

EC2 : Amazon EC2 (Elastic Compute Cloud) is a core service in Amazon Web Services (AWS) that provides scalable virtual servers (instances) in the cloud. It allows you to run applications on virtual machines with flexible compute capacity, meaning you can increase or decrease resources as needed.

### Key Features of EC2:

- **Scalability:** Easily scale up or down based on your workload.
- **Variety of Instance Types:** Optimized for different tasks like general-purpose, compute-optimized, memory-optimized, GPU instances, etc.
- **Customizable:** You can choose the operating system (Linux, Windows), storage, networking, and security settings.
- **Elastic Load Balancing:** Distributes traffic across multiple instances for better performance and availability.
- **Auto Scaling:** Automatically adjusts capacity based on demand.

Fastest way to create server and deployment during POC

Step 1. Create an instance and select ami (amazon machine image), simply search deep learning you will get an appropriate ami .

Step 2. Select key pair and security group and storage

Step 3. Launch the instance.

Step 4. Now go to security and edit security group.

Step 5. You can now edit inbound rules. If you are using streamlit add rule type: custom, port: 8501

Step 6. Connect to ssh client- go to option connect to ssh client.

Step 7. Copy the .pem file and paste it to .ssh folder, you will find it on "shriansh singh".

Step 8. Copy the command that you will find on connect to ssh client option.

Step 9. But if you try to connect with the server you won't be able to do that, let's solve that issue.

Step 10. Right click on .pem file and go to property->security->advanced-> disabled inheritance->remove everything from there->add->write your user name->select full control->ok->apply.

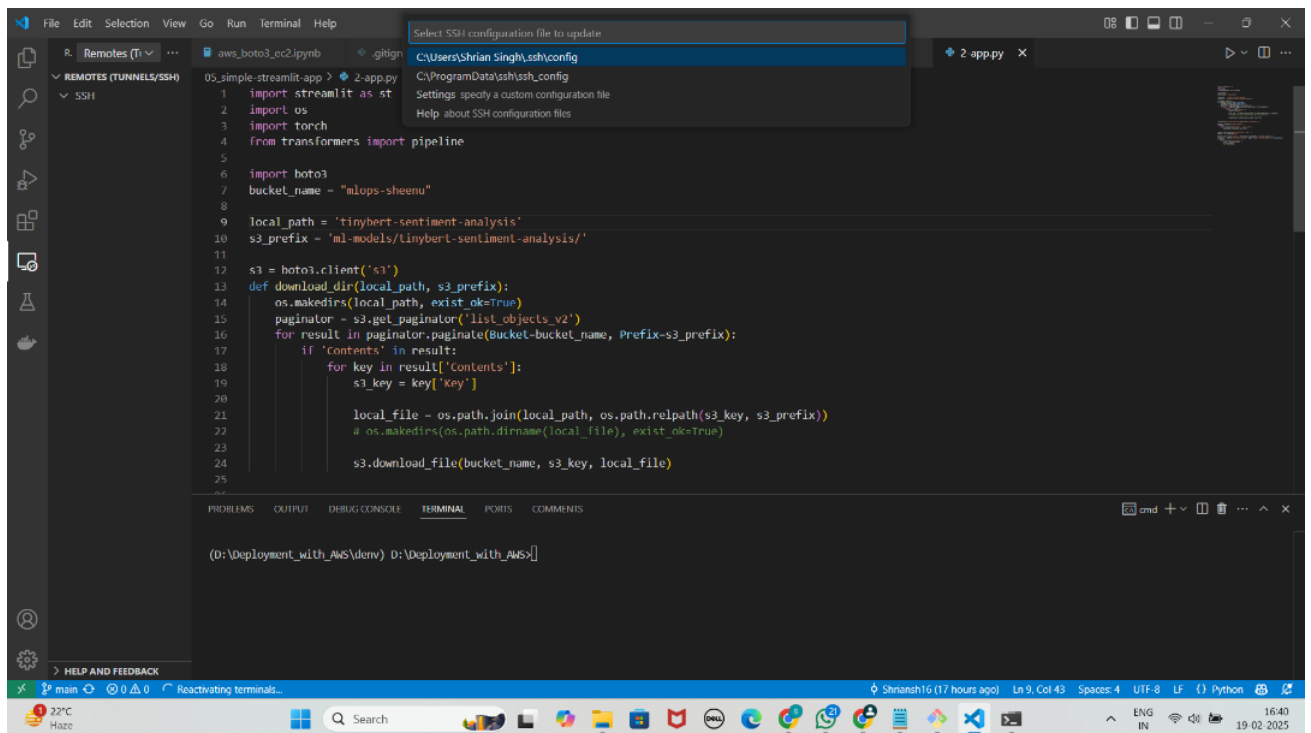
Step 11. In the command mention the entire path of .pem file

