Assignment–3

Module-3 Python

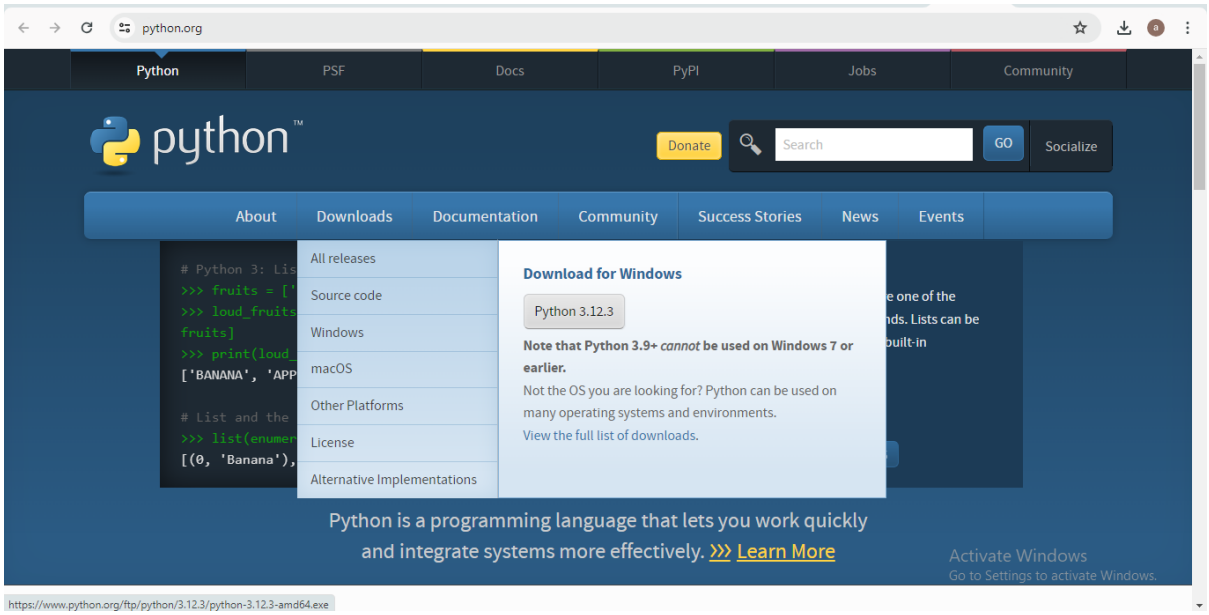
Submitted by : Shaik Junaid Adil

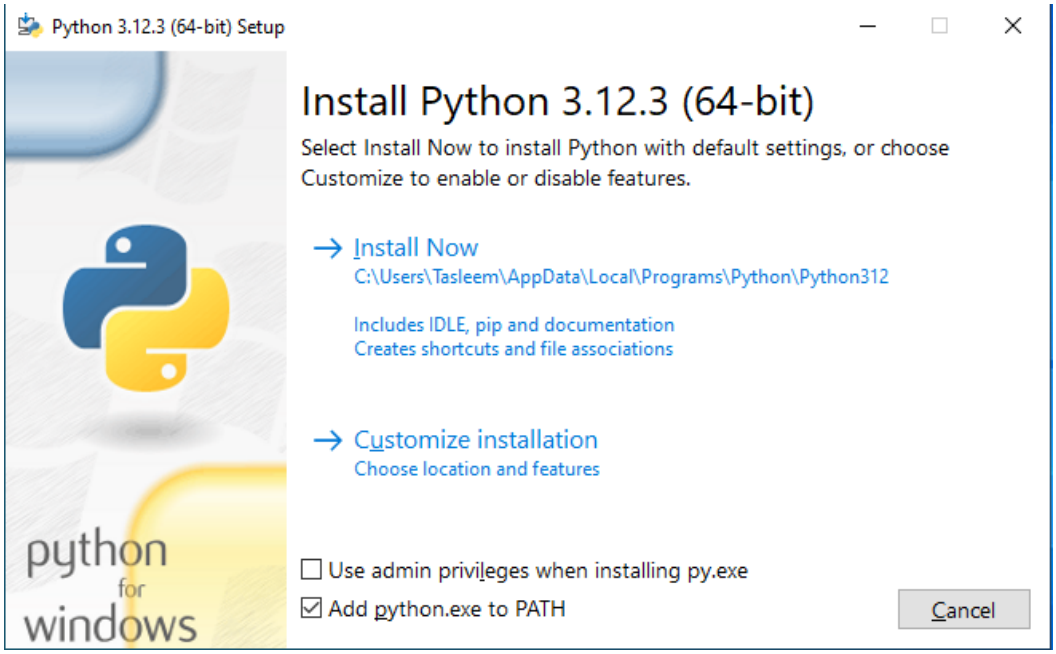
Date of Submission: 13-05-2024

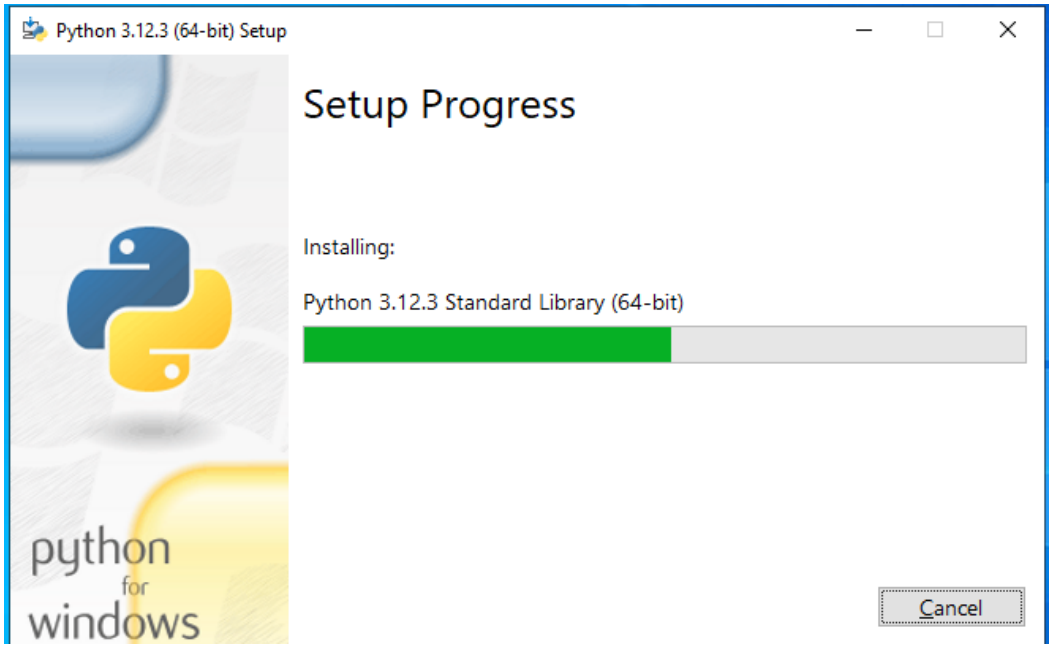
Submitted to: Vikul

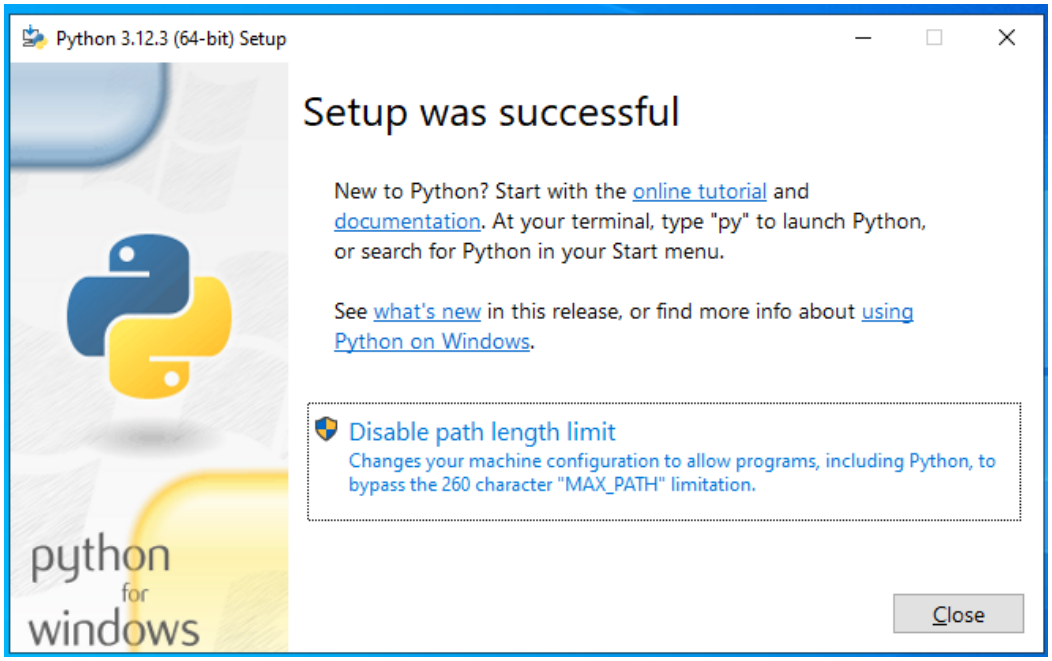
L1 - In Local Windows Machine Install and setup Environment Variable for Python

Step 1: Go to " Python.org " and download the latest version. During the installation ensure to check the box that says "Add Python 3.x to PATH" at the beginning of the installation process. This option sets the environment variable for Python, which allows to run Python from any command prompt or PowerShell window.

