

## Deploying a Web Server in Ubuntu Instance:

Step 1: Login to your AWS Console -> Go to Services -> Select EC2 -> In EC2 Dashboard select Instances -> Click on Launch Instance.



Step 2: Starting with EC2 (Elastic Compute Cloud) and launching a new instance Choose an AMI -> Ubuntu Server 18.04 LTS OS under Free Tier Section.



Step 3: Choose an Instance type which should be free tier eligible -> Select t2 micro and then click, Next: Configure Instance Details.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro Free tier eligible	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes

Cancel Previous Review and Launch Next: Configure Instance Details

Step 4: Configure Instance Details -> No. of instance = 1, Auto-assign Public IP = Enable -> Click Next: Add Storage.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 Launch into Auto Scaling Group

Purchasing option: ☐ Request Spot instances

Network: vpc-5100e43a (default) Create new VPC

Subnet: No preference (default subnet in any Availability Zone) Create new subnet

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: ☐ Add instance to placement group

Capacity Reservation: Open

IAM role: None Create new IAM role

Shutdown behavior: On

Cancel Previous Review and Launch Next: Add Storage

Step 5: Let everything be default in Add Storage.

Delete on Termination must be selected. -> Click Next: Add Tags.

The screenshot shows the AWS Launch Instance Wizard at Step 4: Add Storage. The wizard is for an EC2 instance in the ap-south-1 region. The storage configuration table shows a single volume for the root device (/dev/sda1) with a size of 8 GiB, using General Purpose SSD (gp2) storage type. The 'Delete on Termination' checkbox is checked and highlighted with a red box. The 'Encryption' dropdown is set to 'Not Encrypt'. Below the table, there is a button 'Add New Volume' and a note about free tier eligibility. At the bottom right, the 'Next: Add Tags' button is highlighted with a red box.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-01c49bd5fe5f144e2	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous Review and Launch Next: Add Tags

Step 6: Enter any name you want for your instance in Add Tags. -> Click Next: Configure Security Group.

The screenshot shows the AWS Launch Instance Wizard at Step 5: Add Tags. The wizard is for an EC2 instance in the ap-south-1 region. The 'Add Tags' section shows a table with two columns: 'Key' and 'Value'. A tag is added with the key 'Name' and the value 'Ubuntu'. The 'Instances' and 'Volumes' checkboxes are both checked. At the bottom right, the 'Next: Configure Security Group' button is highlighted with a red box.

Key (128 characters maximum)	Value (256 characters maximum)	Instances	Volumes
Name	Ubuntu	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

Step 7: In Configure Security Group -> Create a new security group ->

Select Type = All Traffic and Source = Anywhere. -> Click Next: Review and launch.

The screenshot shows the 'Step 6: Configure Security Group' page in the AWS Launch Instance Wizard. The 'Assign a security group' section has 'Create a new security group' selected. The 'Security group name' is 'launch-wizard-2' and the 'Description' is 'launch-wizard-2 created 2020-08-17T22:39:22.335+05:30'. A table lists the configured rules:

Type	Protocol	Port Range	Source	Description
All traffic	All	0 - 65535	Anywhere	e.g. SSH for Admin Desktop

A warning message states: 'Warning: Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' At the bottom right, the 'Review and Launch' button is highlighted with a red box.

