

LetsUpgrade - AWS Essentials-Batch 1-Day 3

Project Document

Name : Nantha Kumar Rajasekaren

Email Address : nanthacollege17@gmail.com

Phone Number : 8825862159

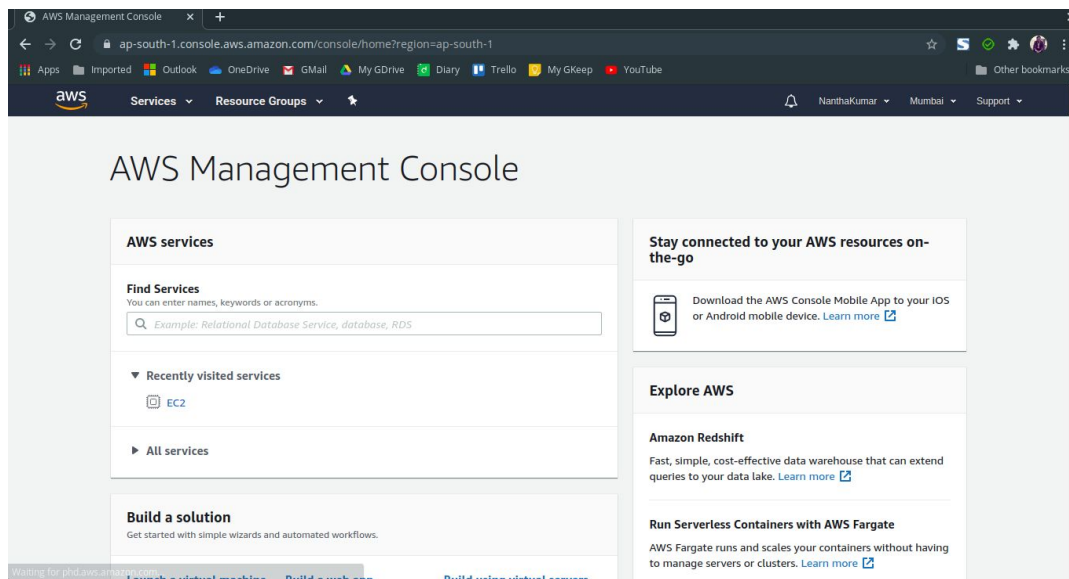
Project 1 : Deploy Web Server in Windows Instance

Project 2 : Deploy Web Server in Linux Instance

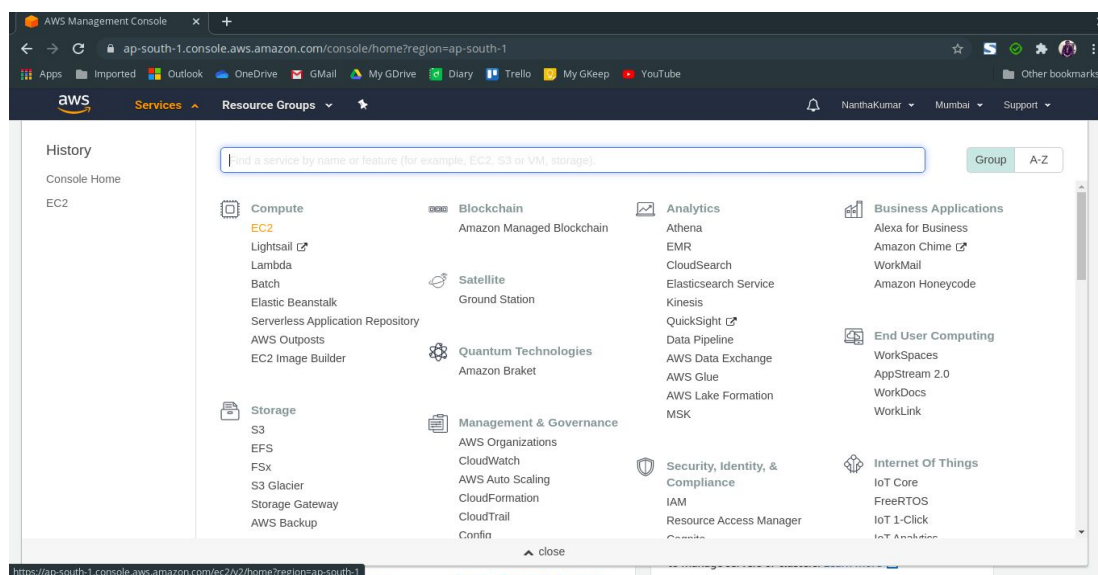
Project 1 : Deploy Web Server in Windows Instance

* Login into the AWS Console - <https://console.aws.amazon.com/console/home>

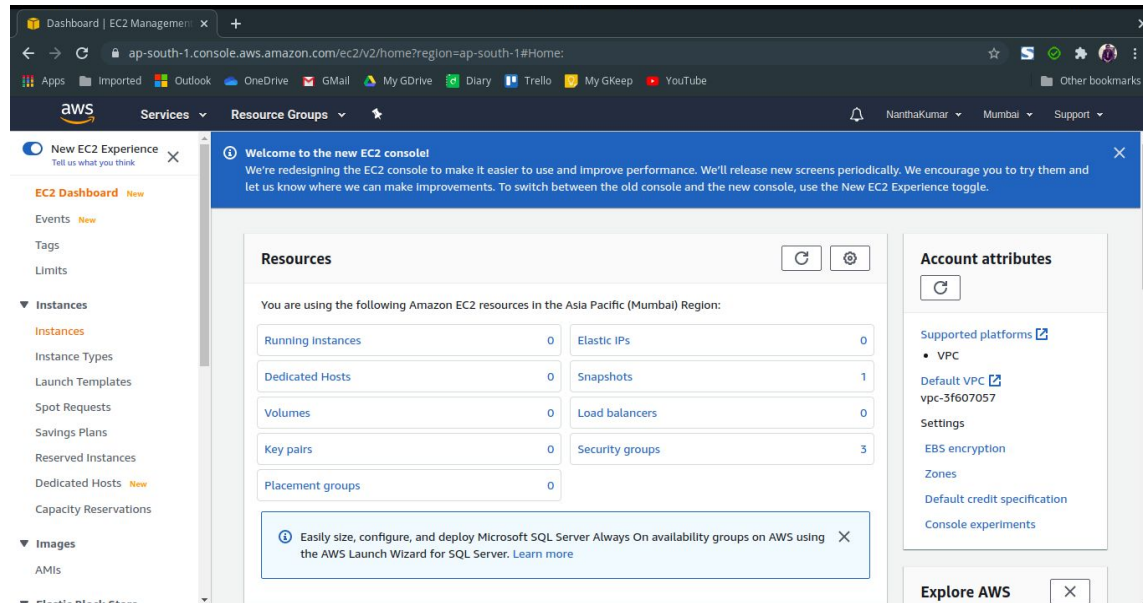
* After Login into it you will the home dashboard of the AWS Console



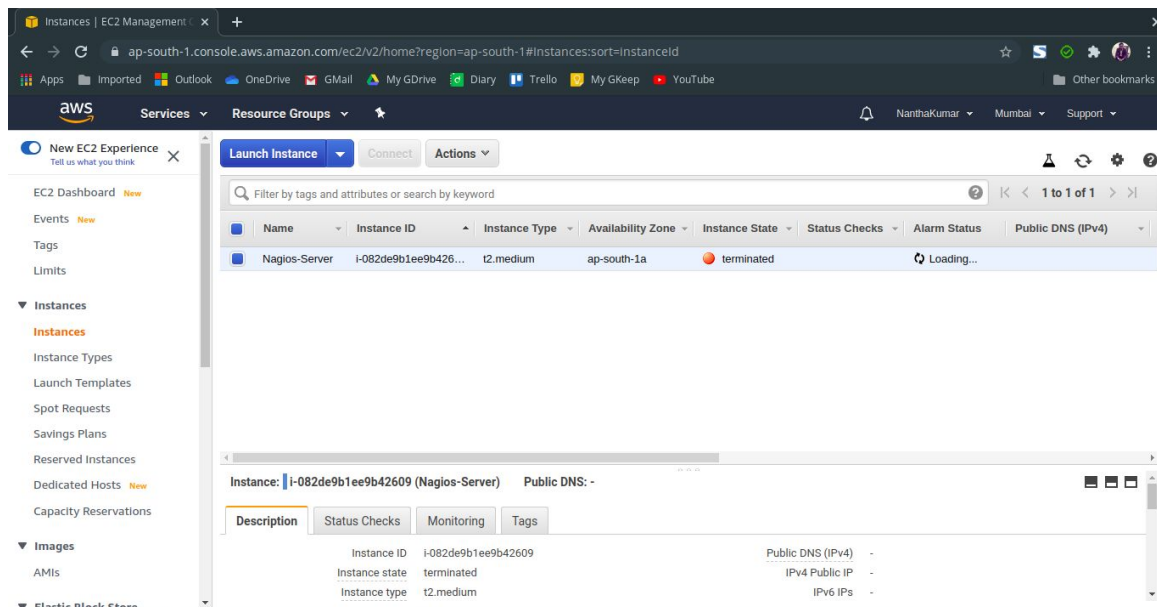
* Then click “Services” to list the AWS Cloud services and then select “EC2” from that