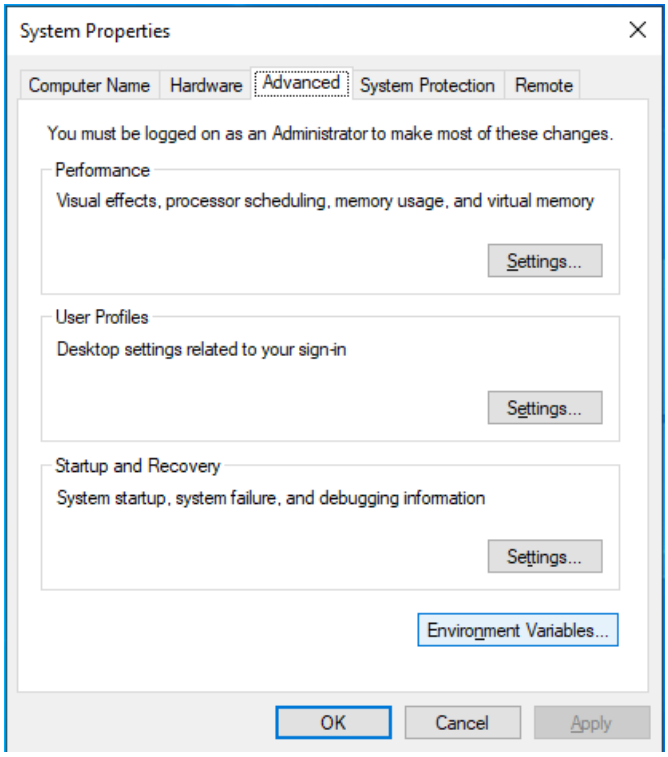


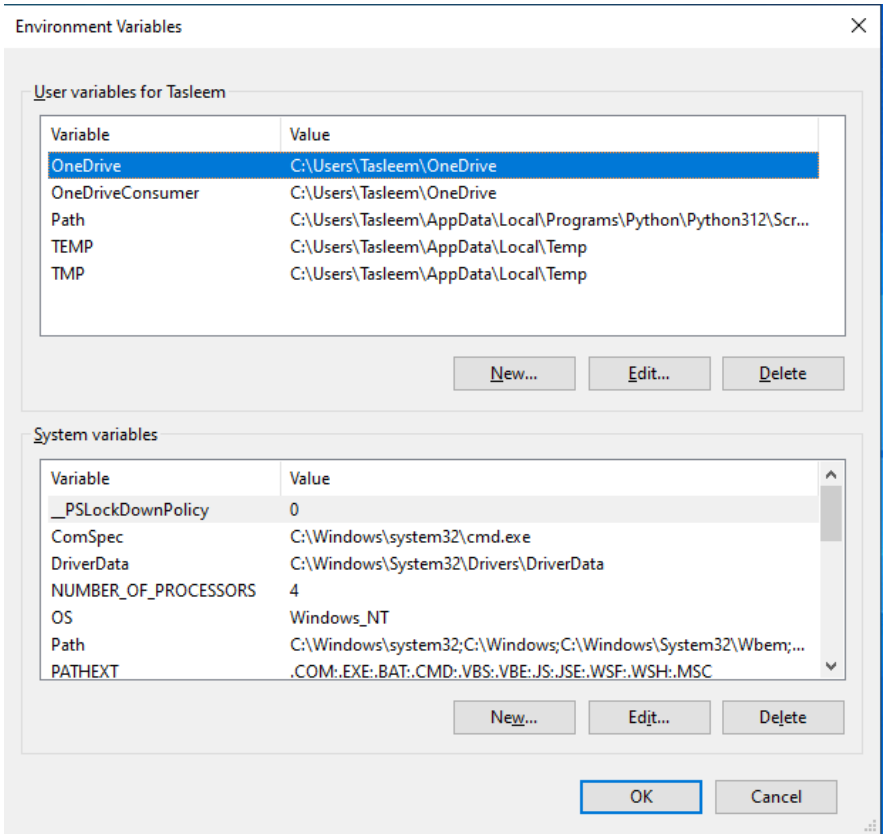



Python 3.12.3 (64-bit) Setup
Setup Progress
Installing:
Python 3.12.3 Standard Library (64-bit)
Cancel
python
for
windows

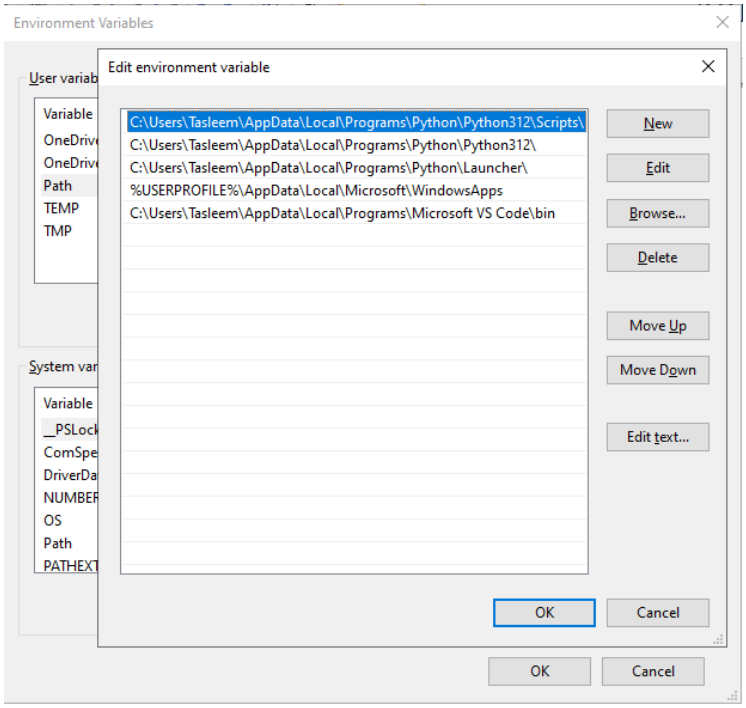

Python 3.12.3 (64-bit) Setup
Setup was successful
New to Python? Start with the online tutorial and documentation. At your terminal, type "py" to launch Python, or search for Python in your Start menu.
See what's new in this release, or find more info about using Python on Windows.
Disable path length limit
Changes your machine configuration to allow programs, including Python, to bypass the 260 character "MAX_PATH" limitation.
Close
python
for
windows

Step 2: To verify, Go to System Properties → Advanced tab → go to Environment Variables..



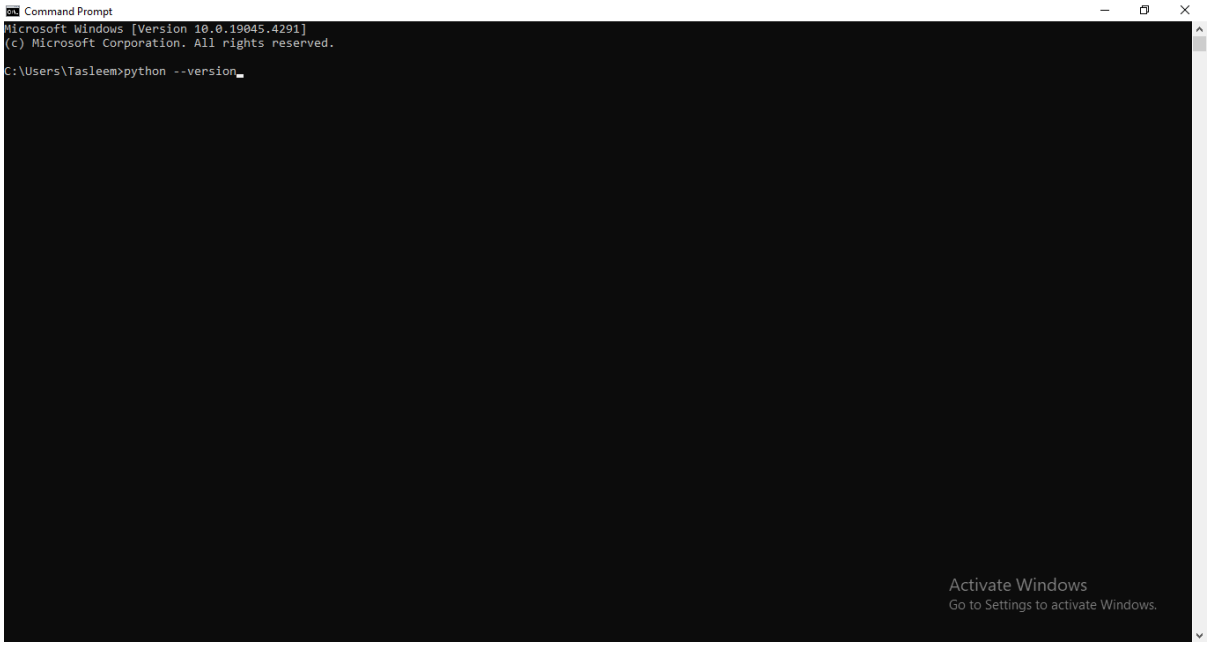


Go to Path,

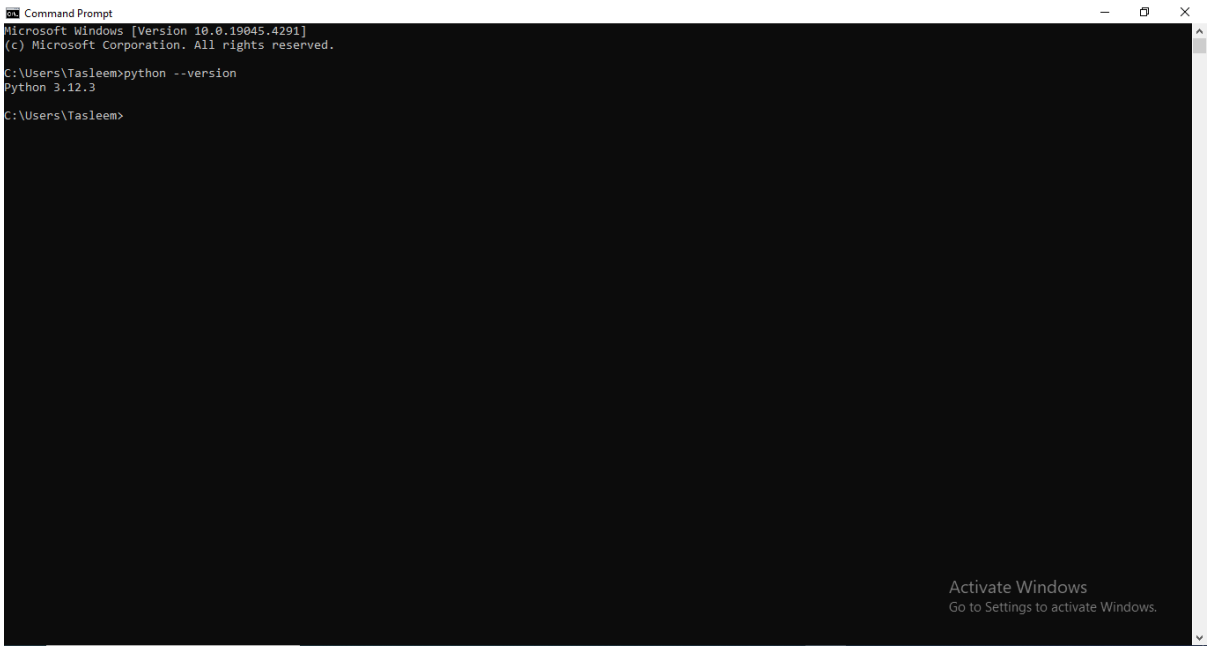


We can see that Python is properly installed in the PATH Environment Variables..

And also we can verify using command prompt.