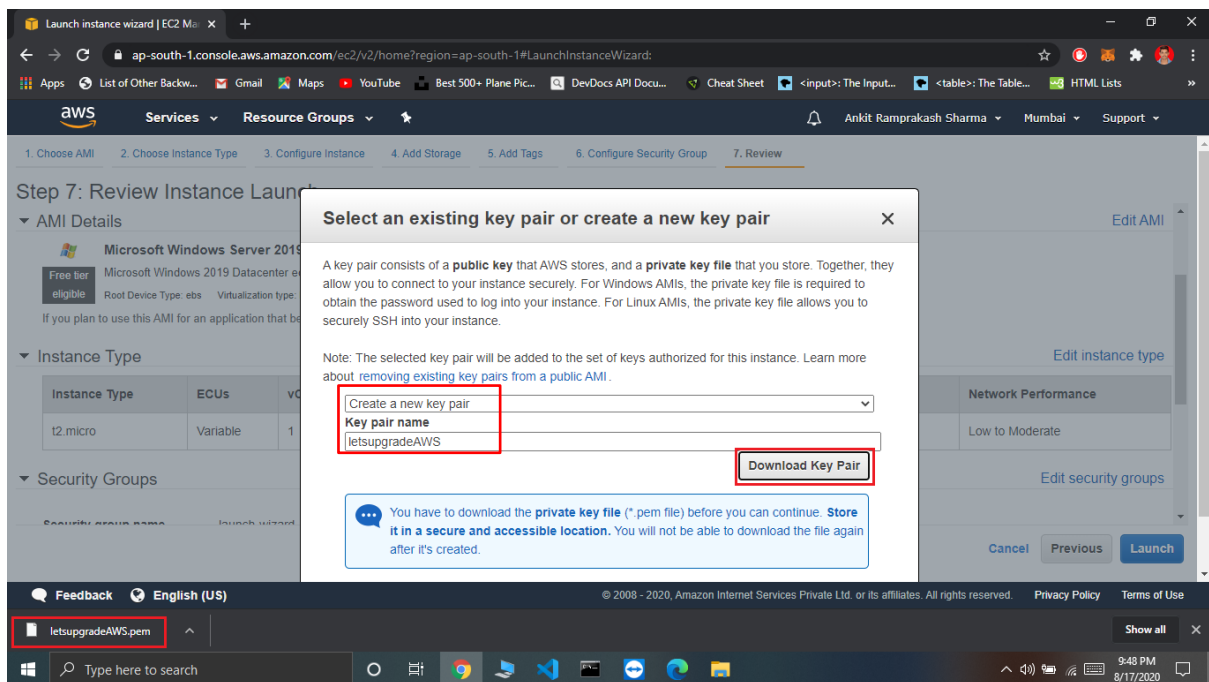
Step 8: Review all steps -> Click Launch.

The screenshot shows the 'Step 7: Review Instance Launch' page. A warning message at the top states: 'Improve your instances' security. Your security group, launch-wizard-2, is open to the world. Your instances may be accessible from any IP address. We recommend that you update your security group rules to allow access from known IP addresses only. You can also open additional ports in your security group to facilitate access to the application or service you're running, e.g., HTTP (80) for web servers. Edit security groups'. Below this, the 'AMI Details' section shows 'Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-02b5fbc2cb28b77b8'. The 'Instance Type' section shows a table with the following data:

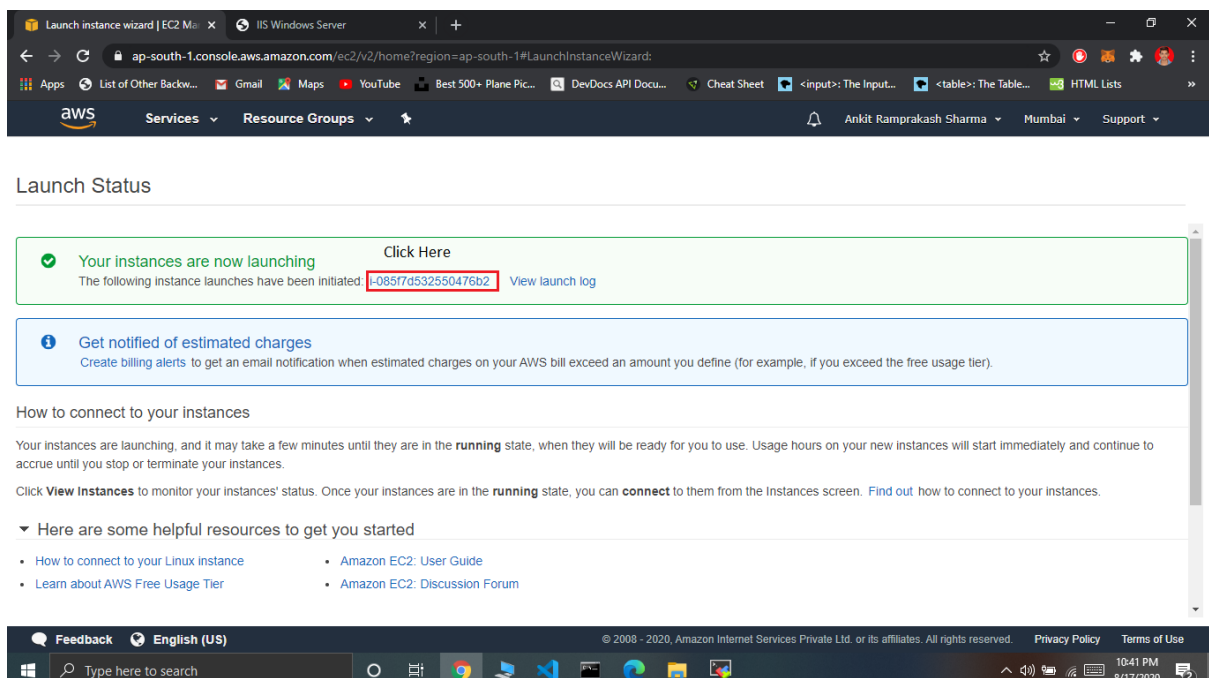
Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	Variable	1	1	EBS only	-	Low to Moderate