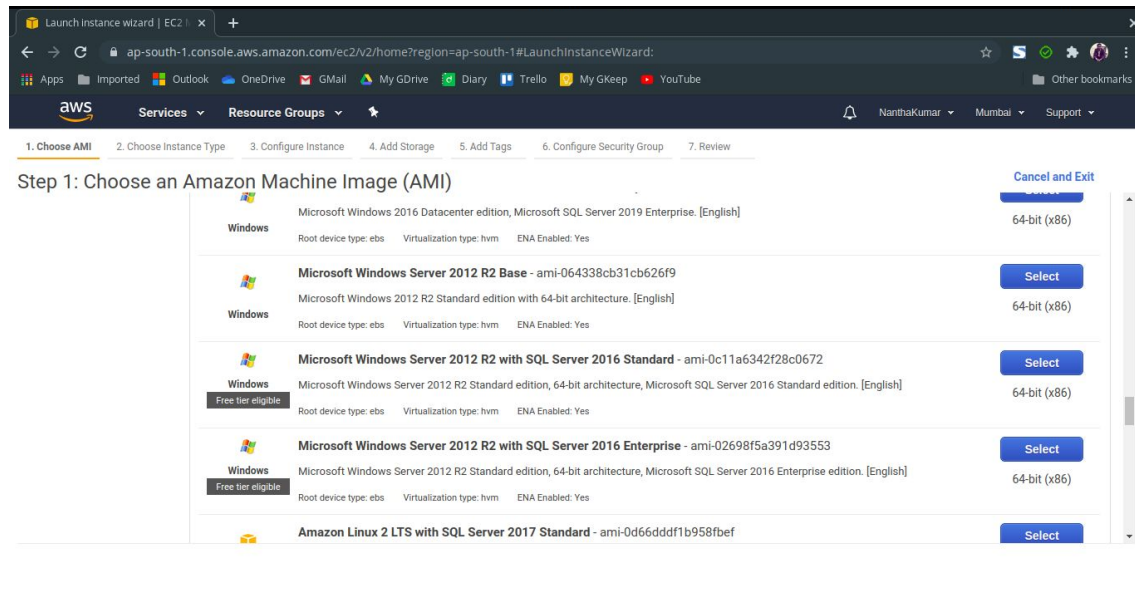
* Select Instance option from the left side panel of the appeared windows



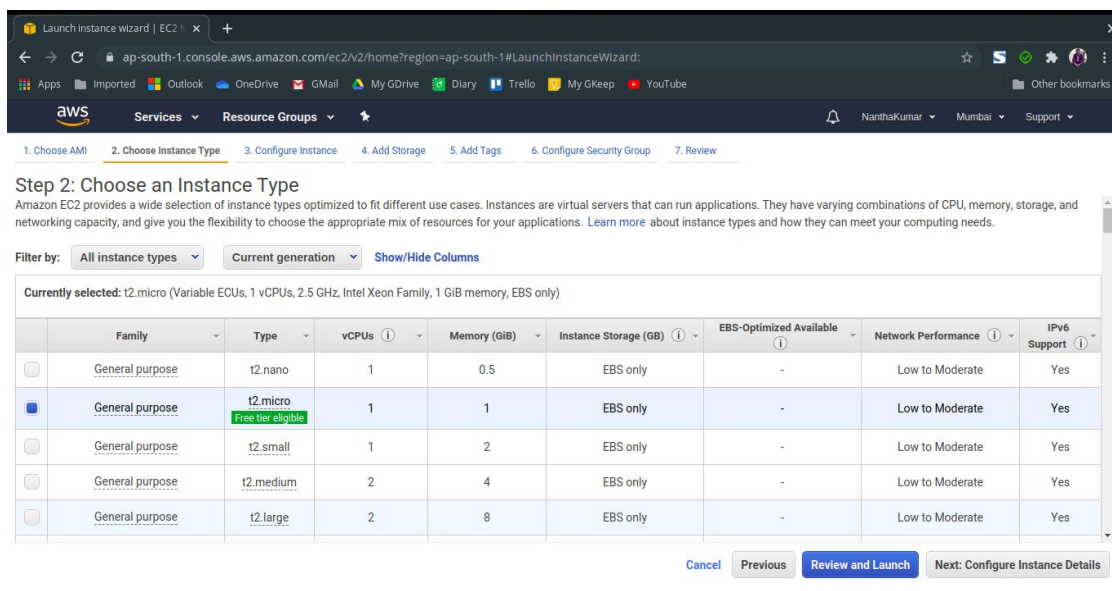
* Then select Launch Instance button from the appeared EC2 window



* Select Machine Image (AMI) from the list. I select the Windows Server 2012 R2 Base (HVM)



* Then select Instance Hardware Configuration from the appeared list. Hereby i select t2.micro, which is eligible for Free-Tier.



* After that configure the VPC and other system roles for the instance, here i configure this instance to launch under pre-exist configured VPC.

Launch instance wizard | EC2 | x +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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-0a16e7e8310a43c54 | Hive_Service [Create new VPC](#)

Subnet subnet-01bcf59c175c65587 | Hive_Service | ap-south-1 [Create new subnet](#)
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Enable)

Placement group ☐ Add instance to placement group

Capacity Reservation Open

Domain join directory No directory [Create new directory](#)

Cancel Previous **Review and Launch** Next: Add Storage

* Then configure the storage for the instance. Until 30 GB of storage is eligible for free-tier users. For Windows Instance it is necessary to configure 30 GB of EBS Storage.

Launch instance wizard | EC2 | x +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0e268d737d09862b5	30	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypt

[Add New Volume](#)

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

Feedback English (US) © 2008 - 2020, Amazon Internet Services Private Ltd. or its affiliates. All rights reserved. Privacy Policy Terms of Use

* Then add the tag value for the instance for identification.

Launch instance wizard | EC2 | x

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum)	Value (256 characters maximum)	Instances (1)	Volumes (1)
<div> <div>⚠ A tag key must be between 1 and 127 characters in length</div> <div> <input type="text"/> <input type="text"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="button" value="x"/> </div> </div>			
<div> <div>Add another tag</div> <div>(Up to 50 tags maximum)</div> </div>			

Cancel Previous **Review and Launch** Next: Configure Security Group

* Then configure the security group for the instance, which acts as a basic protection like firewall for the server (or) instance from the cyber attack. Here i configured my pre-exist secured group, which is associated with my VPC.

Launch instance wizard | EC2 | x

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

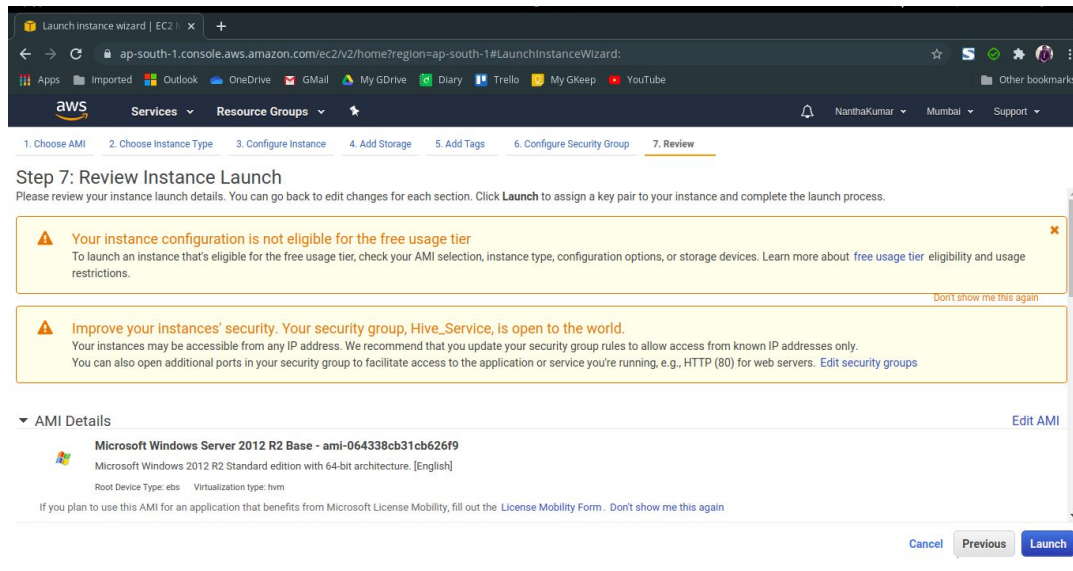
Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Security Group ID	Name	Description	Actions
sg-0969b85e19f22d997	default	default VPC security group	Copy to new
sg-0ede0c80219437d1a	Hive_Service	Hive R&D Security Group	Copy to new

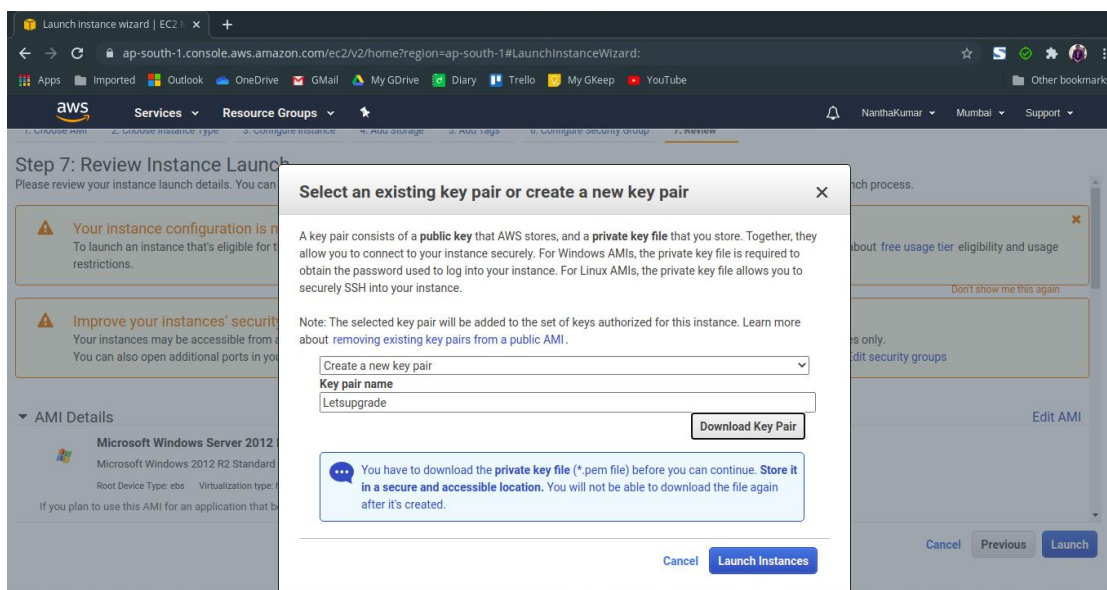
Custom TCP Rule	TCP	40000 - 50000	0.0.0.0/0	FTP access with se...
HTTPS	TCP	443	0.0.0.0/0	Web Service delive...
HTTPS	TCP	443	0.0.0.0/0	Web Service delive...
NFS	TCP	2049	0.0.0.0/0	NFS Storage Mount ...
NFS	TCP	2049	0.0.0.0/0	NFS Storage Mount ...