Step 12. Now we are connected to our server (instance).

Step 13. Now connect vscode with ec2.

Step 14. Go to your vscode and go to extension and search “Remote -SSH” install it if not installed.->now you will be able to see remote extension

Step 15. Click there->new remote (by clicking on + button) and paste the similar command that you used before for making connection.



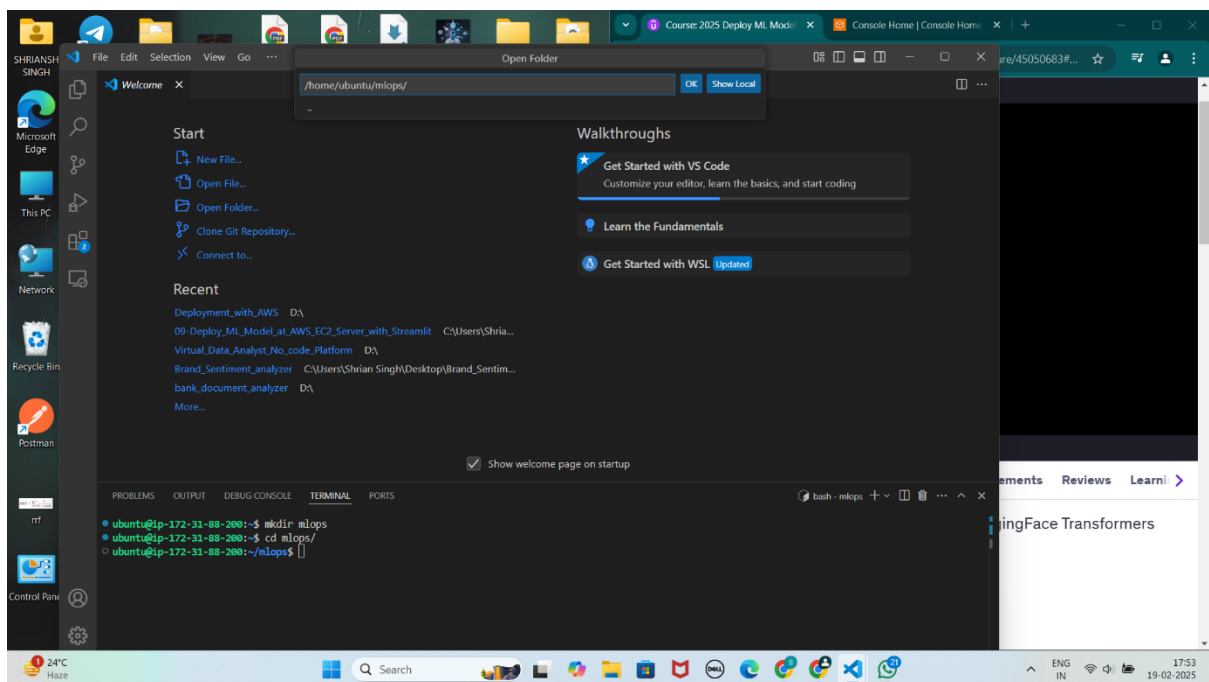
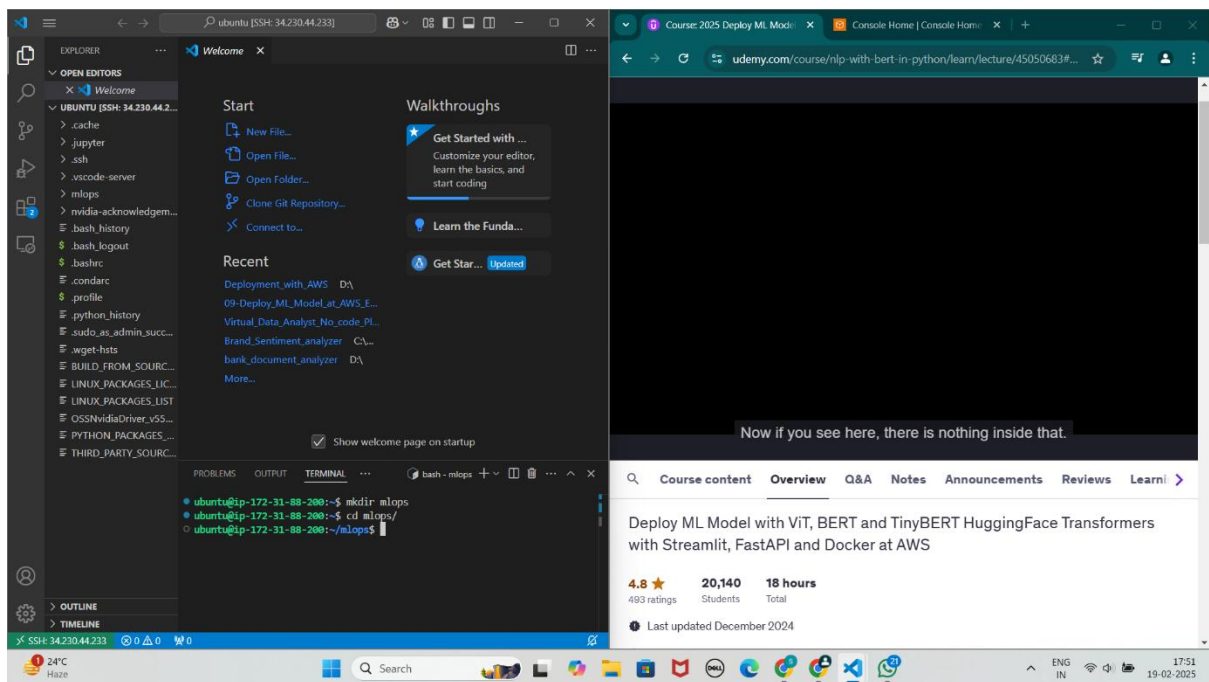
Step 16. Press enter and store in ssh config file.

Step 17. The new window will be opened and also select linux as a platform.

Step 18. Create a new directory- “mkdir name”

“cd name/”

Step 17. Open folder->name and then you will get an empty directory.