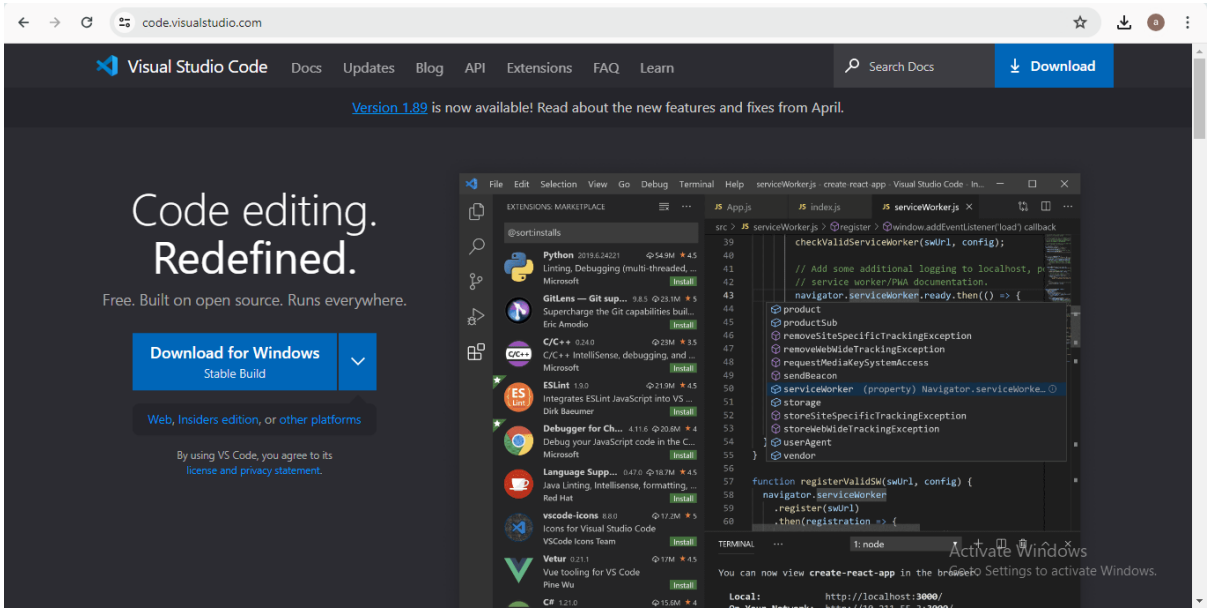


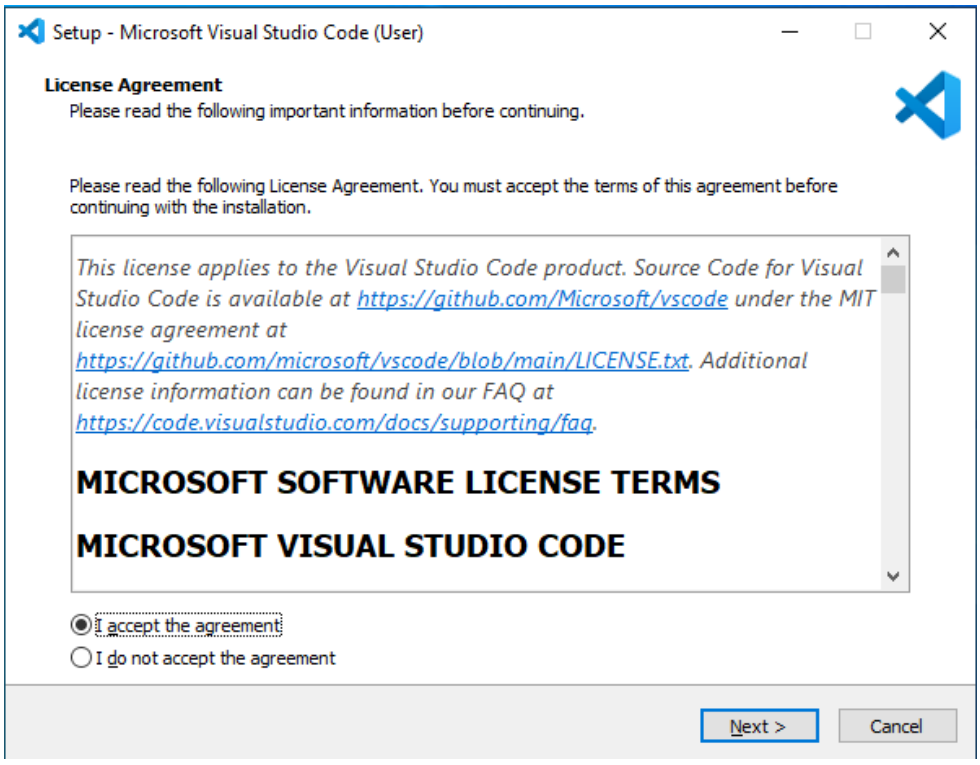
We can see that Python is successfully installed.

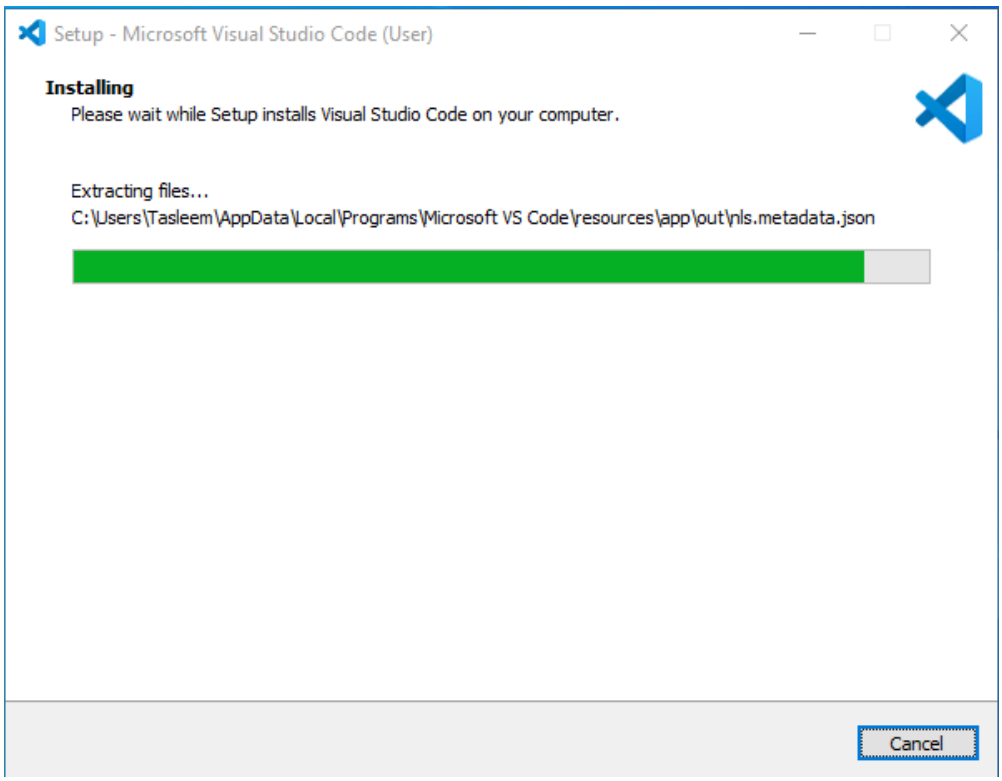


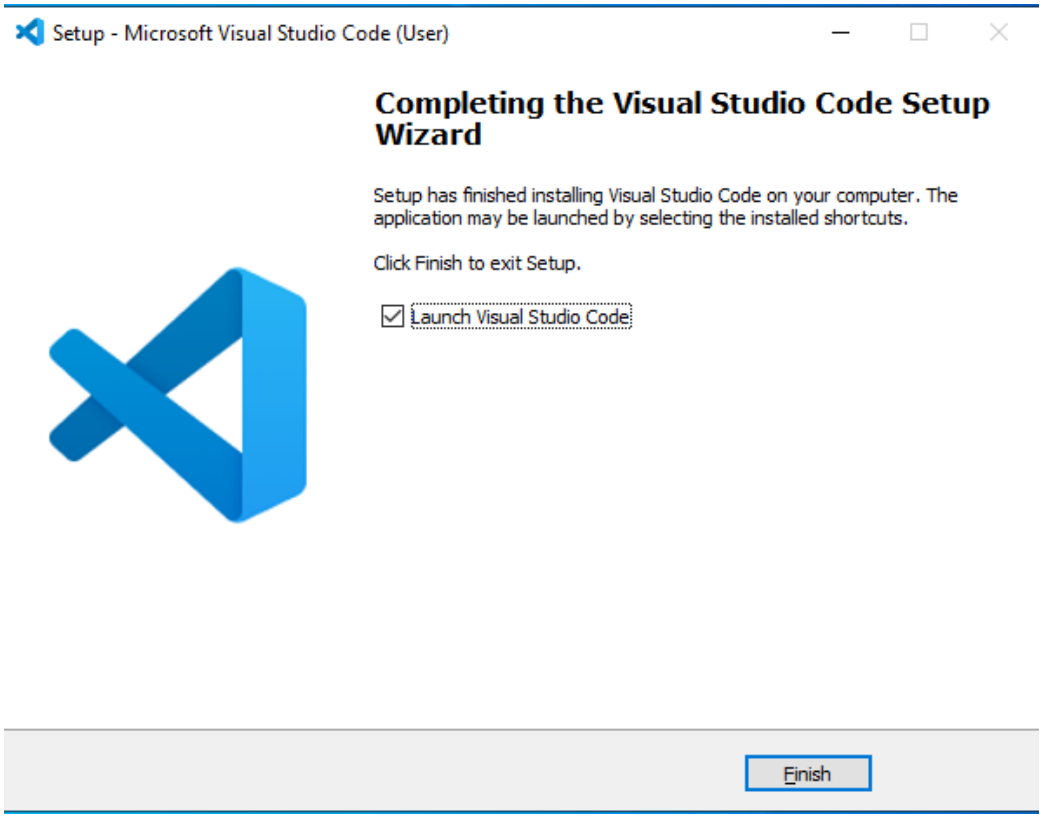
L2 - Install Visual Studio Code and Install Python and Terraform Extensions in VS Code

Step 1: Go to “code.visualstudio.com“, download & Install the VSC.

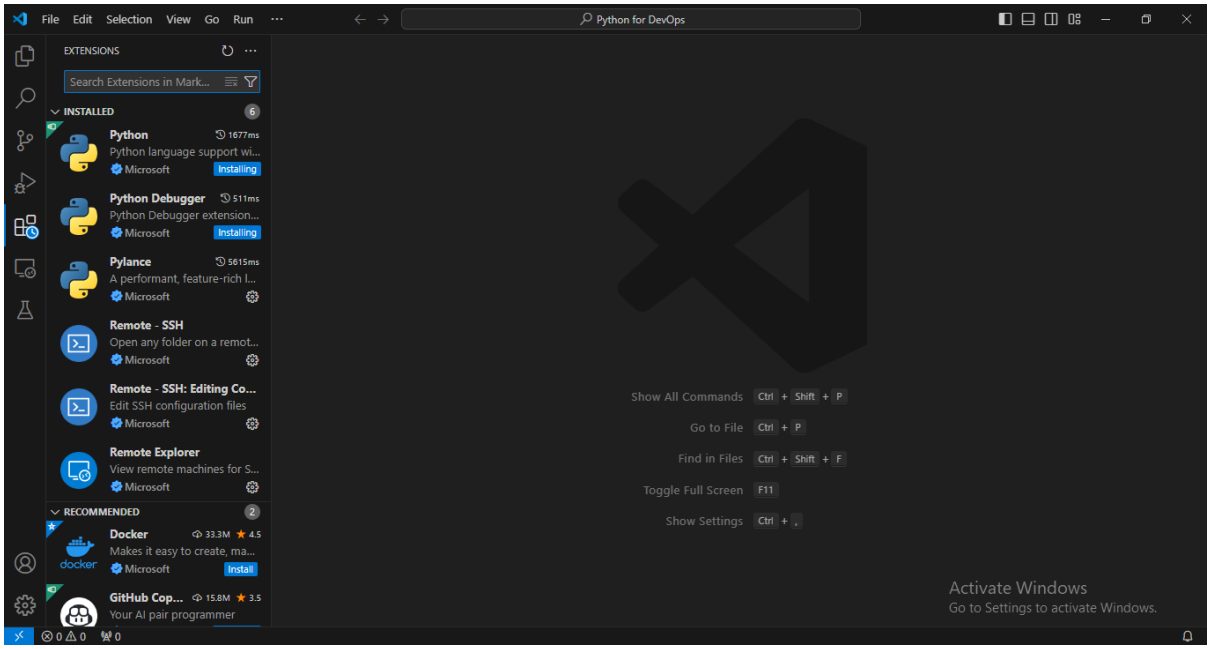





Setup - Microsoft Visual Studio Code (User)
Installing
Please wait while Setup installs Visual Studio Code on your computer.
Extracting files...
C:\Users\Tasleem\AppData\Local\Programs\Microsoft VS Code\resources\app\out\nls.metadata.json
Cancel