At the bottom right, the 'Launch' button is highlighted with a red box.

Step 9: After launching select the existing key pair which you downloaded while creating windows instance **or** create a new key pair show below if you don't have the previous one. -> Click Launch Instance.

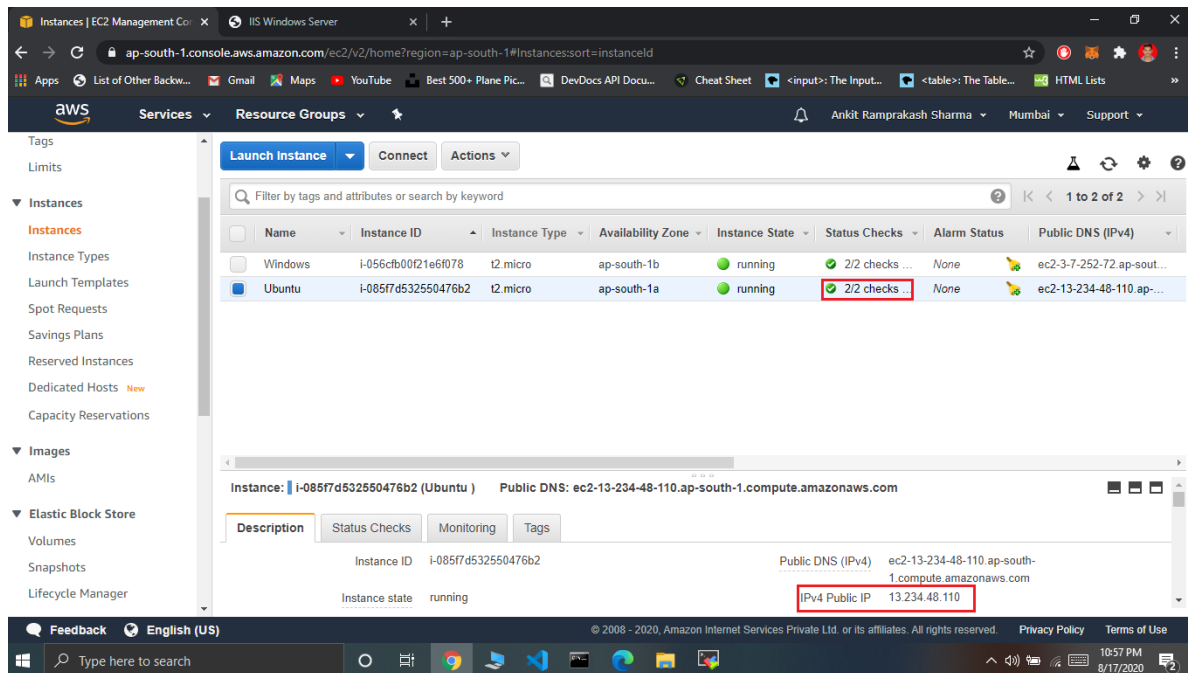


Step 10: Instance is created. -> Click on instance id which will redirect you to EC2 instances list.