Cancel Previous **Review and Launch**

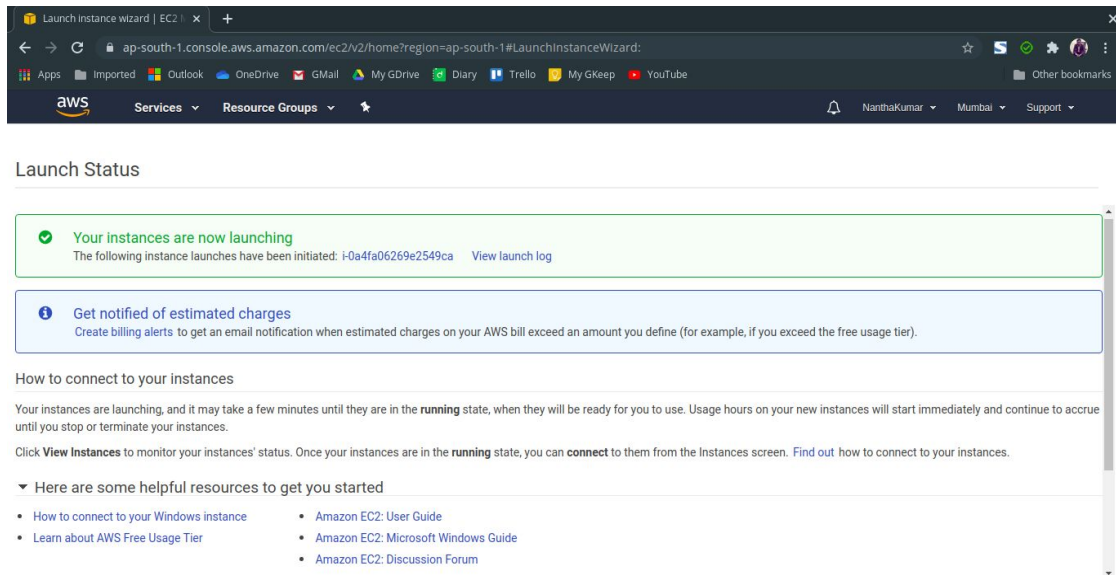
* Then select the option to review the instance configuration and launch it.



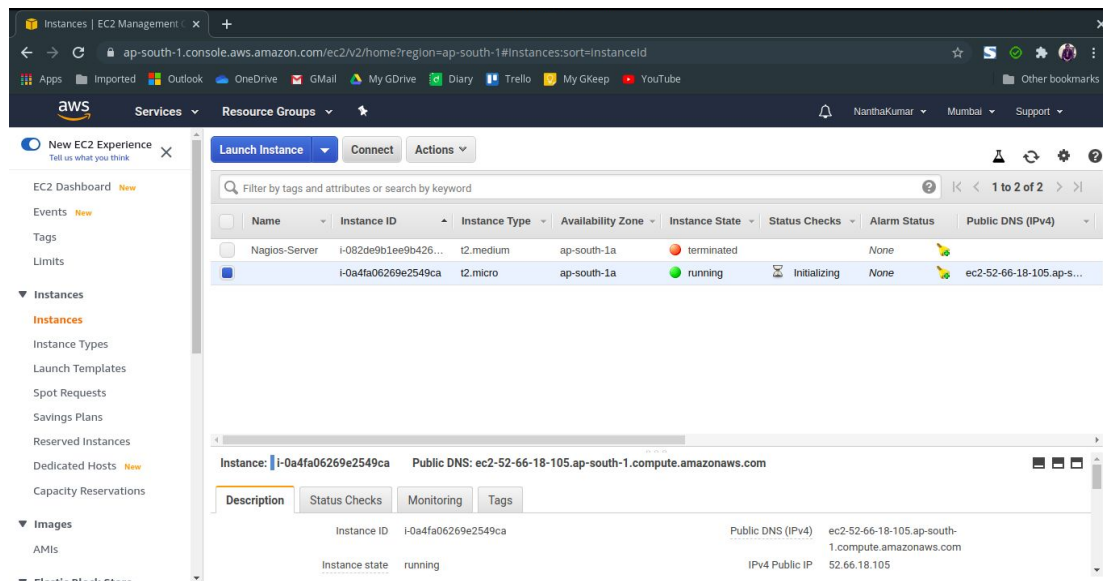
* Then select the launch option to launch the configured instance.



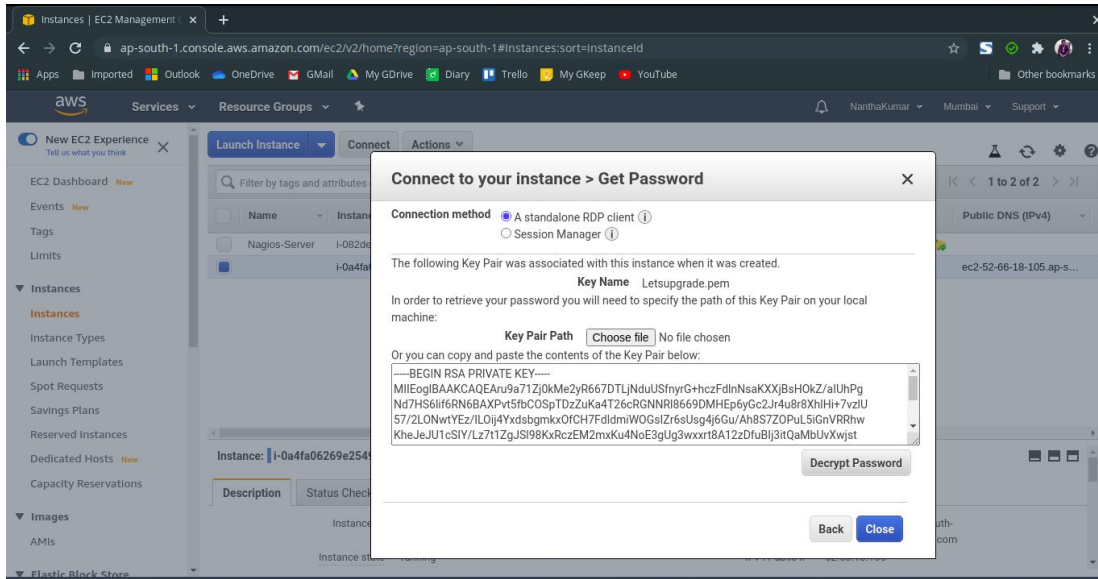
* Create a Key file (.pem), which holds up the encrypted password to access the created server in AWS.



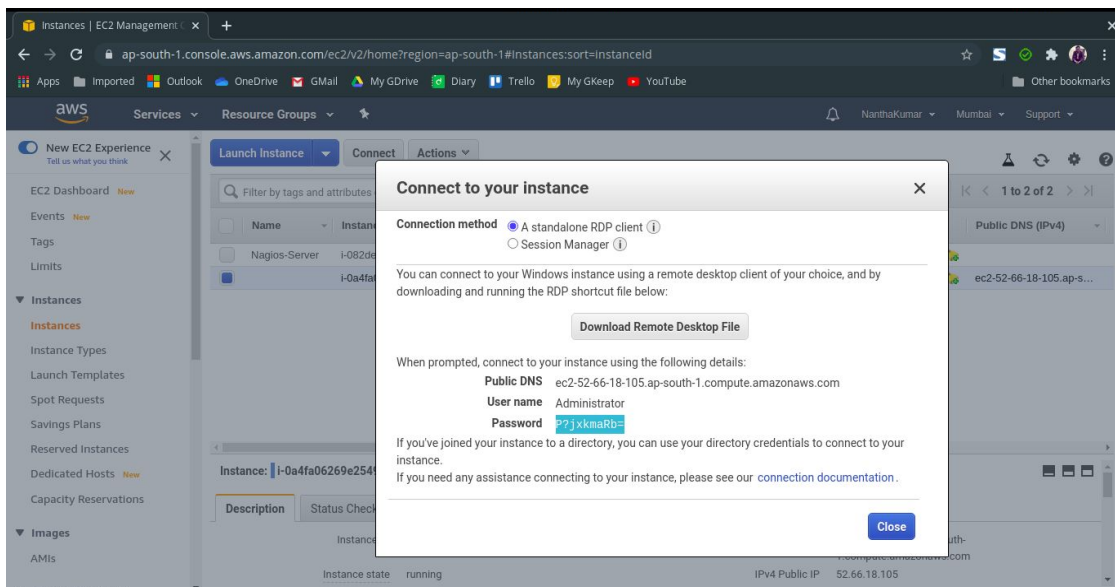
* After all the process is done successfully, Wait for some times to access the server until the status check completely.