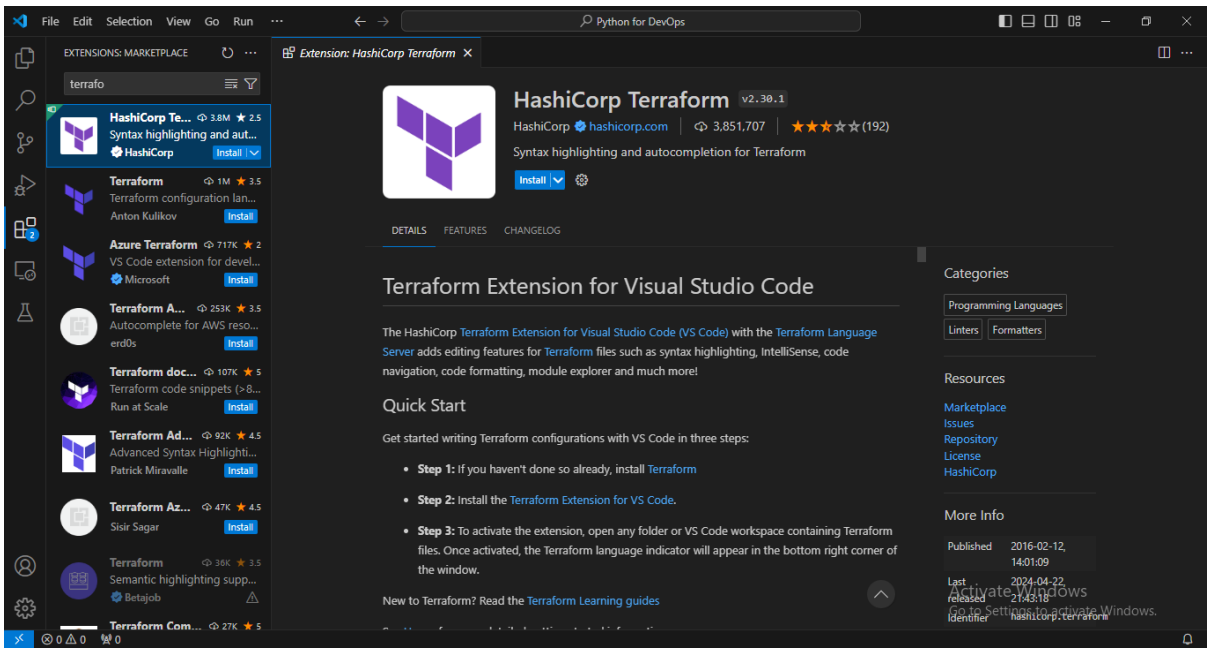

Setup - Microsoft Visual Studio Code (User)
Completing the Visual Studio Code Setup Wizard
Setup has finished installing Visual Studio Code on your computer. The application may be launched by selecting the installed shortcuts.
Click Finish to exit Setup.
Launch Visual Studio Code
Finish

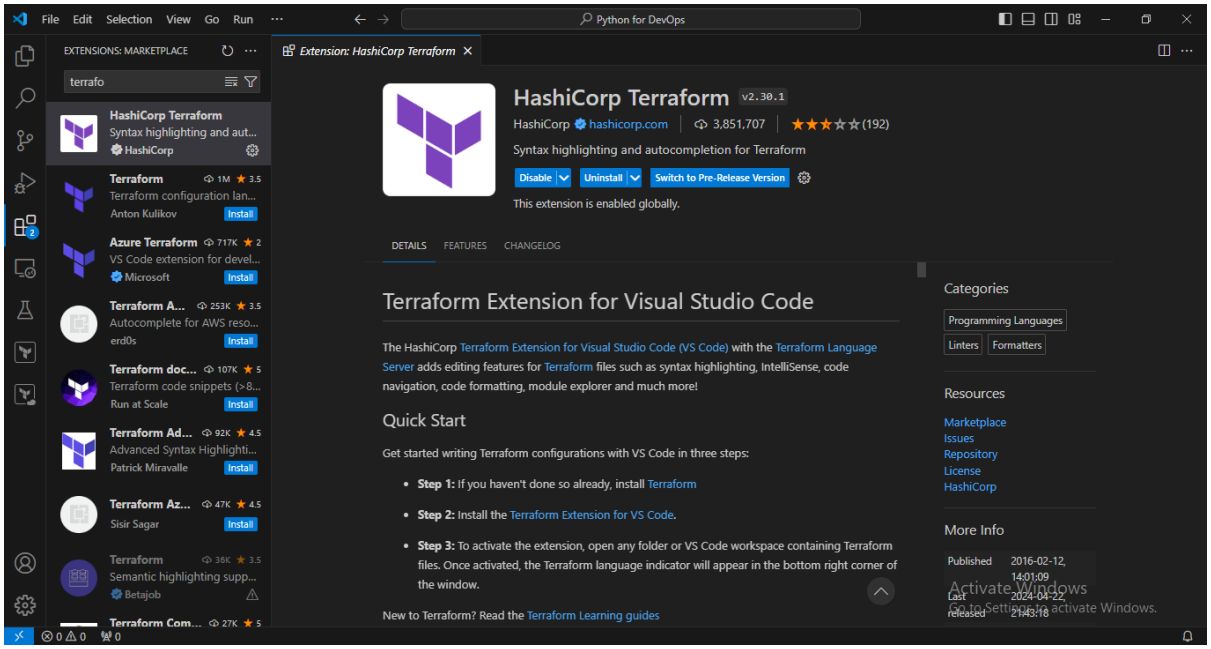
Step 2: Launch the VS Code and go to Extentions from left side panel.



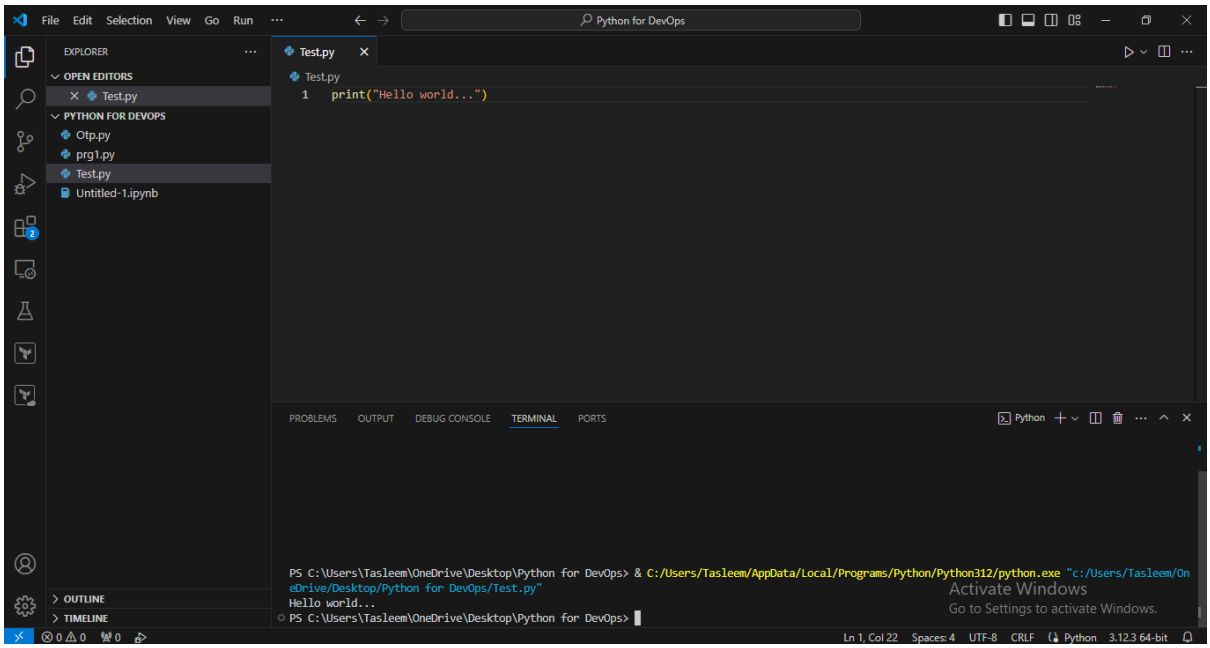
Step 3: Search for Python and click on Install to add the Python Extension to VS Code.

Step 4: Then search for Terraform and Install to add the Extension.



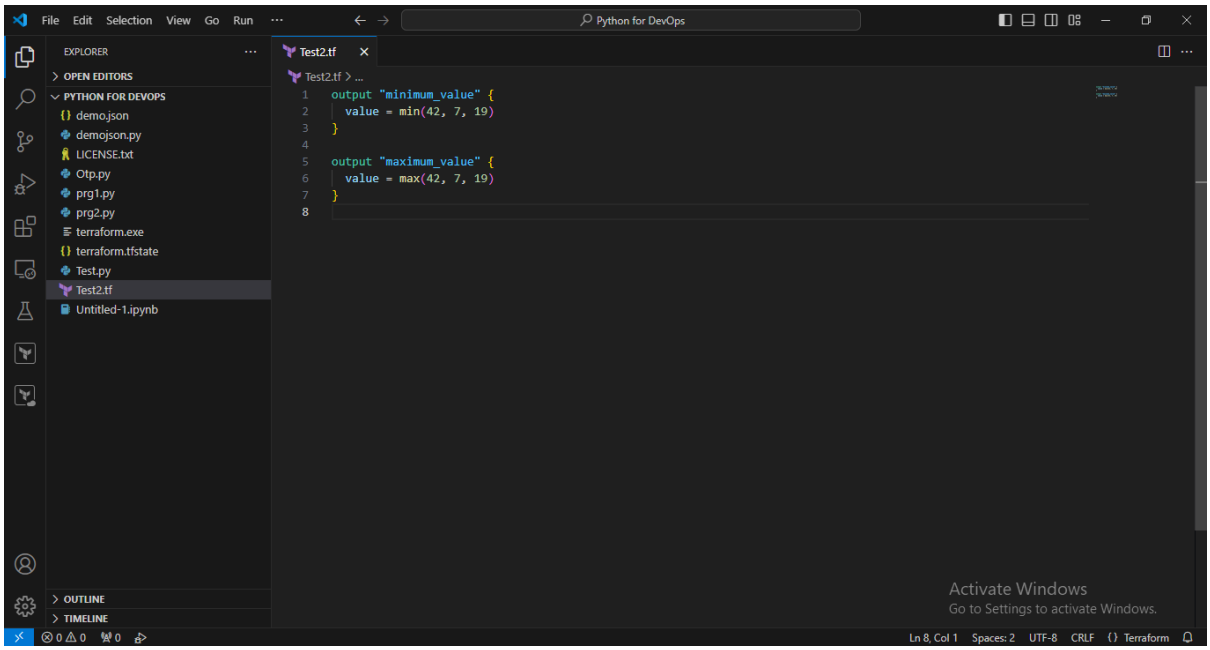


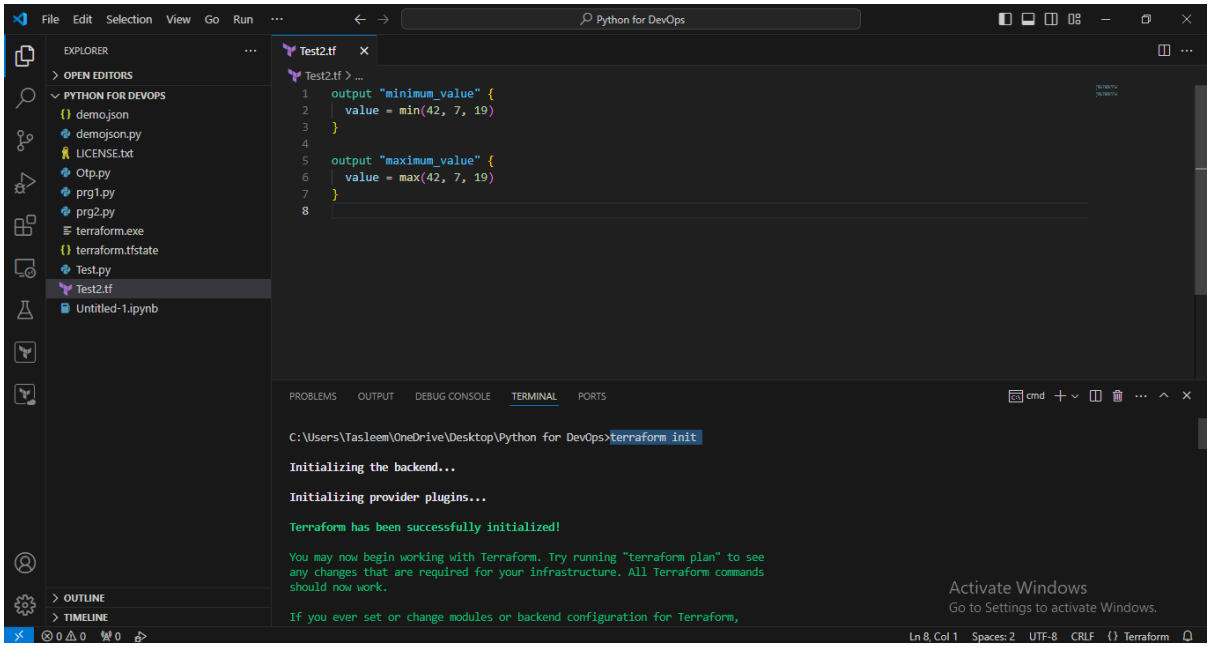
Step 5: To verify Python Extension, create a new file with .py extension “Test.py”. Open the new Terminal and Run the python file “print(“Hello world…”)”.