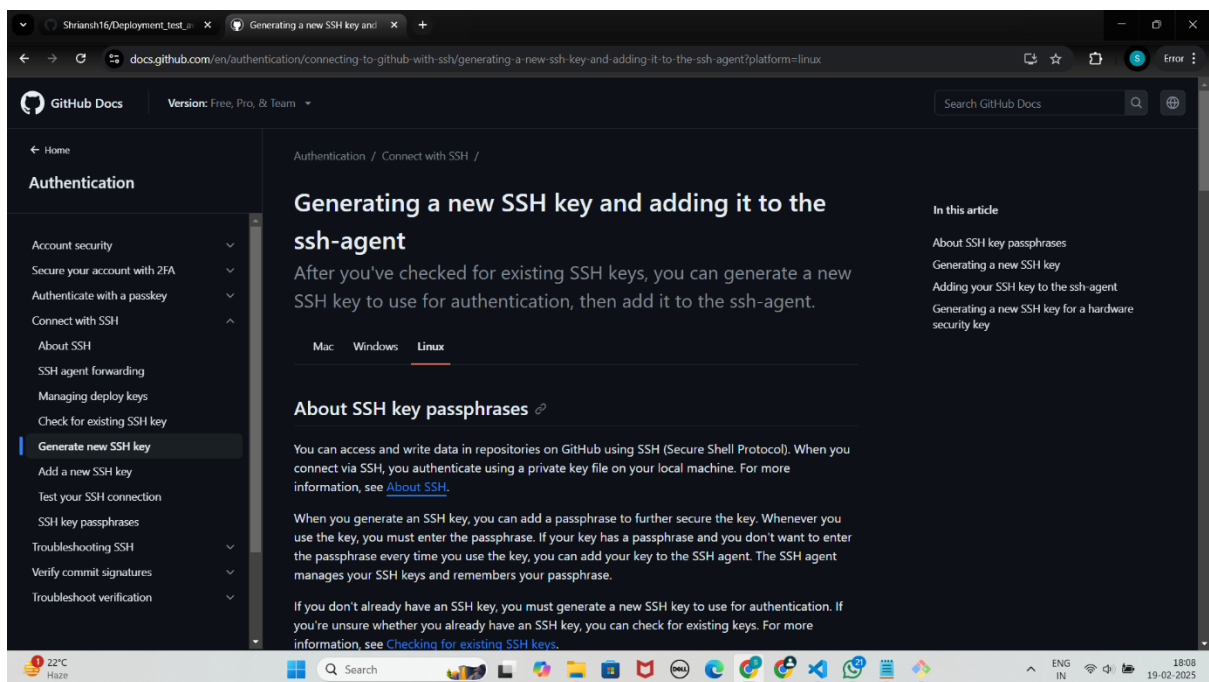
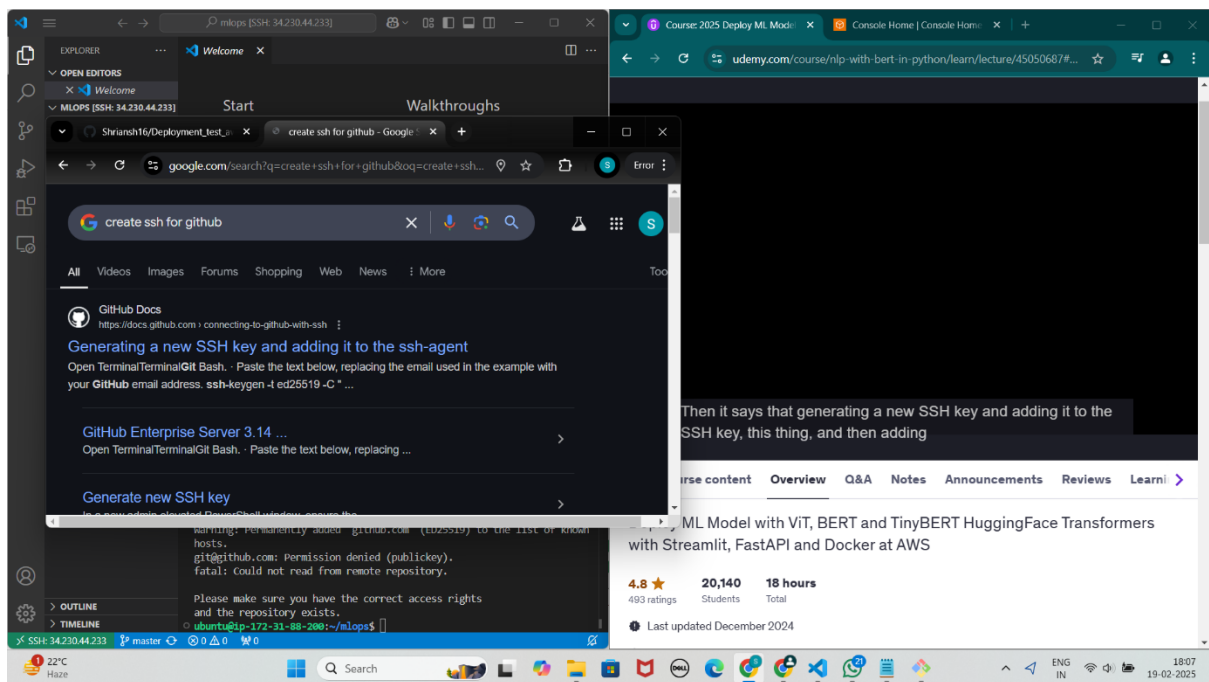