Step 11: Wait till the status checks are done. After Status Checks are done, copy the Public IP Address of the Ubuntu instance.

Here it is: **13.234.48.110**

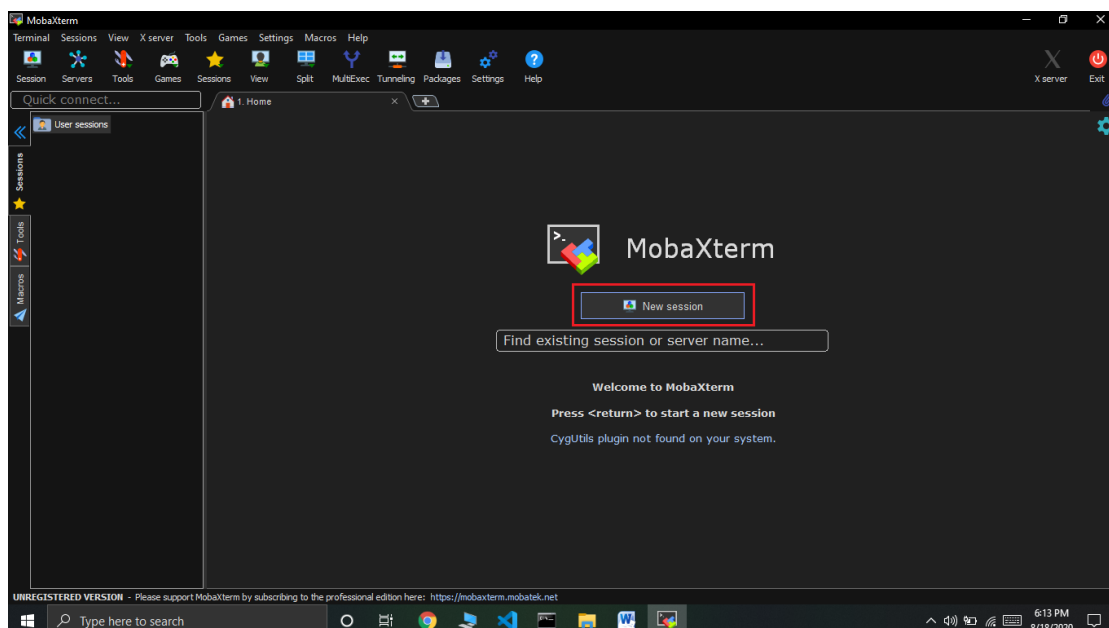


Step 12: **Now to connect and launch your web server**, Download MobaXterm Portable Edition from the link below:

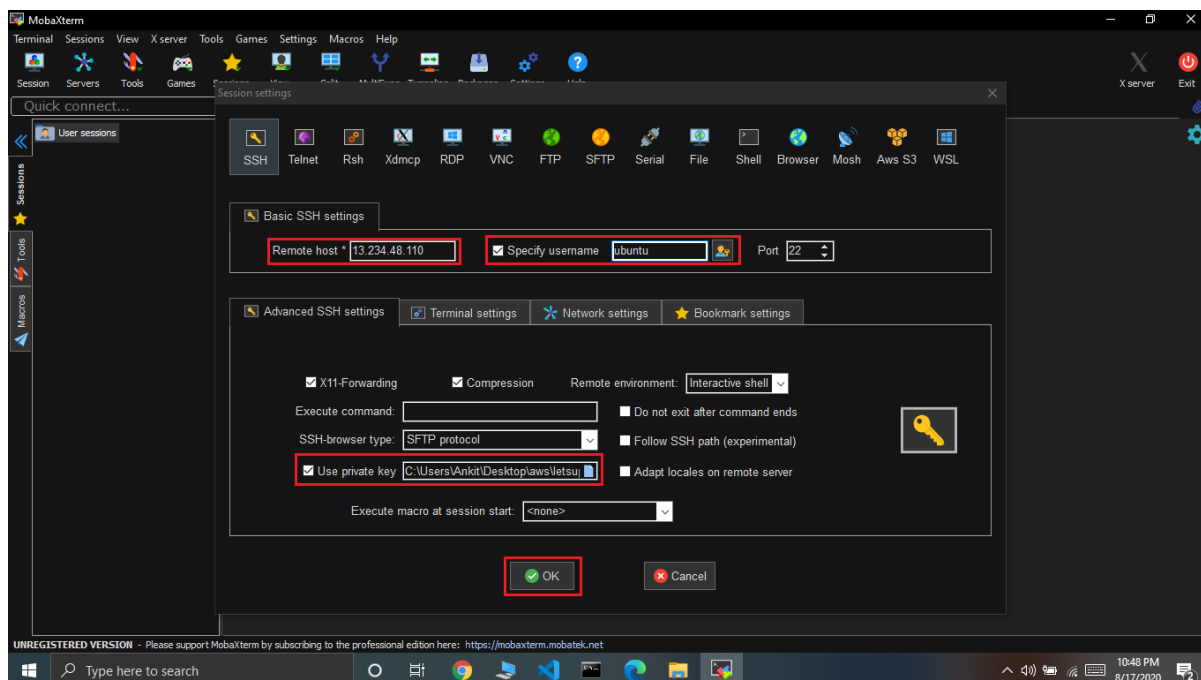
<https://mobaxterm.mobatek.net/download-home-edition.html>

After downloading extract the zip file to get the MobaXterm.exe file.

Step 13: Open MobaXterm.exe -> click on New Session



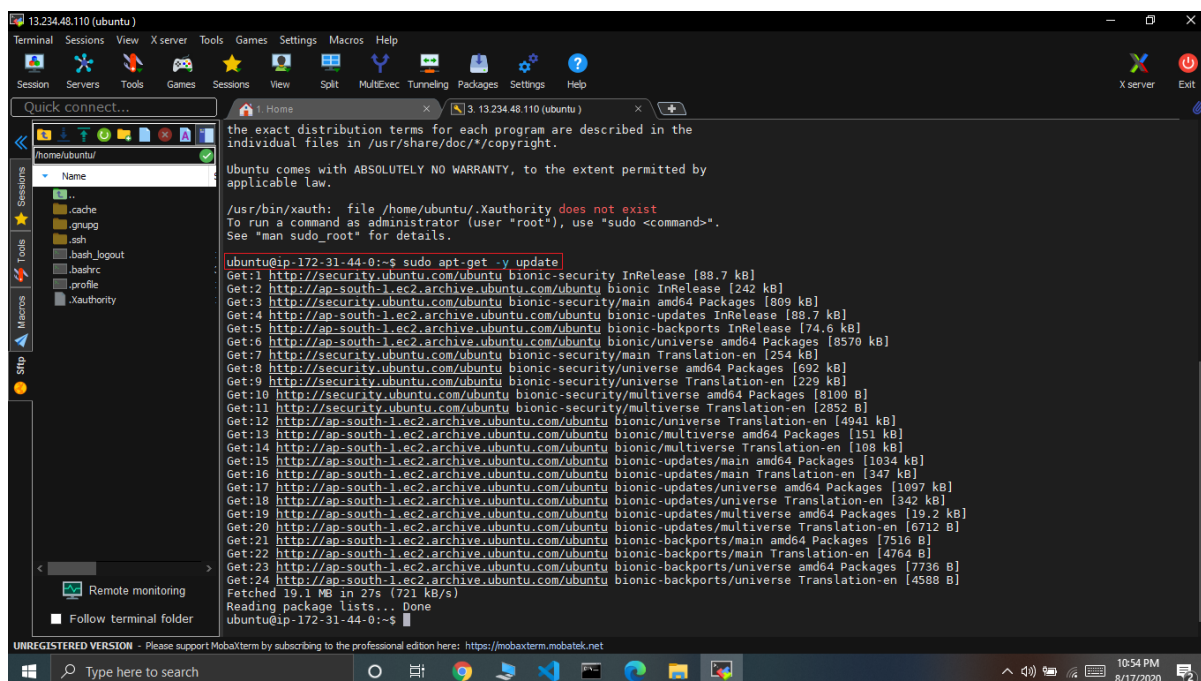
Step 14: Select SSH and give it a specific username -> Paste the Public IP Address generated after Ubuntu instance creation from Step 11 in Remote host and upload your key pair .pem file from Step 9 in Advanced SSH Settings -> click OK.