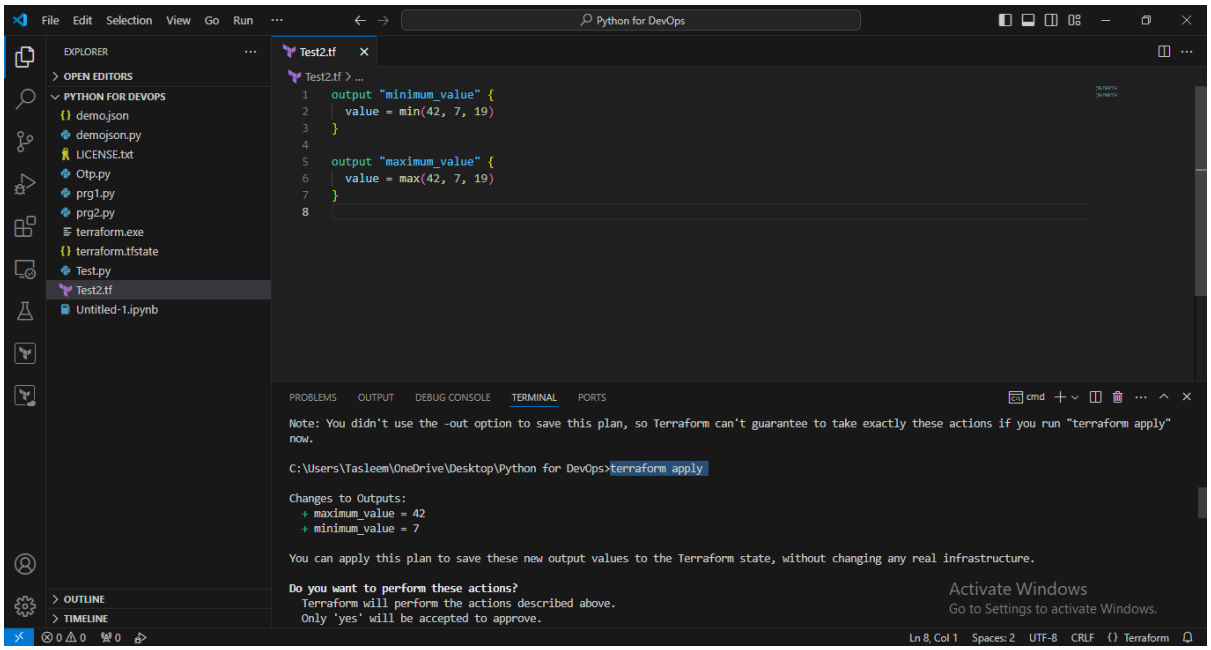
We can see the output in the Terminal below.

Step 6: To verify Terraform Extension, create a new file with the name “Test2.tf“ and execute the command “terraform init” to initialize Terraform

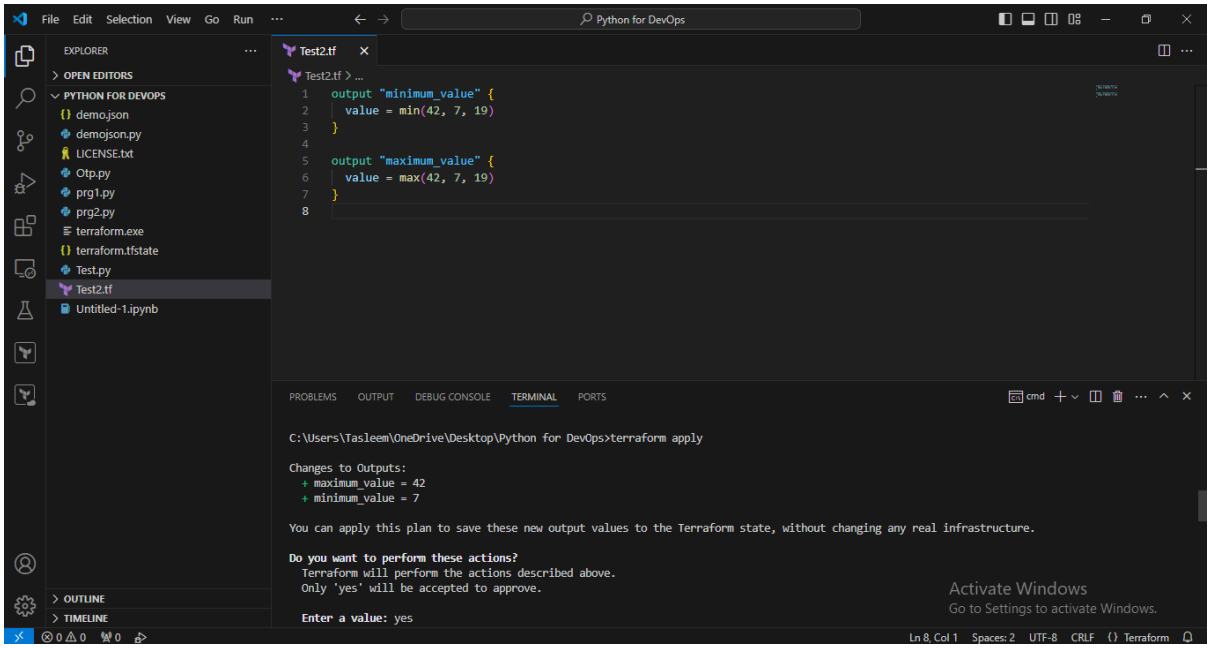


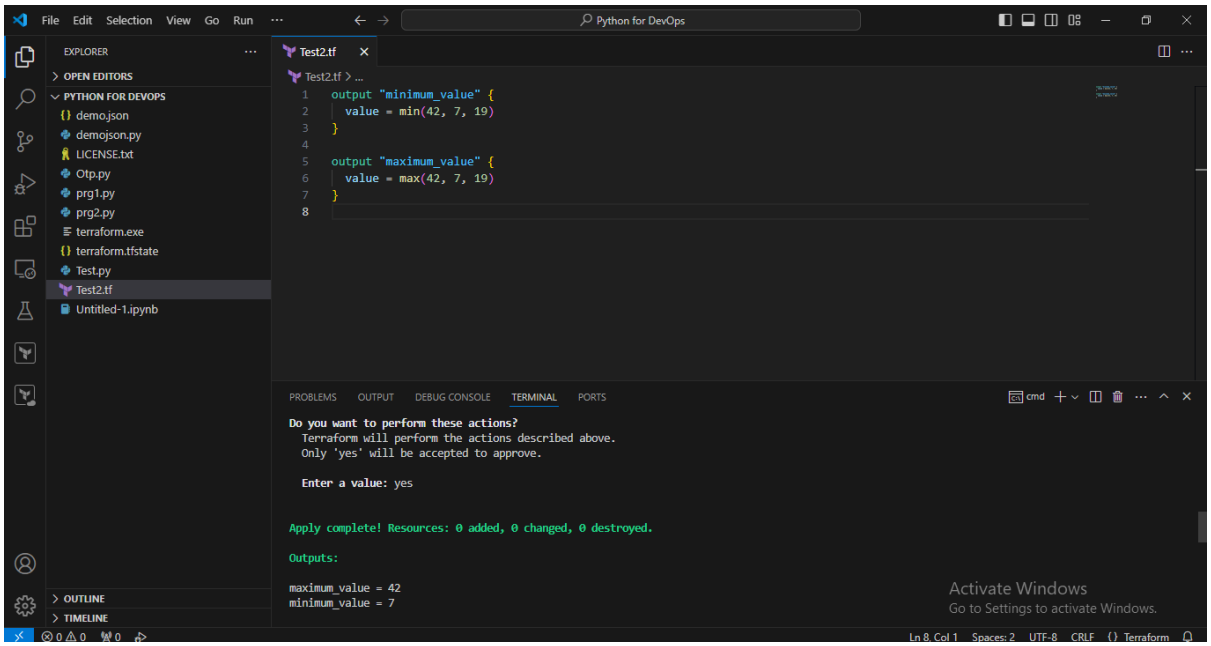


Then run the command “terraform apply”



We can see the output is correct. So we can enter yes to apply



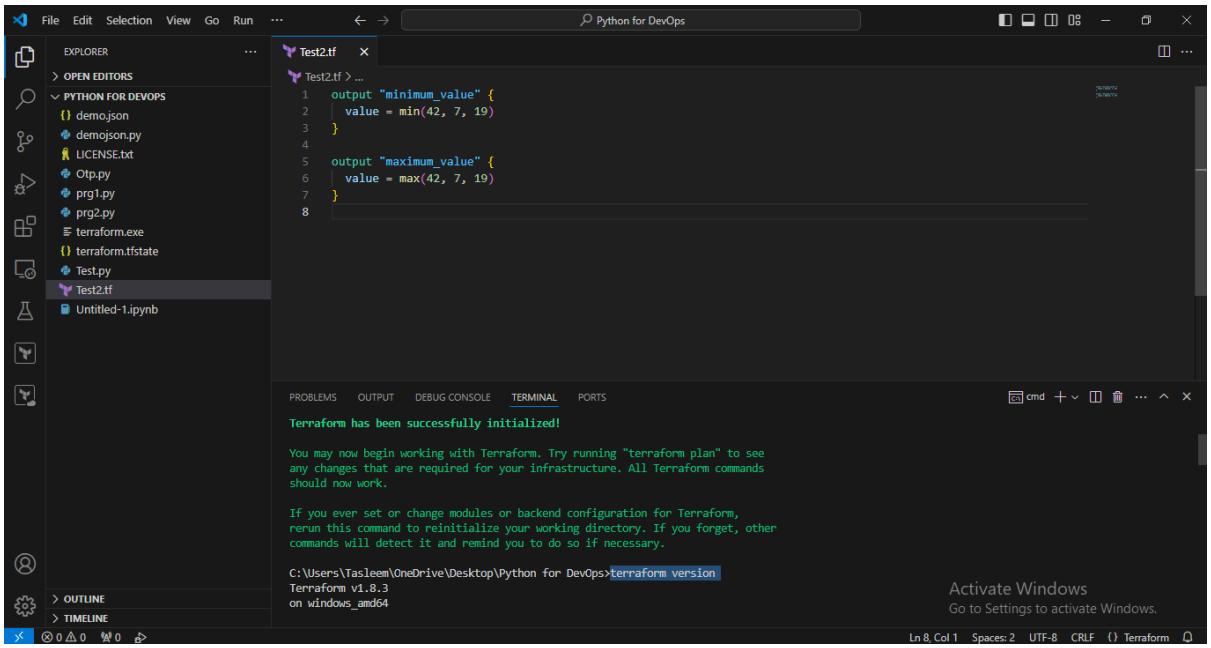


We can see the Output in the above snip.