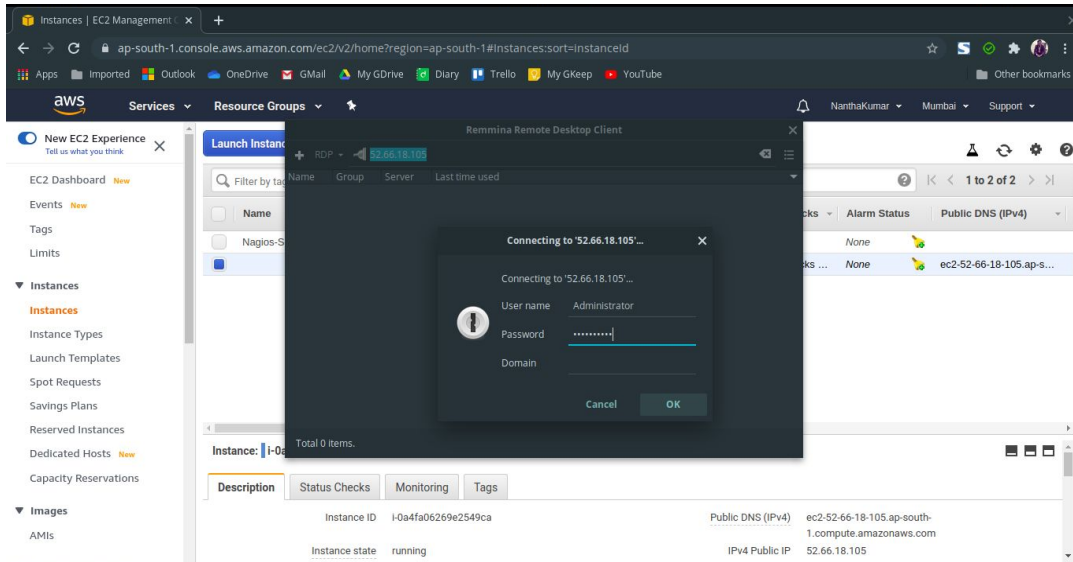
* Then select the connect button to decrypt the key file into the password to access the server.



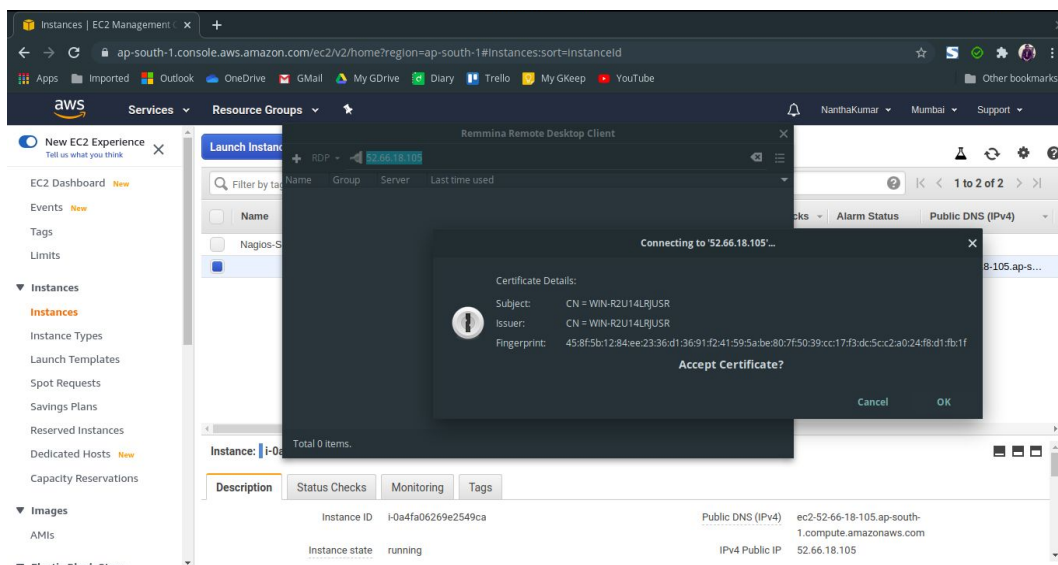
* After uploaded the key in the appeared panel, select the Decrypt button to achieve the password.

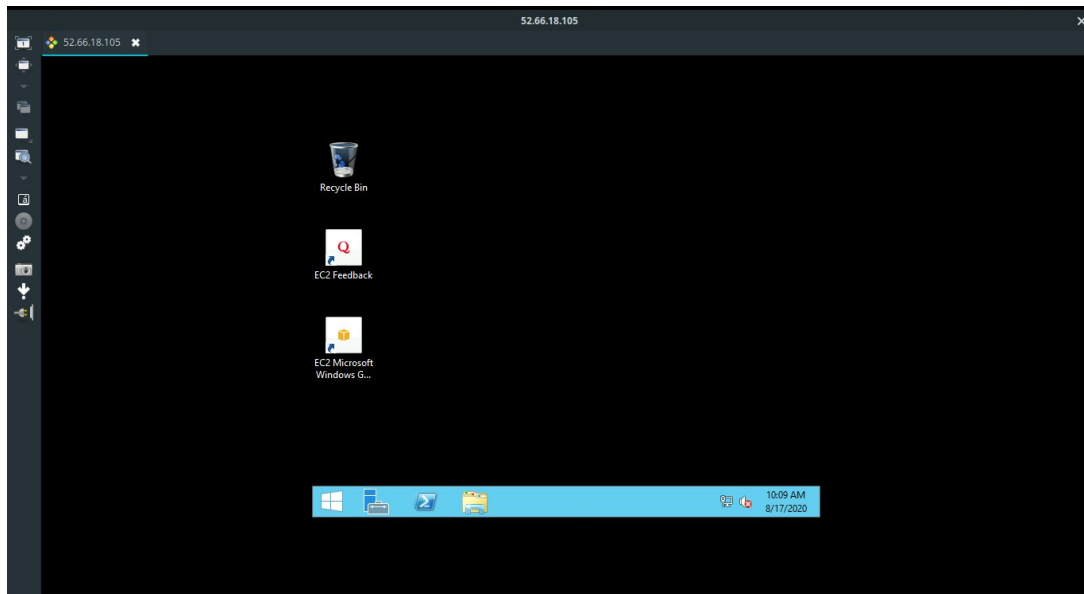


* There are a lot of applications using RDP connection, even AWS will provide a Microsoft RDP file for the server access. But hereby i'm using "remmina" for RDP access



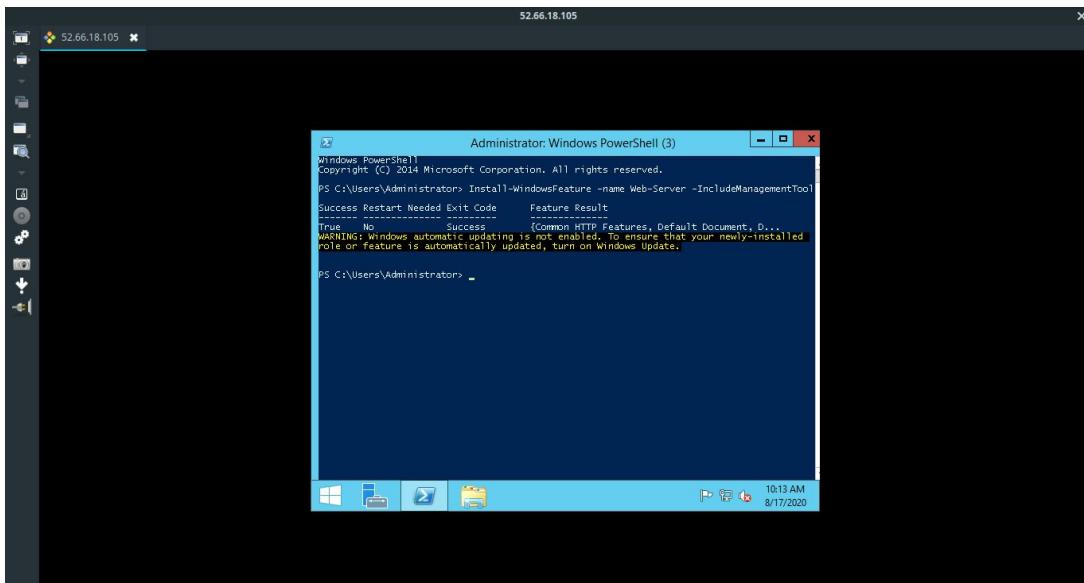
* Type decrypted password in the appeared panel to access the server