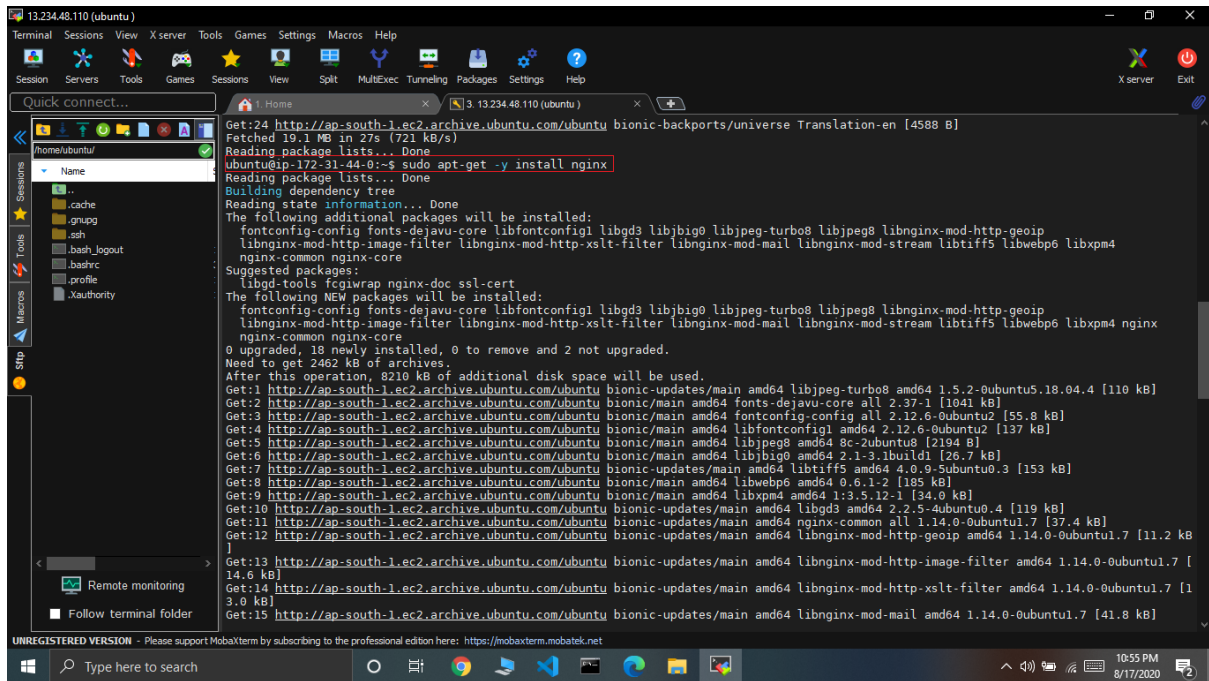
Step 15: After clicking OK an Ubuntu Bash will appear in that run the following two commands given below one at a time:

```
sudo apt-get -y update
```

```
sudo apt-get -y install nginx
```