Maximun_value = 42

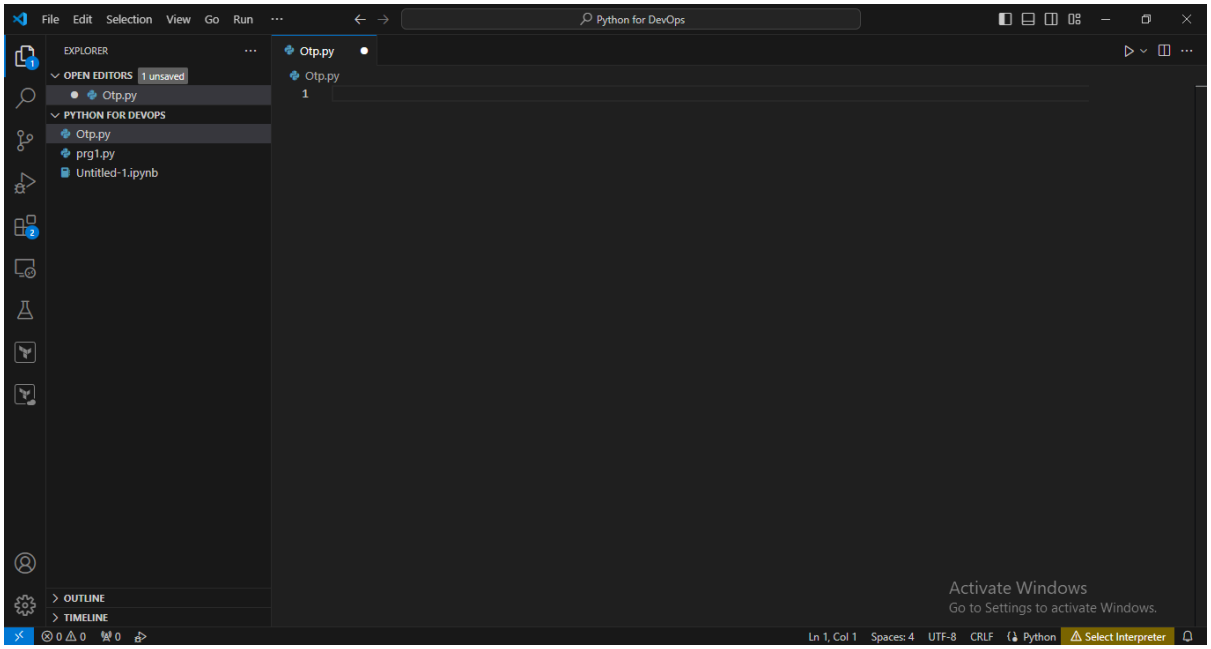
Minimum_value = 7

We can also see the Terraform version : v1.8.3

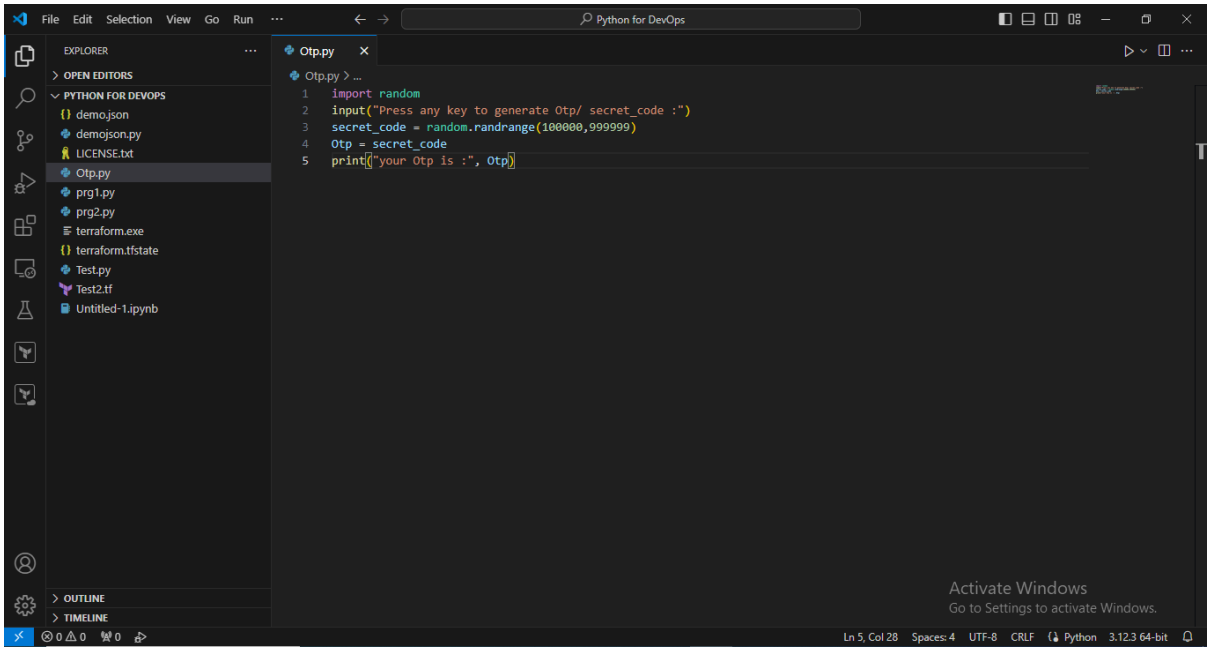


L3 - Create Python Console Application to randomly generate OTP kind of secure code.

Step 1: Go to VSC and create a new file "**Otp.py**"

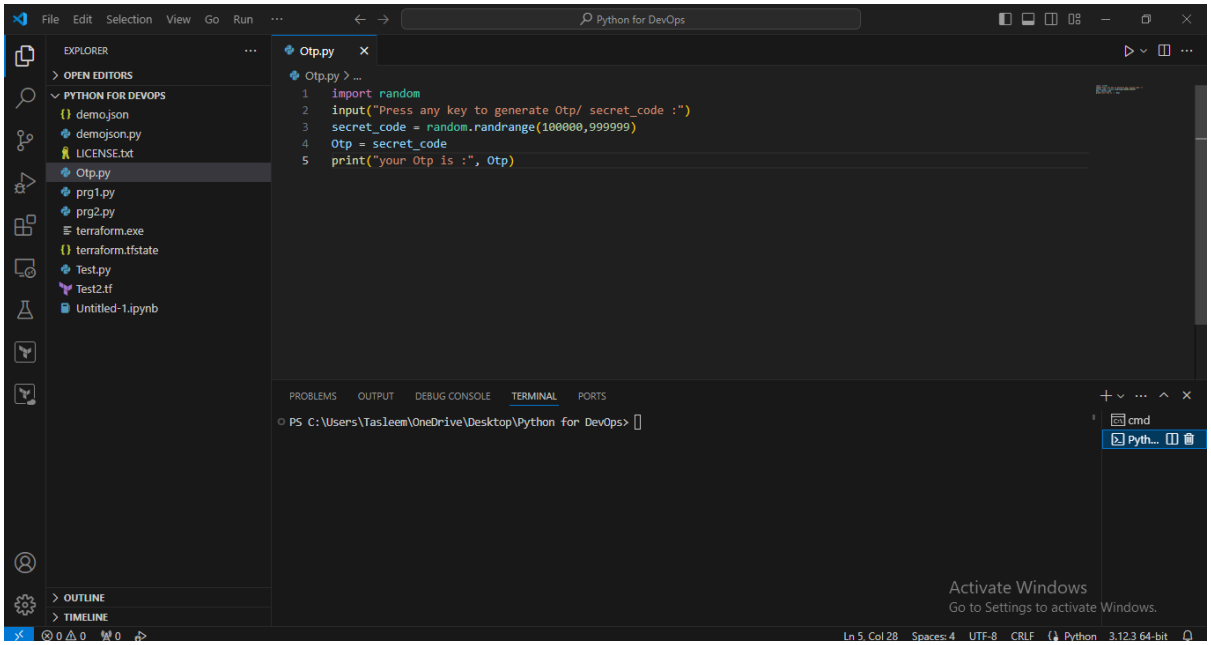


Step 2: enter the programs as shown below.