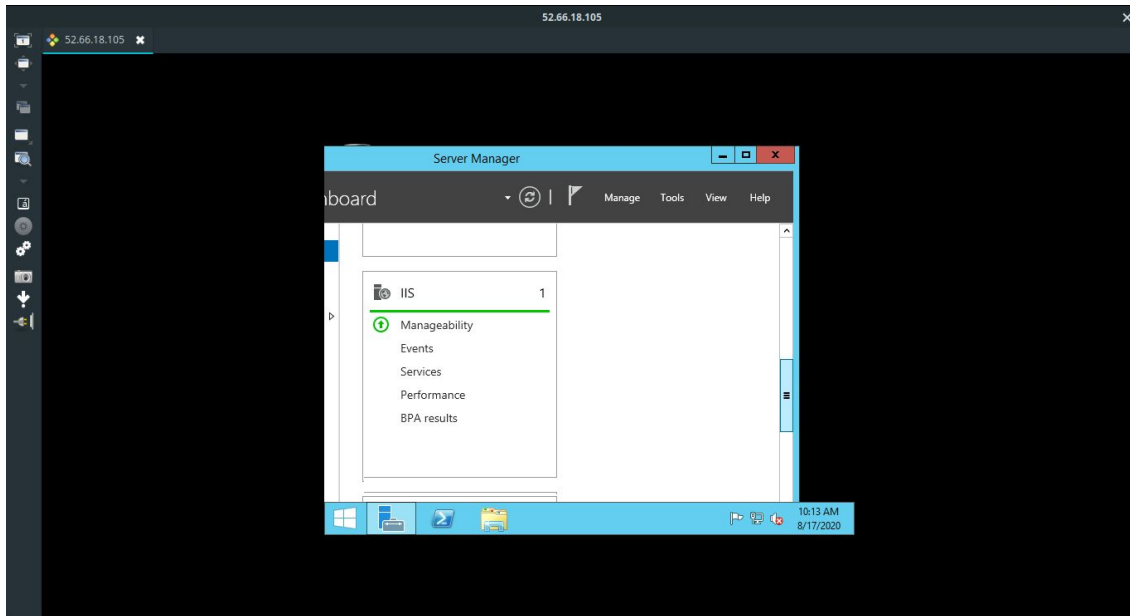


* After the remmina connects with the server, open the “powershell” in the task bar of the Windows and type the following command to install IIS-Internet Information Services.

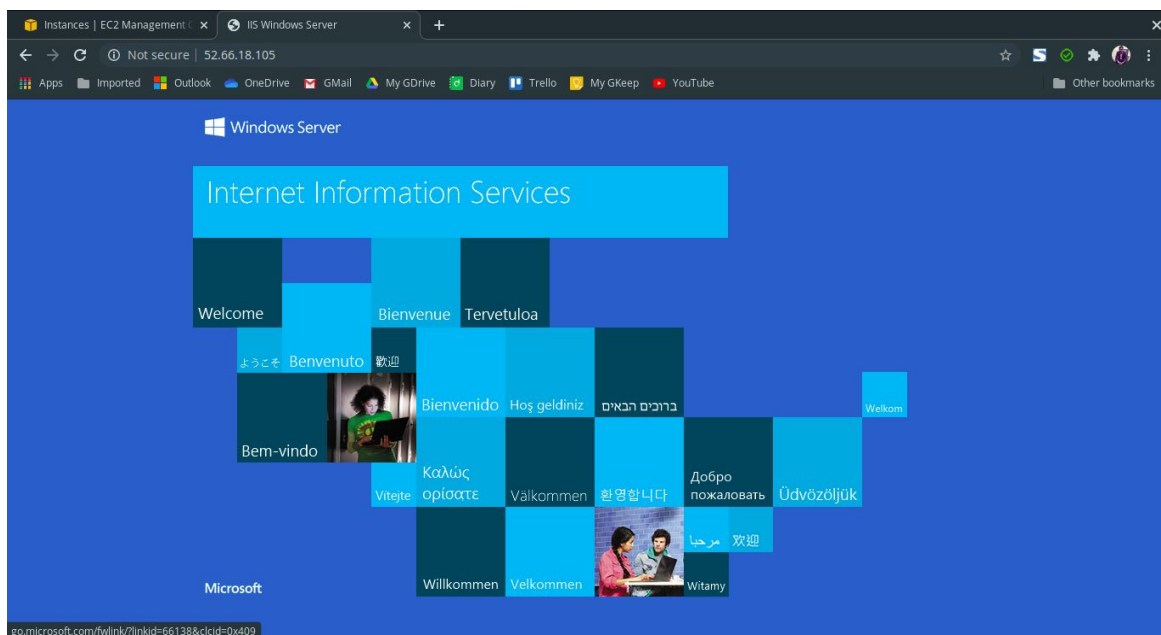
`Instance-WindowsFeature -name Web-Server -IncludeManagementTools`



* After it installs the IIS successfully, open the server manager to ensure that the IIS is running properly.



* Achieve the public IP address of the server from the AWS-EC2 console and point the IP address in the browser.

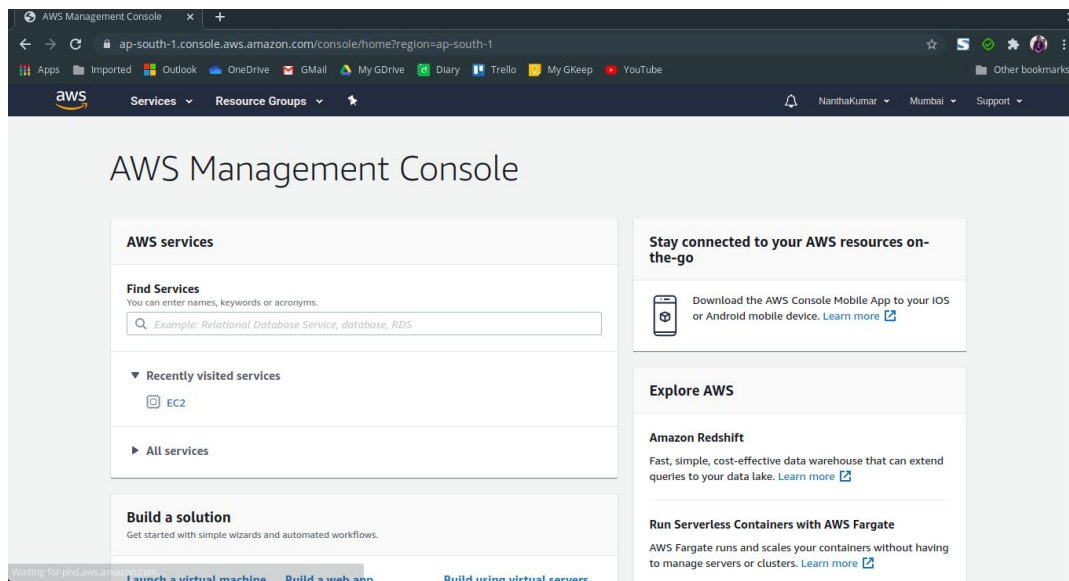


Thus Project-1 Successfully completed.

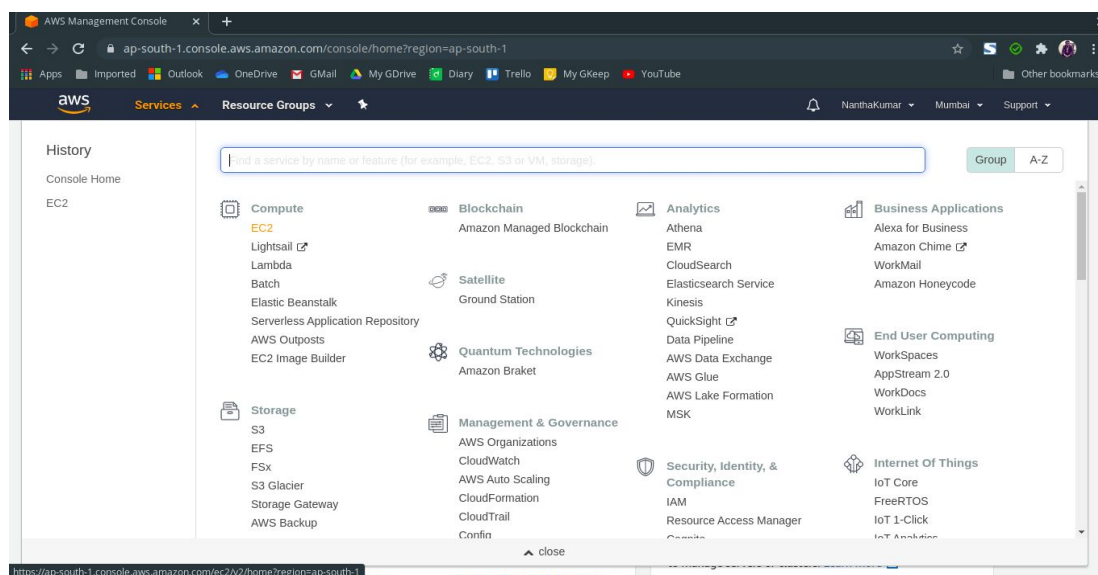
Project 2 : Deploy Web Server in Linux Instance