Step 18. Now you can create new files, folders according to your own requirements.

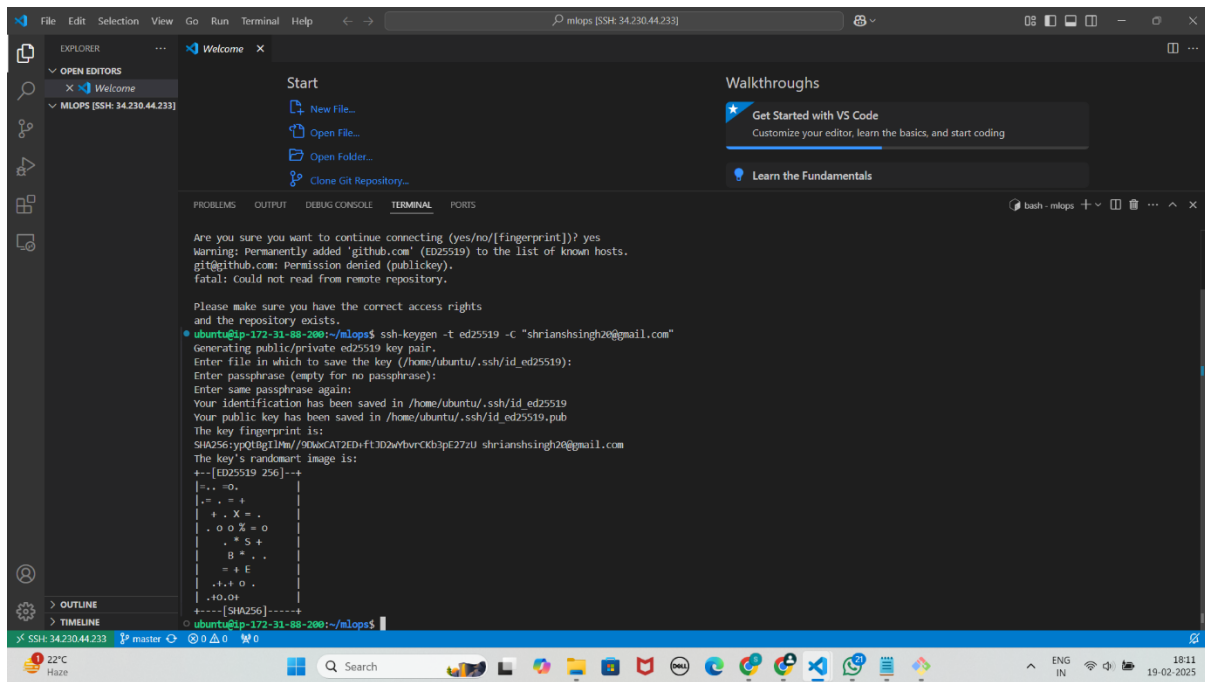
Step 19. Now lets clone the github repo.

Step 20. First connect your machine with your github account.

Step 21. Go to google and search “create ssh github”



Select linux and copy the given command don't forget to mention your email id on the command.



Then your ssh agent and also add your key  
you will get all the commands on the same page

Step 22. Go to your github account and setting ->ssh-> new key-> get your key using the command you will get on the previous website-> add it there

Step 23. Done, your github account has been connected.