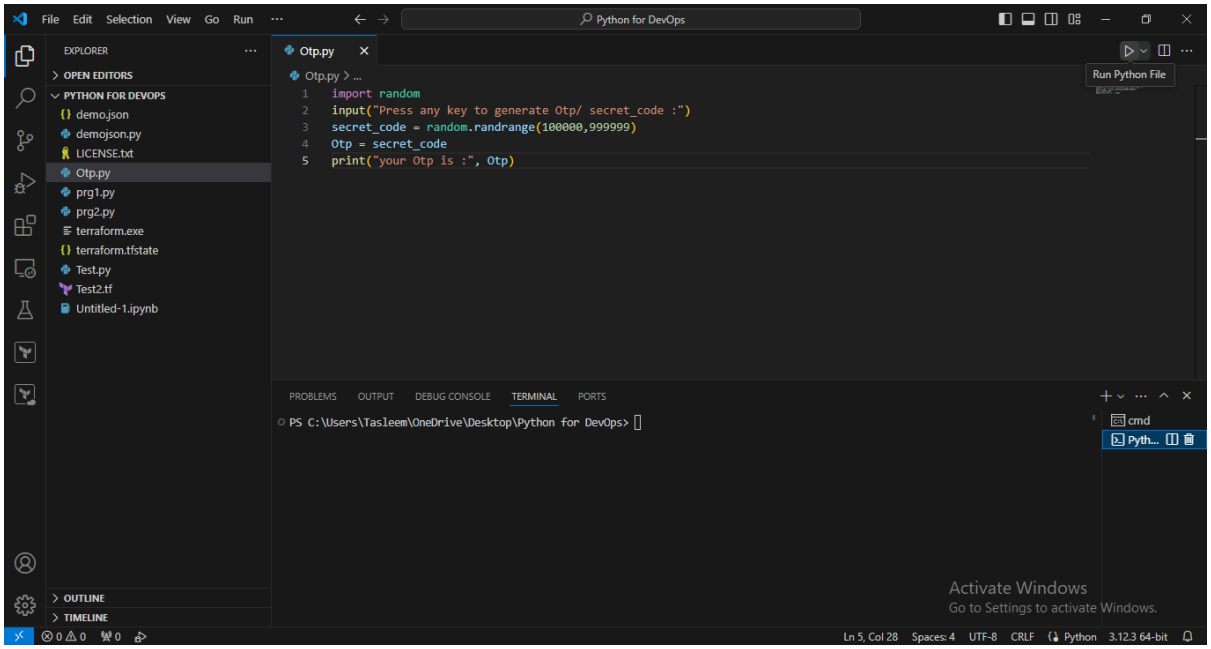


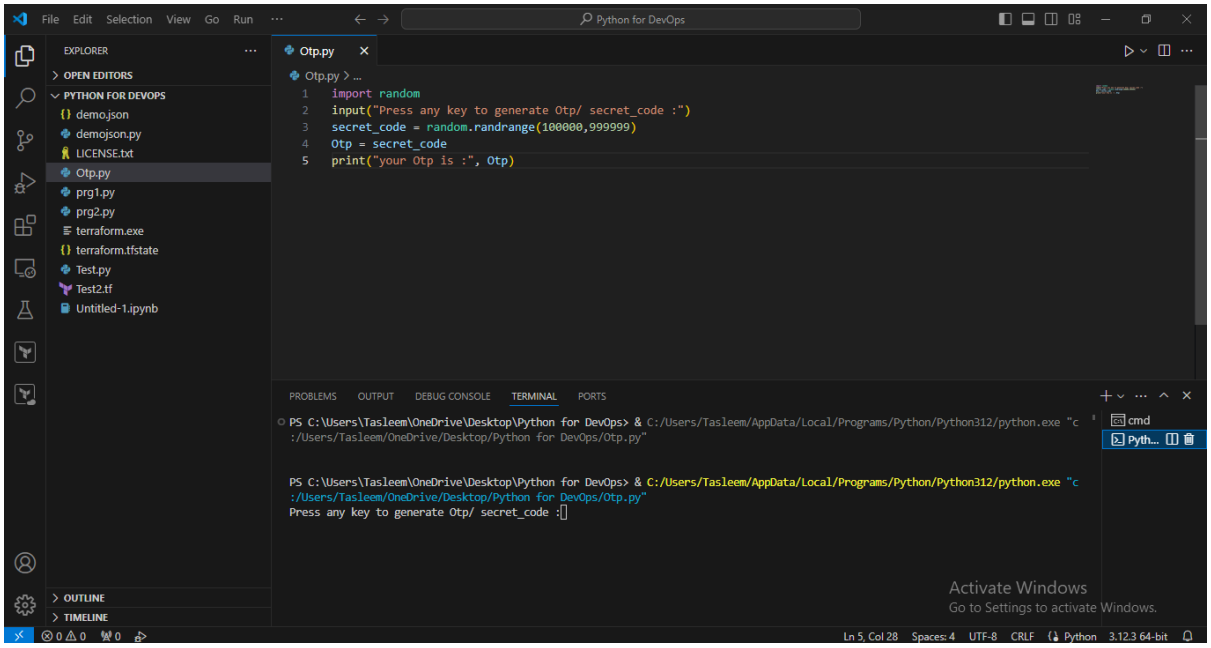
```python
import random
input("Press any key to generate Otp/ secret_code :")
secret_code = random.randrange(100000,999999)
Otp = secret_code
print("your Otp is :", Otp)
```

Step 3: Then open the new Terminal

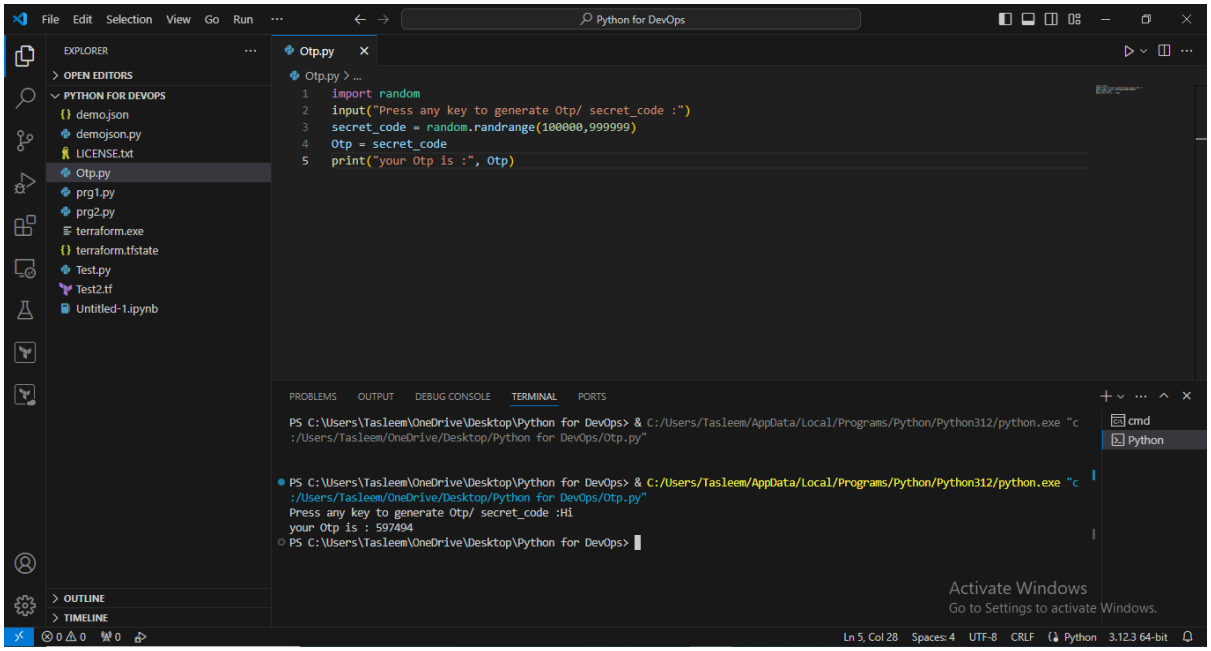


Step 4: Run the python file

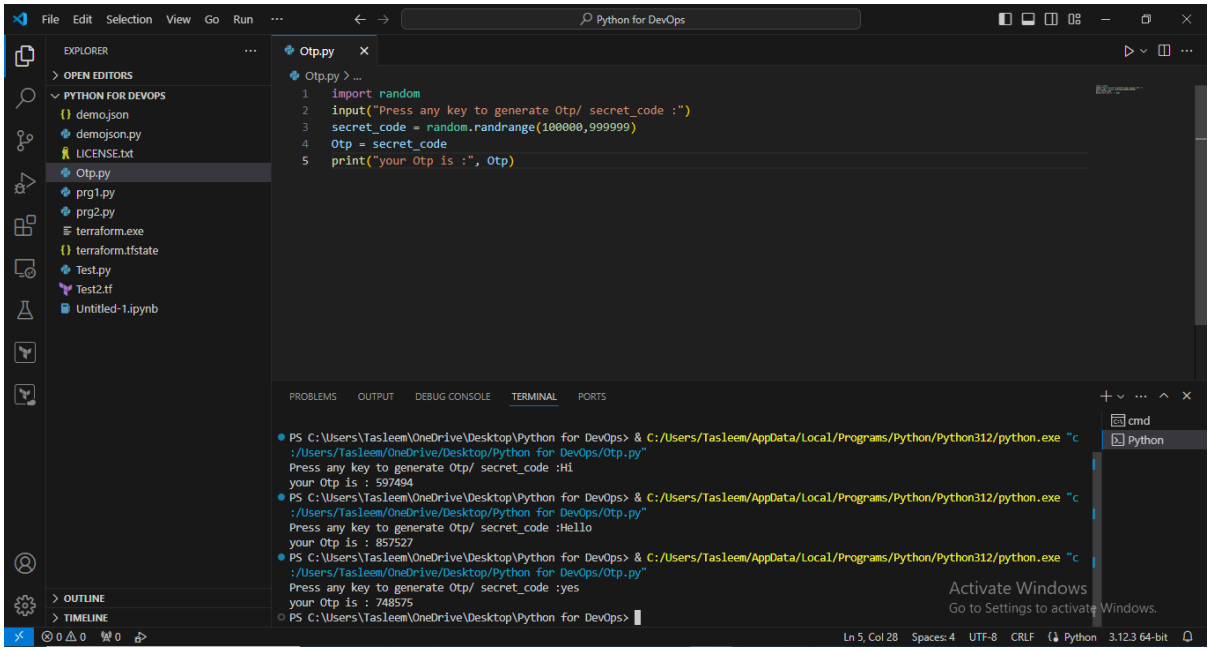




Press any key to generate Otp : give any key and press enter. For Example used " Hi "



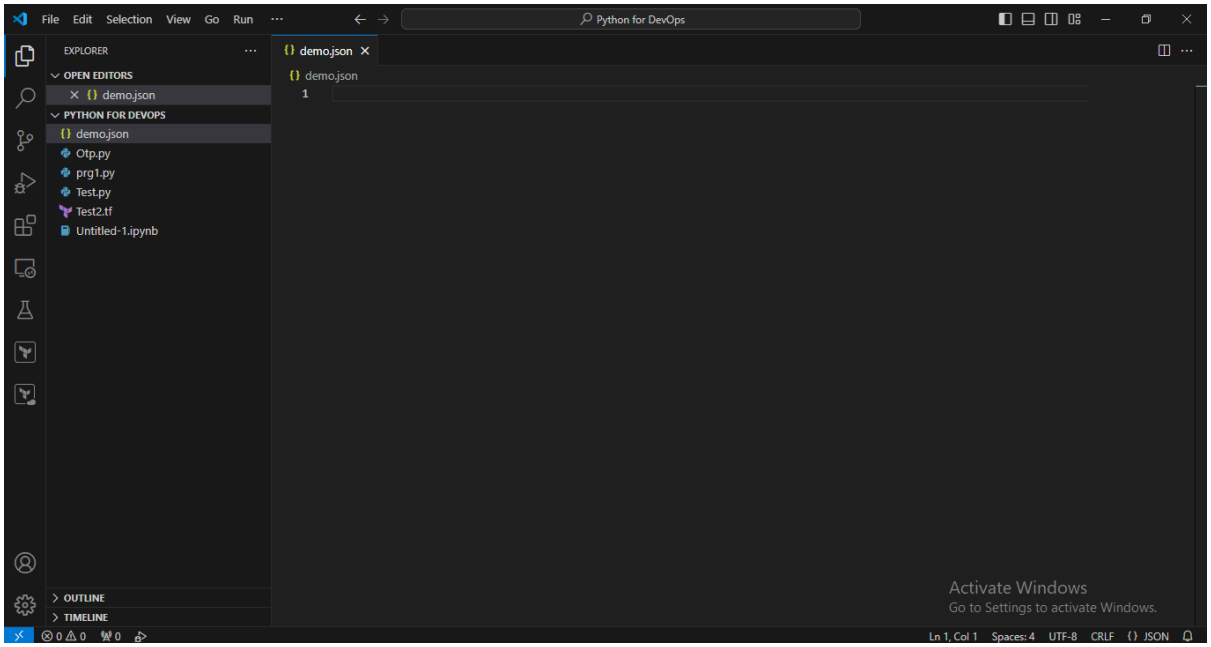
We can see the Otp has been generated.

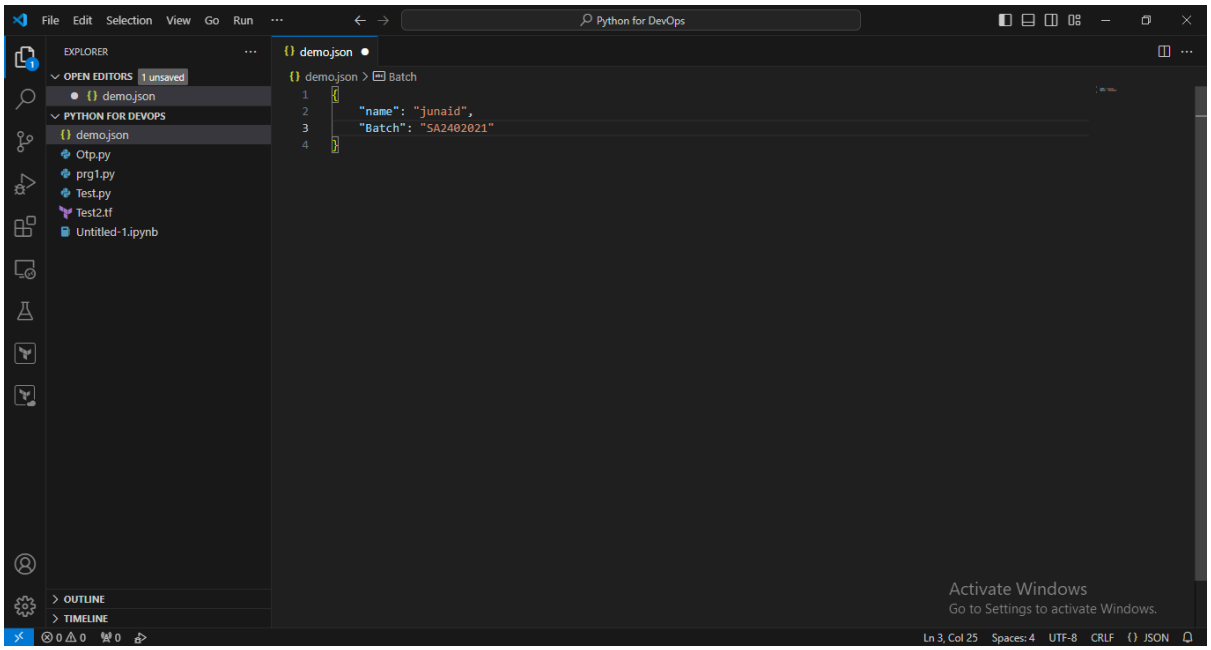


Here we can see how many times we run the file, we get random numbers as we are using the #random module.