* Login into the AWS Console - <https://console.aws.amazon.com/console/home>

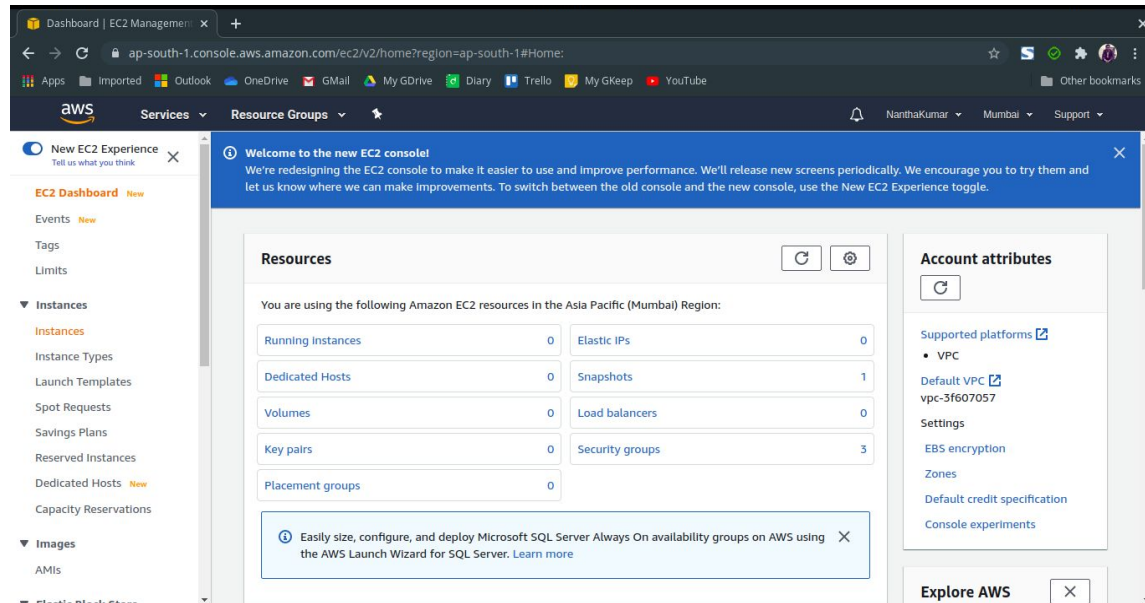
* After Login into it you will the home dashboard of the AWS Console



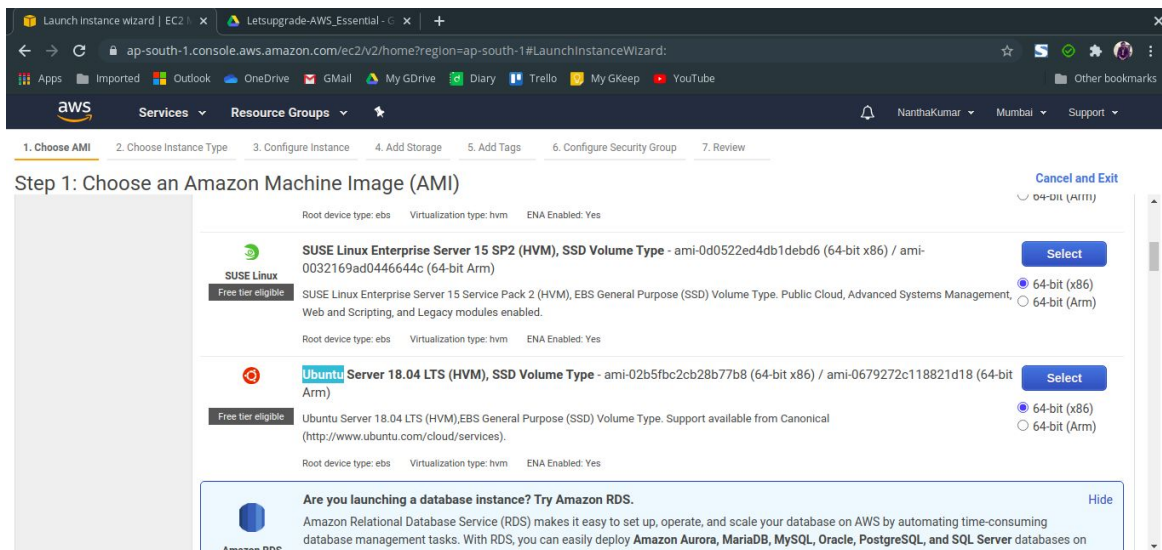
* Then click “Services” to list the AWS Cloud services and then select “EC2” from that



* Select Instance option from the left side panel of the appeared windows

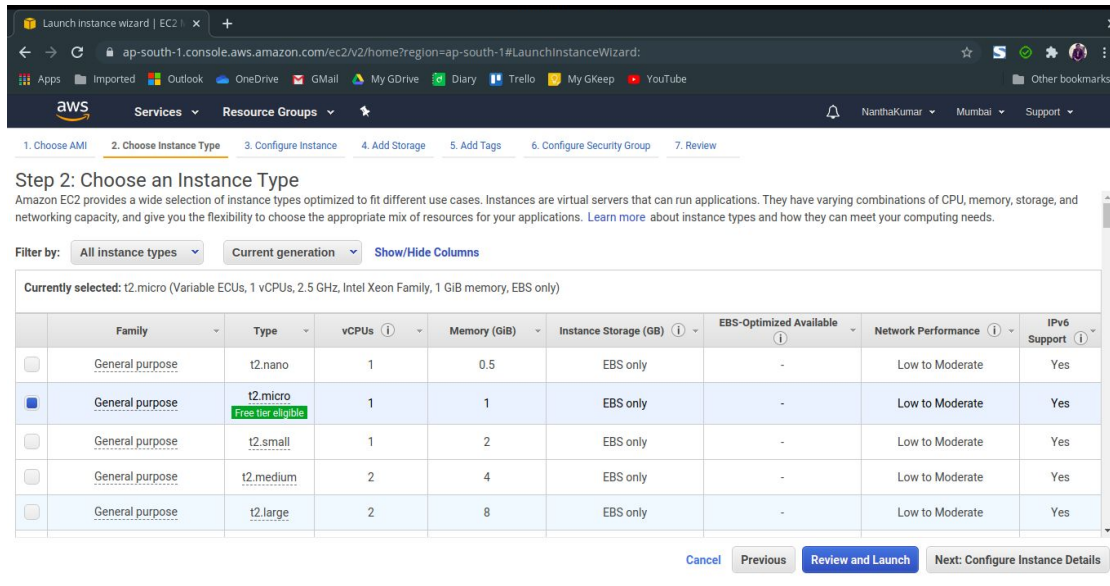


* Then select Launch Instance button from the appeared EC2 window



* Select Machine Image (AMI) from the list. I select the Ubuntu Server 16.04 LTS (HVM).

* Then select Instance Hardware Configuration from the appeared list. Hereby I select t2.micro, which is eligible for Free-Tier.



Launch instance wizard | EC2 | x

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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.

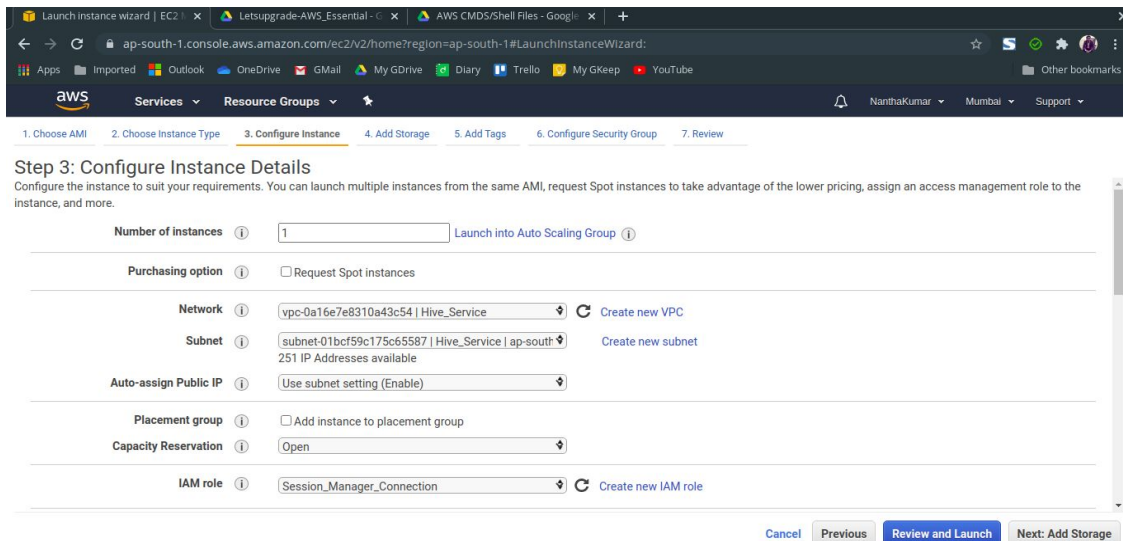
Filter by: All instance types Current generation Show/Hide Columns

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 (GiB)	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

* After that configure the VPC and other system roles for the instance, here i configure this instance to launch under pre-exist configured VPC.



Launch instance wizard | EC2 | x Letsupgrade-AWS_Essential - Google AWS CMD/Shell Files - Google

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

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-0a16e7e8310a43c54 | Hive_Service Create new VPC

Subnet subnet-01bcf59c175c65587 | Hive_Service | ap-south-1 Create new subnet
251 IP Addresses available

Auto-assign Public IP Use subnet setting (Enable)

Placement group ☐ Add instance to placement group

Capacity Reservation Open

IAM role Session_Manager_Connection Create new IAM role

Cancel Previous Review and Launch Next: Add Storage

* Along with the VPC, i associate the “SystemManager” IAM role for my instance to access the server through SSH connection instead of using Mobaxterm or some other

Launch instance wizard | EC2 | x Letsupgrade-AWS_Essential - C x AWS CMD/Shell Files - Google x +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 3: Configure Instance Details

Device	Network Interface	Subnet	Primary IP	Secondary IP addresses	IPv6 IPs
eth0	New network interface	subnet-01bcf59c	Auto-assign	Add IP	Add IP

Add Device

Advanced Details

Metadata accessible ☒ Enabled

Metadata version ☐ V1 and V2 (token optional)

Metadata token response hop limit ☐ 1

User data ☒ As text ☐ As file ☐ Input is already base64 encoded

```
#!/bin/sh
sudo apt-get update && apt-get upgrade -y
sudo apt-get install ec2-instance-connect
sudo less /lib/systemd/system/ssh.service.d/ec2-instance-connect.conf
```

Cancel Previous Review and Launch Next: Add Storage

* Then configure the storage for the instance. Until 30 GB of storage is eligible for free-tier users. For Linux Instance it is necessary to configure 8 GB of EBS Storage.