Step 24. Simply clone the repo.

Step 25. Aws configure-> access key and secret access key and region (without "").  
(go to step 28 if you don't want to give credentials.)

```
++-[ED25519 256]--+
|..=0.
|..=+
|+.X=.
|.00% = 0
|.5+
|0+.
|=E
|+.0.
|+.0+
+---[SHA256]-----
ubuntu@ip-172-31-88-200:~/mlps$ eval "$(ssh-agent -s)"
Agent pid 3480
ubuntu@ip-172-31-88-200:~/mlps$ ssh-add ~/.ssh/id_ed25519
Identity added: /home/ubuntu/.ssh/id_ed25519 (shrianshsingh2@gmail.com)
ubuntu@ip-172-31-88-200:~/mlps$ cat ~/.ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZD11NTESAAAAIDUp8J9MCIVdgiQke9G0ks4HdJan+p9E1HzbW43jLI shrianshsingh2@gmail.com
ubuntu@ip-172-31-88-200:~/mlps$ git clone git@github.com:Shriansh16/Deployment_test_aws.git
Cloning into 'Deployment_test_aws'...
remote: Enumerating objects: 6, done.
remote: counting objects: 100% (6/6), done.
remote: compressing objects: 100% (6/6), done.
remote: Total 6 (delta 0), reused 6 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (6/6), 21.26 KiB | 21.26 MiB/s, done.
ubuntu@ip-172-31-88-200:~/mlps$ configure aws
configure: command not found
ubuntu@ip-172-31-88-200:~/mlps$ aws configure
AWS Access Key ID [None]: AKIA42QPMILYNR84BRZA
AWS Secret Access Key [None]: K080V3/u2oiVa7w2+LZr4B5387XTdUvop0/lbbz
Default region name [None]: "us-east-1"
Default output format [None]:
ubuntu@ip-172-31-88-200:~/mlps$
```

(don't include " " in region)

Step 26. Go to repo using cd command and install the requirements.txt.

Step 27. Now simply run the app, deployment has been done.

Step 28. But now I want to do one thing I don't want to enter credential for connecting it with s3 bucket, lets do that .

Step 29. Create a new iam role->create role->aws service->use case: ec2-> add permission policy: s3 full access->role name (any)->create role.

Step 30. How to attach this role? Go to instances->actions->security->modify iam role->choose role->update iam role

Course: 2025 Deploy ML Model | IAM | Global | app - Streamlit

us-east-1.console.aws.amazon.com/iam/home?region=us-east-1#/roles/create?trustedEntityType=AWS\_SERVICE&selectedService=EC2&selectedUseCase=EC2&policies=arn%3Aaws%3Aiam%3A...

Search [Alt+S]

Global Bruce\_Wayne

IAM > Roles > Create role

Step 1 Select trusted entity  
Step 2 Add permissions  
Step 3 Name, review, and create

### Name, review, and create

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
Maximum 64 characters. Use alphanumeric and '+', '@', '-', '\_' characters.

**Description**  
Add a short explanation for this role.  
Allows EC2 instances to call AWS services on your behalf.  
Maximum 1000 characters. Use letters (A-Z and a-z), numbers (0-9), tabs, new lines, or any of the following characters: '.,:;/[]{}%\*^&~='.

**Step 1: Select trusted entities** [Edit](#)

**Trust policy**

```
1- {  
2-   "Version": "2012-10-17",  
3-   "Statement": [  
4-     {  
5-       "Effect": "Allow",  
6-       "Action": [  
7-         "sts:AssumeRole"
```

Course: 2025 Deploy ML Model | Instances | EC2 | us-east-1 | app - Streamlit

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#Instances:

Search [Alt+S]

United States (N. Virginia) Bruce\_Wayne

EC2 > Instances

Dashboard  
EC2 Global View  
Events

**Instances**  
Instances  
Instance Types  
Launch Templates  
Spot Requests  
Savings Plans  
Reserved Instances  
Dedicated Hosts  
Capacity Reservations