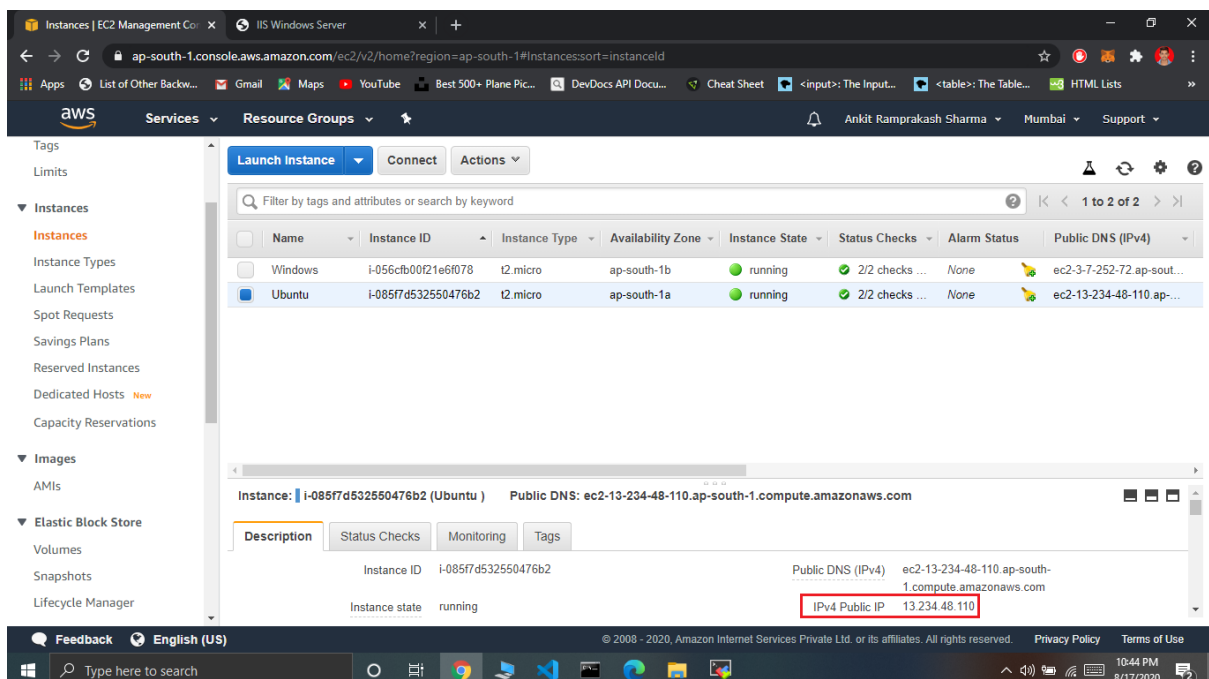


```
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-backports/universe Translation-en [4588 B]
Fetched 10.1 MB in 27s (721 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-44-0:~$ sudo apt-get -y install nginx
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg-turbo8 libjpeg9 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4
  nginx-common nginx-core
Suggested packages:
  libgd-tools fcgiwrap nginx-doc ssl-cert
The following NEW packages will be installed:
  fontconfig-config fonts-dejavu-core libfontconfig1 libgd3 libjpeg-turbo8 libjpeg9 libnginx-mod-http-geoip
  libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter libnginx-mod-mail libnginx-mod-stream libtiff5 libwebp6 libxpm4 nginx
  nginx-common nginx-core
0 upgraded, 18 newly installed, 0 to remove and 2 not upgraded.
Need to get 2462 kB of archives.
After this operation, 8210 kB of additional disk space will be used.
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libjpeg-turbo8 amd64 1.5.2-0ubuntu5.18.04.4 [110 kB]
Get:2 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 fonts-dejavu-core all 2.37-1 [1041 kB]
Get:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 fontconfig-config all 2.12.6-0ubuntu2 [55.8 kB]
Get:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libfontconfig1 amd64 2.12.6-0ubuntu2 [137 kB]
Get:5 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libjpeg9 amd64 8c-2ubuntu8 [2194 B]
Get:6 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libjpeg-turbo8 amd64 2.1-3-1build1 [26.7 kB]
Get:7 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtiff5 amd64 4.0.9-5ubuntu0.3 [153 kB]
Get:8 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libwebp6 amd64 0.6.1-2 [185 kB]
Get:9 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic/main amd64 libxpm4 amd64 1:3.5.12-1 [34.0 kB]
Get:10 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgd3 amd64 2.2.5-4ubuntu0.4 [119 kB]
Get:11 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 nginx-common all 1.14.0-0ubuntu1.7 [37.4 kB]
Get:12 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnginx-mod-http-geoip amd64 1.14.0-0ubuntu1.7 [11.2 kB]
Get:13 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnginx-mod-http-image-filter amd64 1.14.0-0ubuntu1.7 [14.6 kB]
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnginx-mod-http-xslt-filter amd64 1.14.0-0ubuntu1.7 [13.0 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates/main amd64 libnginx-mod-mail amd64 1.14.0-0ubuntu1.7 [41.8 kB]
```