Launch instance wizard | EC2 | x Letsupgrade-AWS_Essential - C x AWS CMD/Shell Files - Google x +

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

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

Add New Volume

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

* Then add the tag value for the instance for identification.

Launch instance wizard | EC2 | x

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 5: Add Tags

A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver.
A copy of a tag can be applied to volumes, instances or both.
Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (128 characters maximum) Value (256 characters maximum) Instances 1 Volumes 1

A tag key must be between 1 and 127 characters in length

Add another tag (Up to 50 tags maximum)

Cancel Previous Review and Launch Next: Configure Security Group

* Then configure the security group for the instance, which acts as a basic protection like firewall for the server (or) instance from the cyber attack. Here I configured my pre-exist secured group, which is associated with my VPC.

Launch instance wizard | EC2 | x

ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#LaunchInstanceWizard:

Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

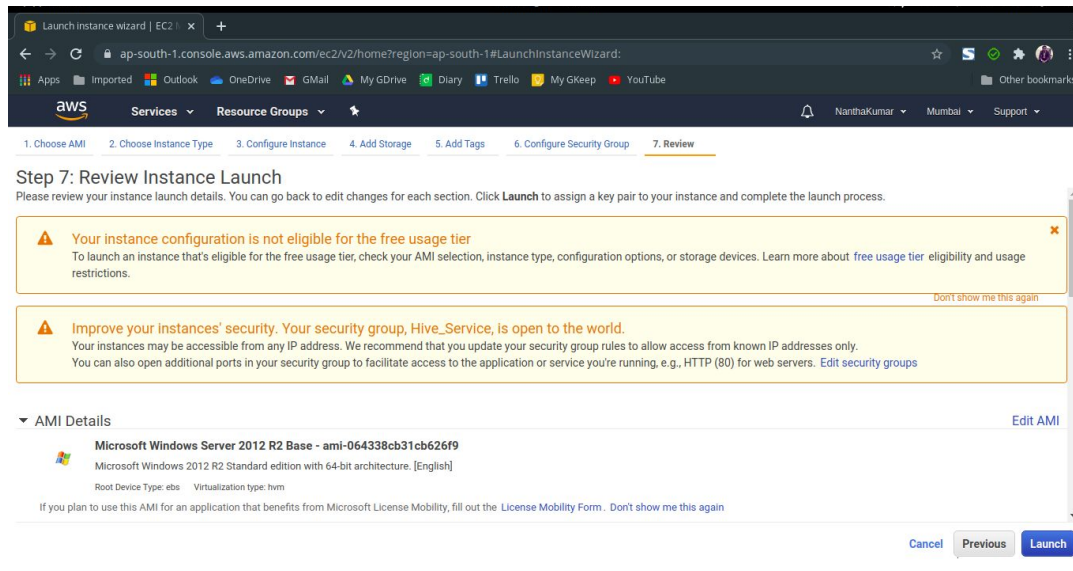
Assign a security group: ☐ Create a new security group ☒ Select an existing security group

Security Group ID	Name	Description	Actions
sg-0969b85e19f22d997	default	default VPC security group	Copy to new
sg-0ede0c80219437d1a	Hive_Service	Hive R&D Security Group	Copy to new

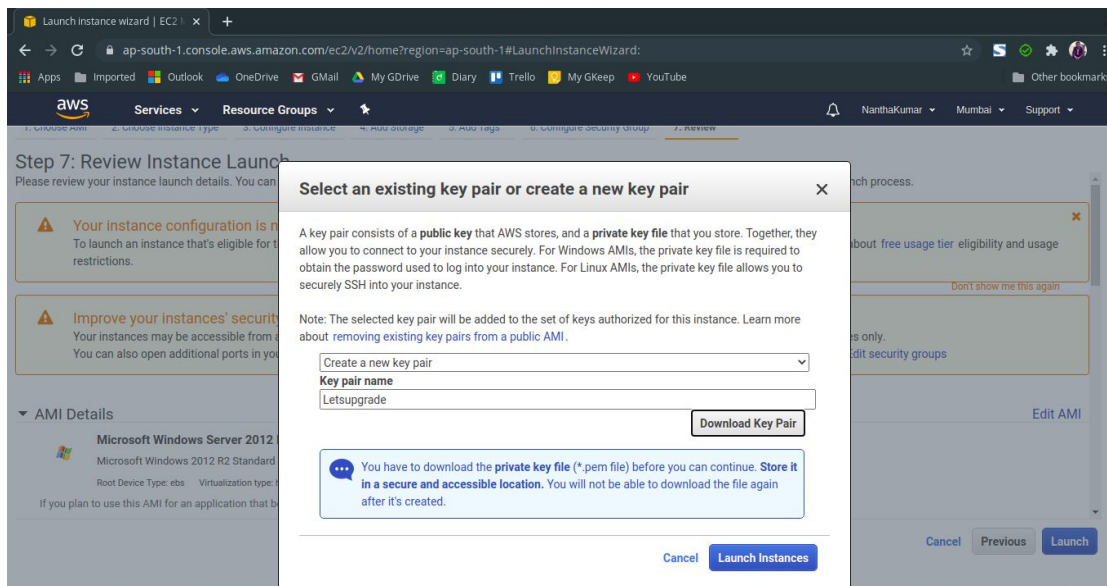
Rule Name	Protocol	Port Range	Source/Destination	Action
Custom TCP Rule	TCP	40000 - 50000	::/0	FTP access with se...
HTTPS	TCP	443	0.0.0.0/0	Web Service delive...
HTTPS	TCP	443	::/0	Web Service delive...
NFS	TCP	2049	0.0.0.0/0	NFS Storage Mount ...
NFS	TCP	2049	::/0	NFS Storage Mount ...

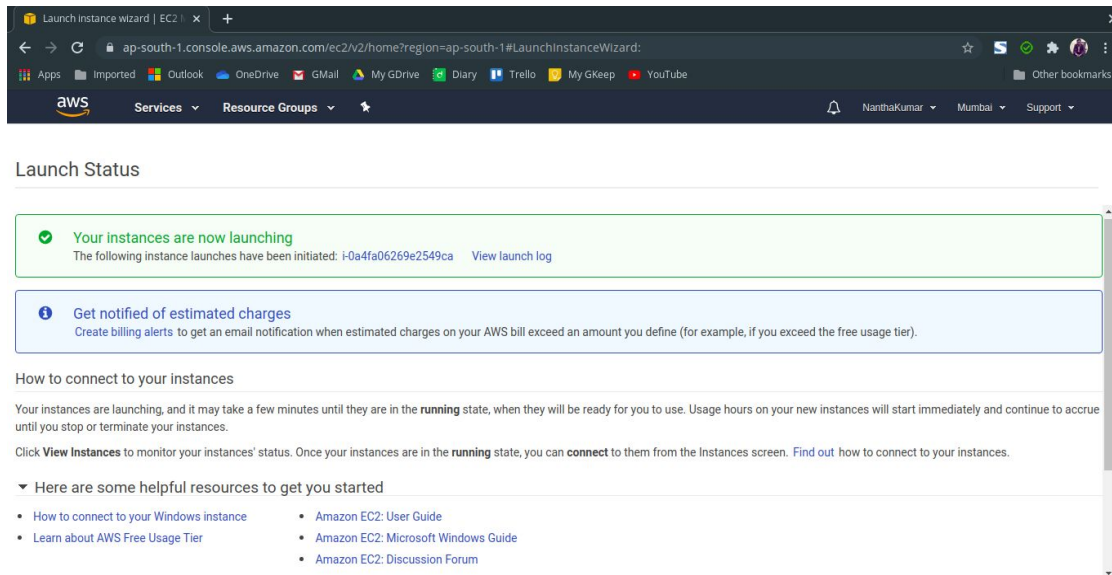
Cancel Previous Review and Launch

* Then select the option to review the instance configuration and launch it.

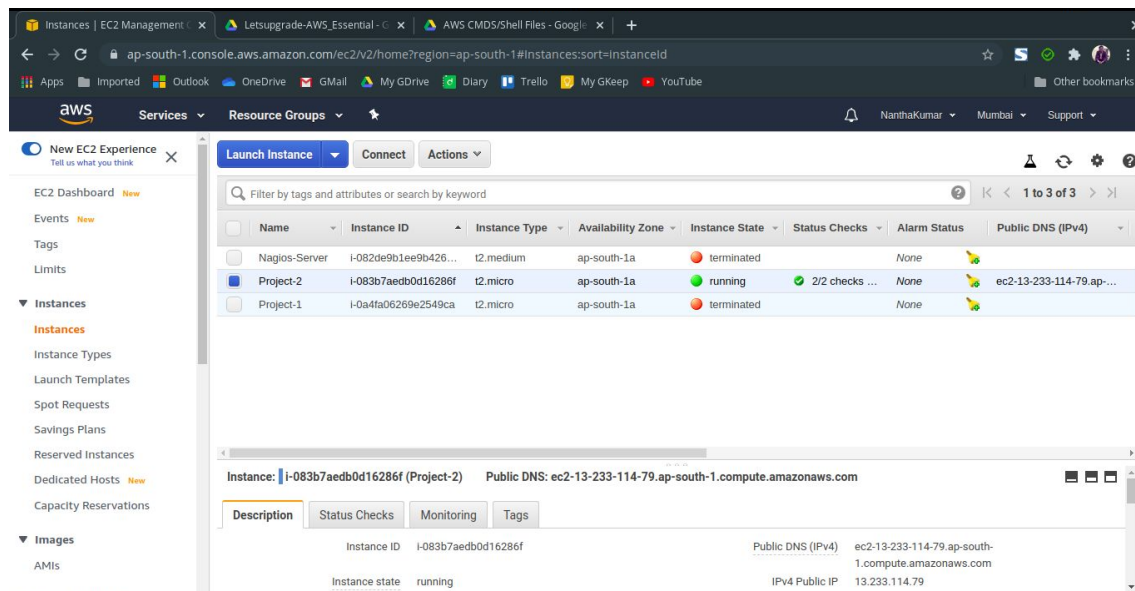


* Then select the launch option to launch the configured instance. Create a Key file (.pem), which holds up the encrypted password to access the created server in AWS.