**Images**  
AMIs  
AMI Catalog

**Elastic Block Store**  
Volumes  
Snapshots  
Lifecycle Manager

**Network & Security**

### Instances (1/1) Info

Last updated less than a minute ago

[Connect](#) [Instance state](#) [Actions](#) [Launch instances](#)

Find Instance by attribute or tag (case-sensitive)

All states

<input checked="" type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm s
<input checked="" type="checkbox"/>	hummer12	i-0d7af087732603f35	Running	t2.micro	2/2 checks pass	<a href="#">View al</a>

[Change security groups](#)  
[Get Windows password](#)  
[Modify IAM role](#)

[Connect](#)  
[View details](#)  
[Manage instance state](#)  
[Instance settings](#)  
[Networking](#)  
[Security](#)  
[Image and templates](#)  
[Monitor and troubleshoot](#)

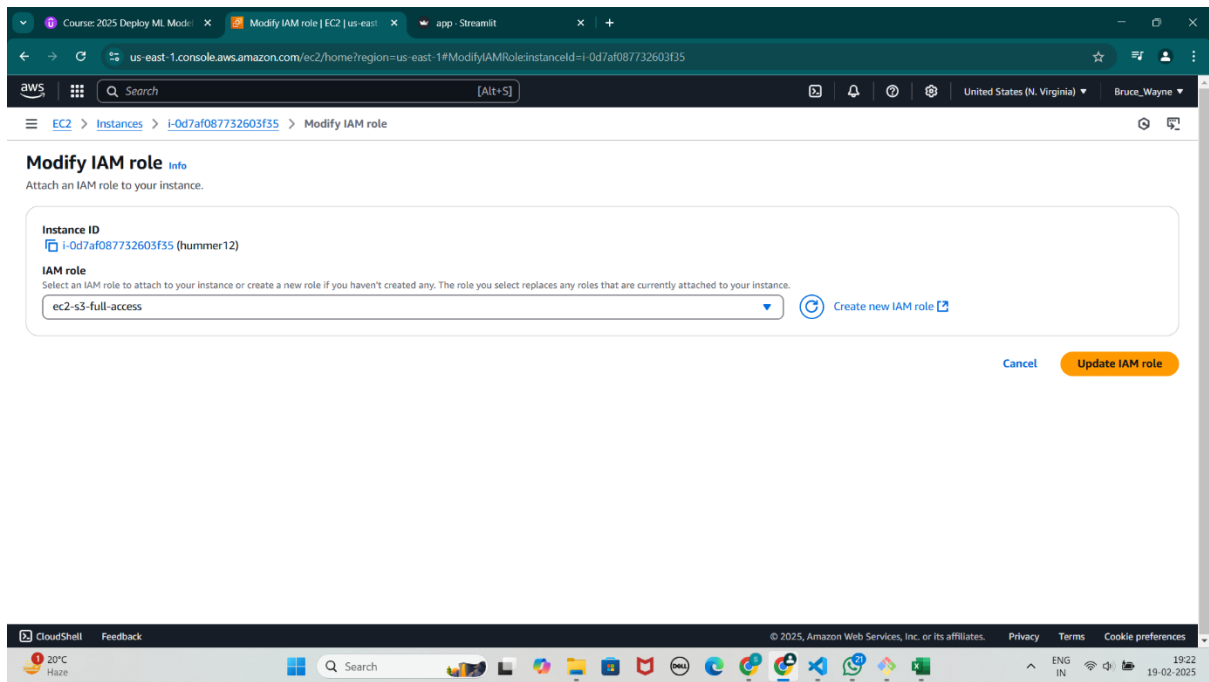
[Public IP](#)  
[ec2-34-2](#)

### i-0d7af087732603f35 (hummer12)

[Details](#) [Status and alarms](#) [Monitoring](#) [Security](#) [Networking](#) [Storage](#) [Tags](#)