These commands will install **nginx web server**.

Step 16: After all the nginx packages are installed. Copy the Public IP Address of the Ubuntu instance.

Here it is: **13.234.48.110**

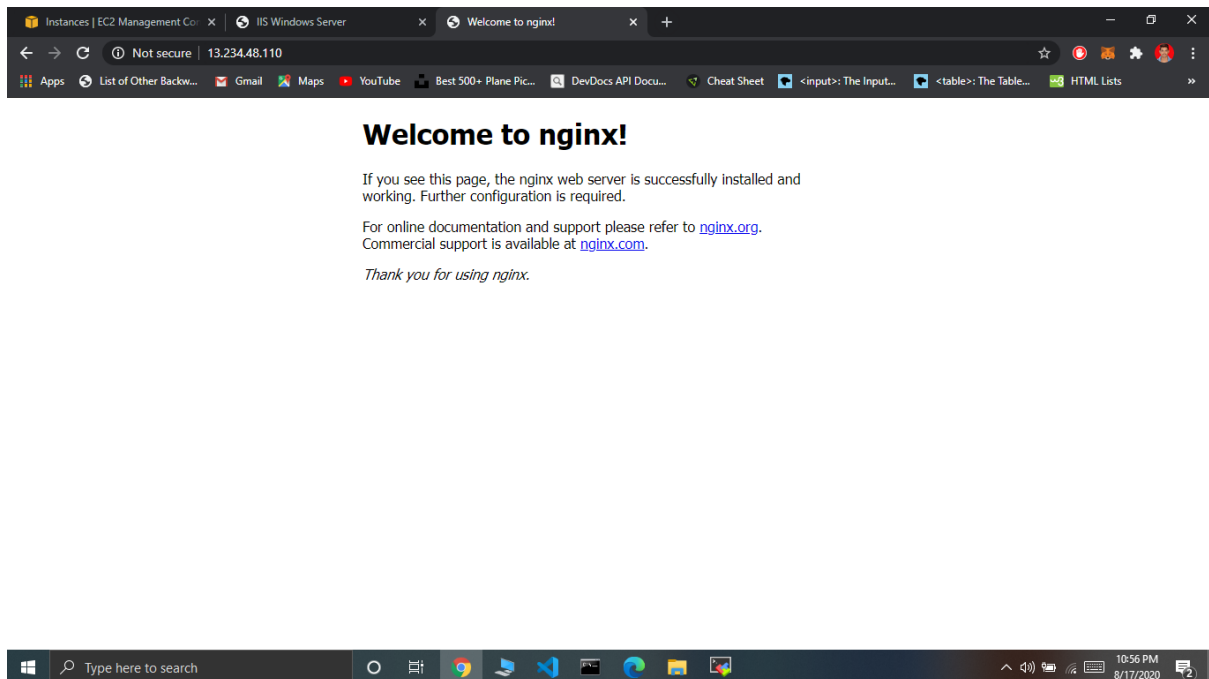




Step 17: After installation webserver will be deployed and can be viewed by public IP address available on the EC2 instances list. -> Copy the IP Address and paste and open it in the Browser.

Step 18: Webserver Deployed and Viewed.

With the Public IP address: 13.234.48.110



**TASK DONE.**

**MAKE SURE THAT THE INSTANCES CREATED ARE TERMINATED AFTER THE USAGE TO AVOID UNNECESSARY CHARGES.**