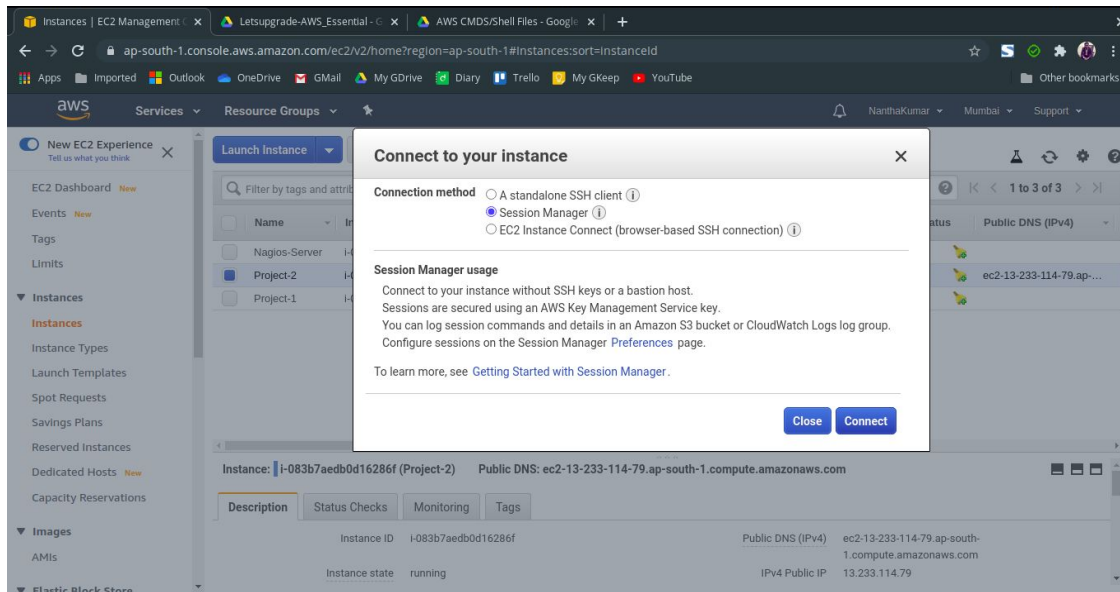




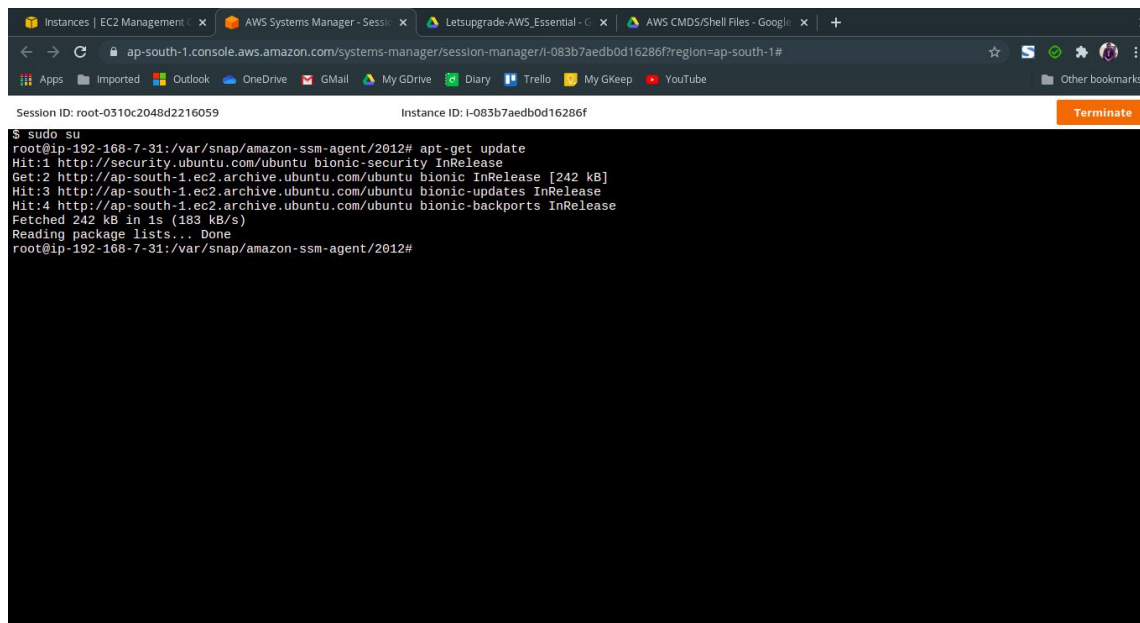
* After all the process is done successfully, Wait for some time to access the server until the status check completely.



* Then select the connect button to access the server.



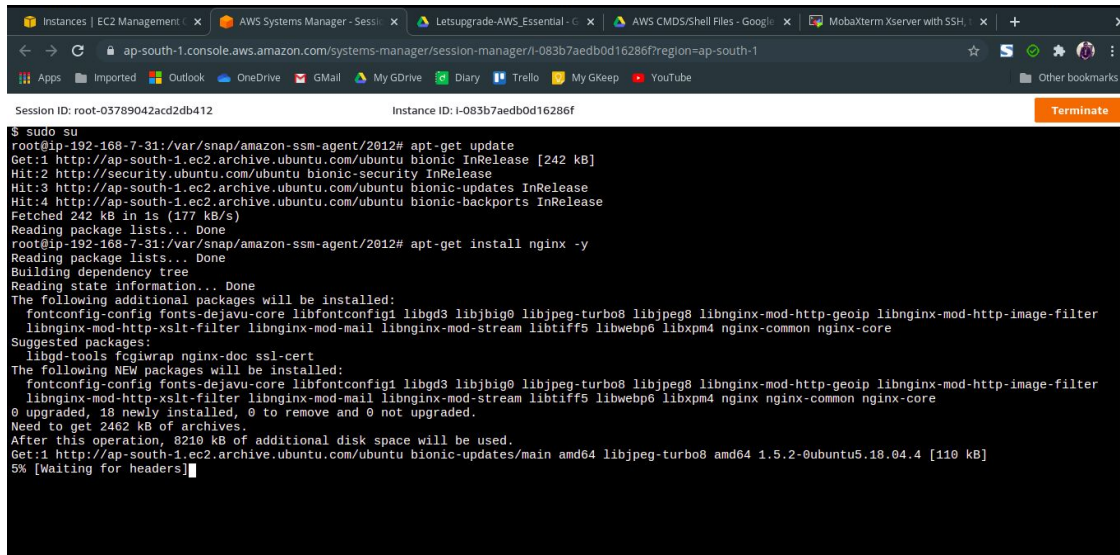
* Here I'm using “Session Manager” to connect with the server through SSH instead of other 3rd party applications for the security reason.



* After session manager opens, type the following command to install and configure web server (Nginx).

apt-get update (To update the system packages)

apt-get install nginx -y (To install the Nginx web server service)

A screenshot of a terminal window running on an AWS EC2 instance. The terminal shows the execution of 'sudo su' to become root, followed by 'apt-get update' which updates the package lists. Then, 'apt-get install nginx -y' is executed, which installs Nginx and its dependencies. The output shows the packages to be installed, the disk space requirements, and the progress of the installation. The terminal window is titled 'Instances | EC2 Management' and shows the instance ID 'i-083b7aedb0d16286f' and the session ID 'root-03789042acd2db412'. There is a 'Terminate' button in the top right corner of the terminal window.

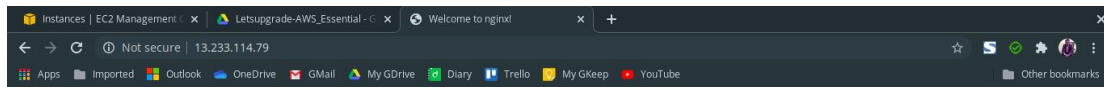
```
$ sudo su
root@ip-192-168-7-31:/var/snap/amazon-ssm-agent/2012# apt-get update
Get:1 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic InRelease [242 kB]
Hit:2 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:3 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-updates InRelease
Hit:4 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu bionic-backports InRelease
Fetched 242 kB in 1s (177 kB/s)
Reading package lists... Done
root@ip-192-168-7-31:/var/snap/amazon-ssm-agent/2012# apt-get install nginx -y
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 libjbig0 libjpeg-turbo8 libjpeg8 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 libjbig0 libjpeg-turbo8 libjpeg8 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 0 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]
5% [Waiting for headers]
```

After it installed successfully. Start the service to run always.

systemctl enable nginx

systemctl start nginx

After it runs successfully, then trap the public IP address of the server from the AWS-EC2 console and the point the browser to that IP address.



Thus Project-2 Successfully completed.