**Instance summary Info**

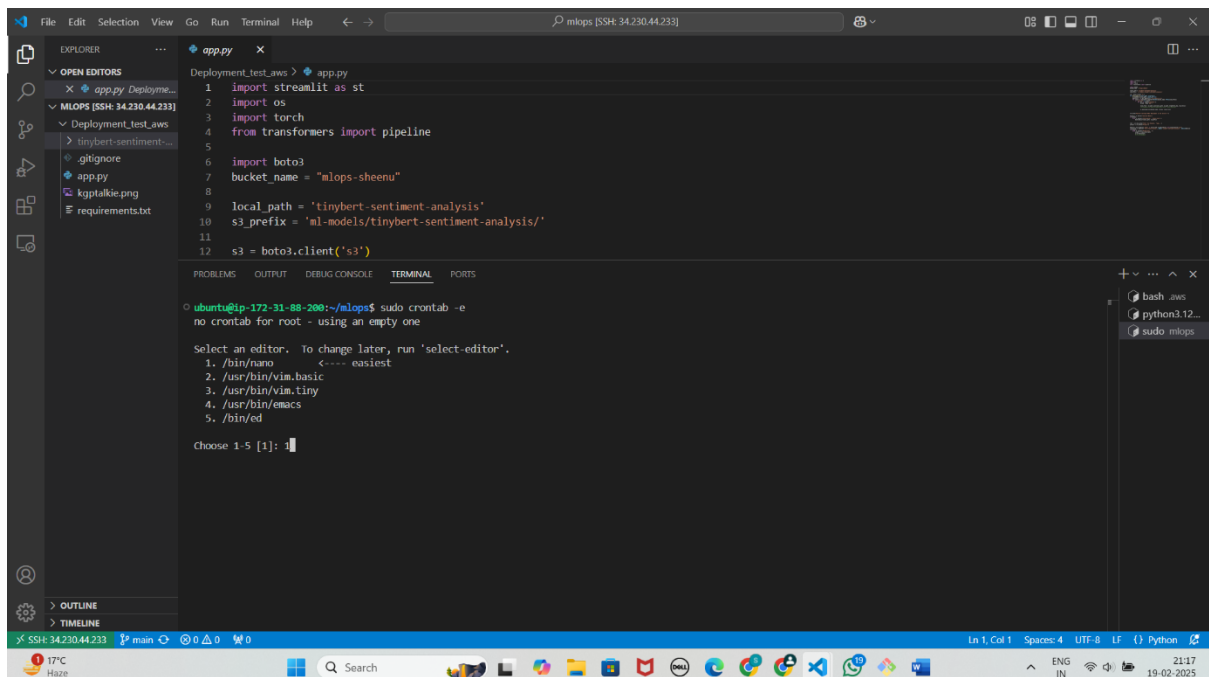
<b>Instance ID</b> i-0d7af087732603f35	<b>Public IPv4 address</b> 34.230.44.233   <a href="#">open address</a>	<b>Private IPv4 addresses</b> 172.31.88.200
<b>IPv6 address</b> -	<b>Instance state</b> Running	<b>Public IPv4 DNS</b> ec2-34-230-44-233.compute-1.amazonaws.com   <a href="#">open address</a>



Now our ec2 can access the s3 bucket without credentials

Step 31. Now we have to add that functionality so that whenever we restart our instance our streamlit must run.

Step 32. Use the command “sudo crontab -e” -> use nano editor (1)





Step 33. Go below and write this command

```
@reboot /opt/conda/bin/python -m streamlit run  
/home/ubuntu/mlops/Deployment_test_aws/app.py
```

(location will vary)

Ctrl+s and ctrl+x

Now if we stop and start our instance our streamlit app will be opened

Step 34. But the thing is for every restart we will get different ips

If we want the permanent ip use the service call elastic ip but it is paid

And assign it with the instance