L4 - Create Python Console Application to read the contents of .json file and print in the VS Code python console output

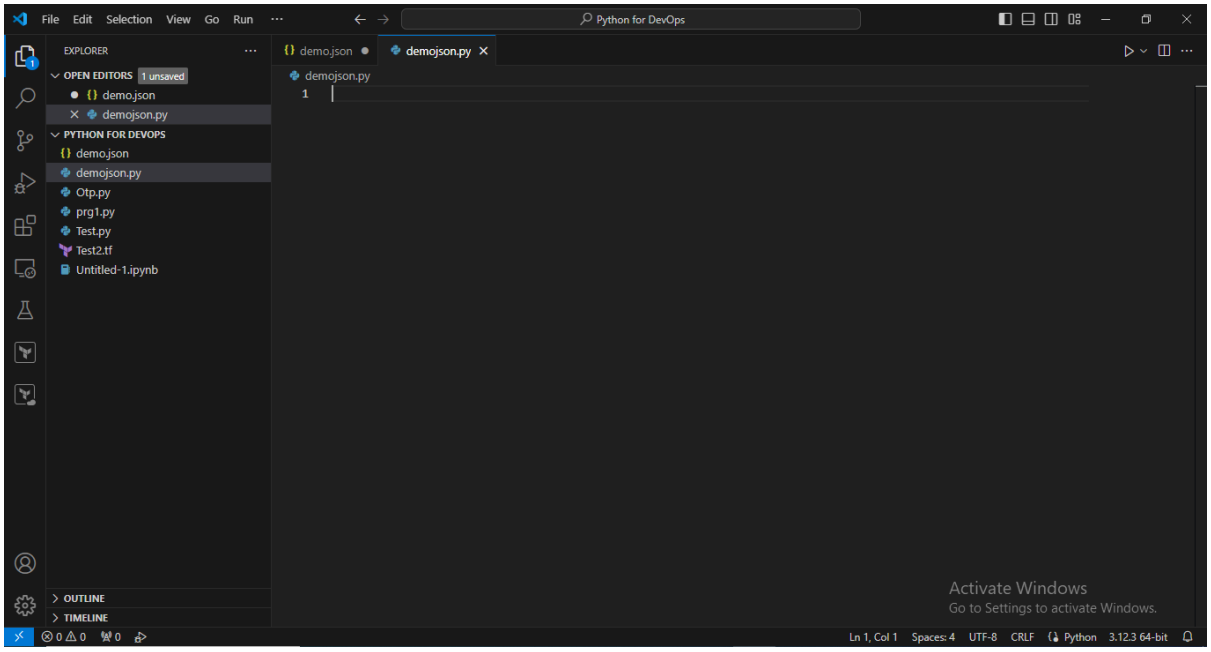
Step 1: Create a Json file “Demo.json”





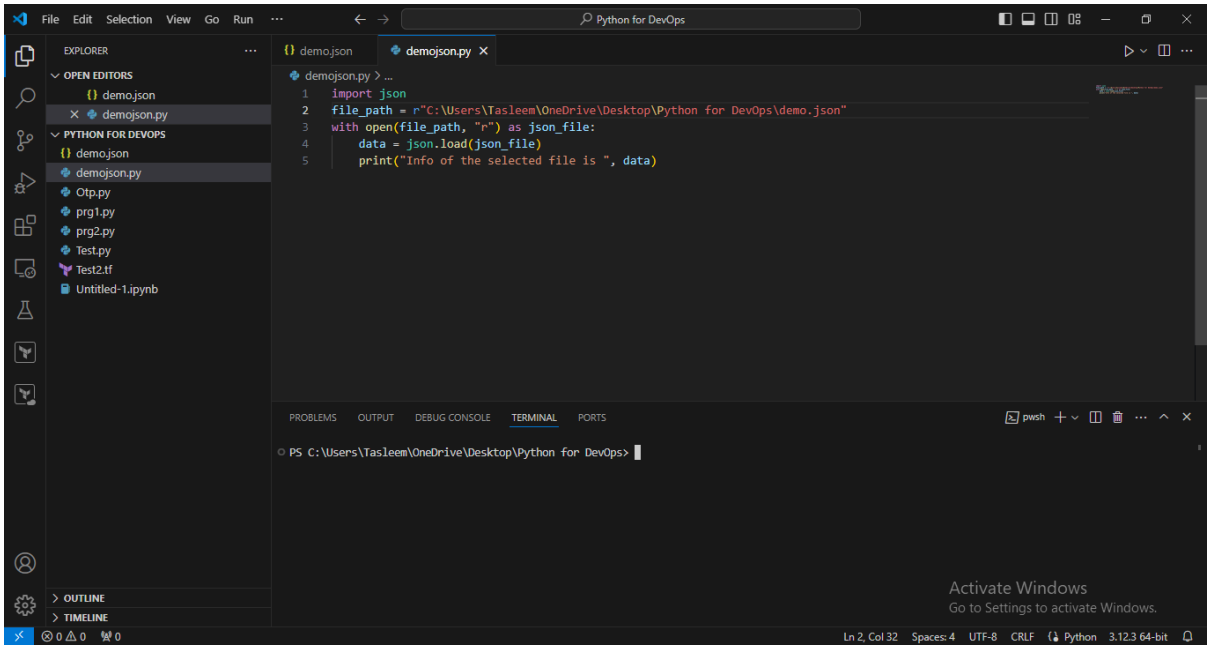
Created a Json file with the program in it.

Step 2: Then create a Python file with the name “demojson.py”

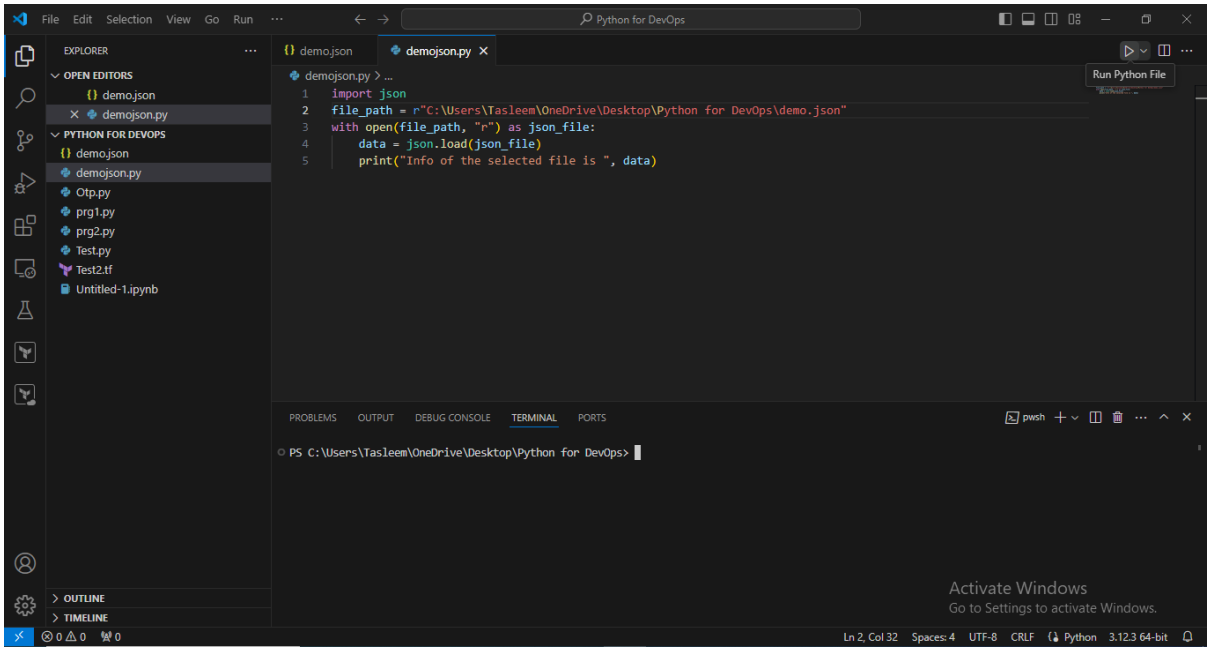


Step 3: Then enter the program as shown below

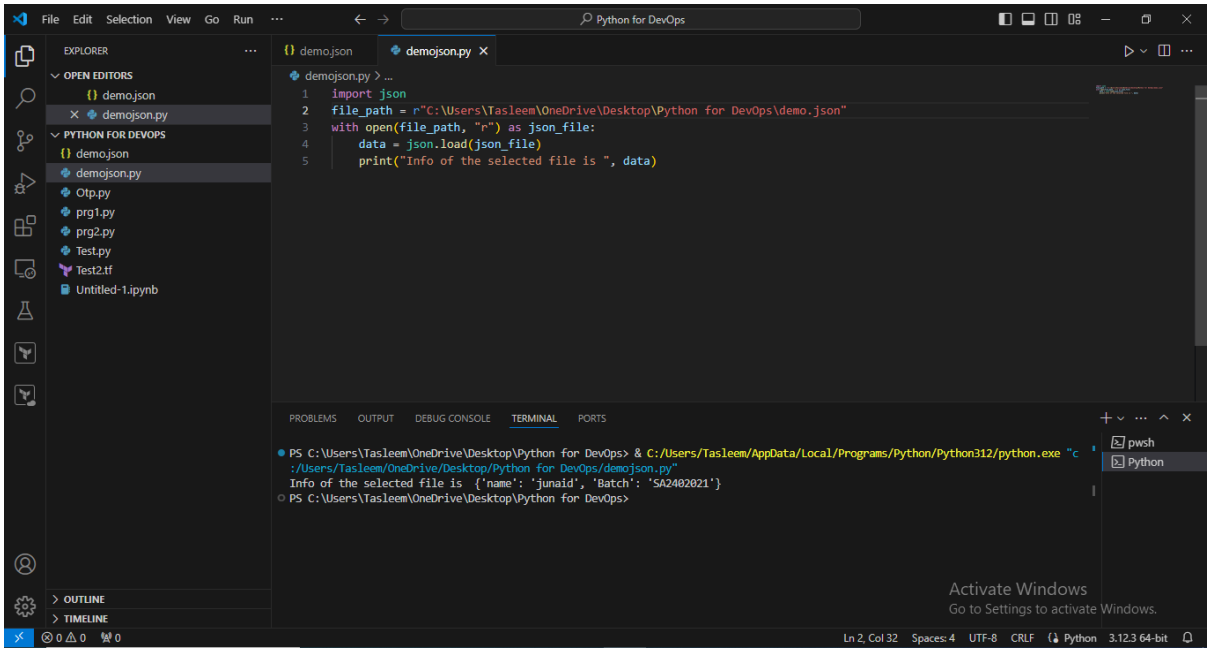
```
Import json
file_path = r"C:\Users\Tasleem\OneDrive\Desktop\Python for DevOps\demo.json"
with open(file_path,"r") as json_file:
    data = json.load(json_file)
    print("Info of the selected file is ", data)
```



Step 4: Run the python file using Run option on top right side

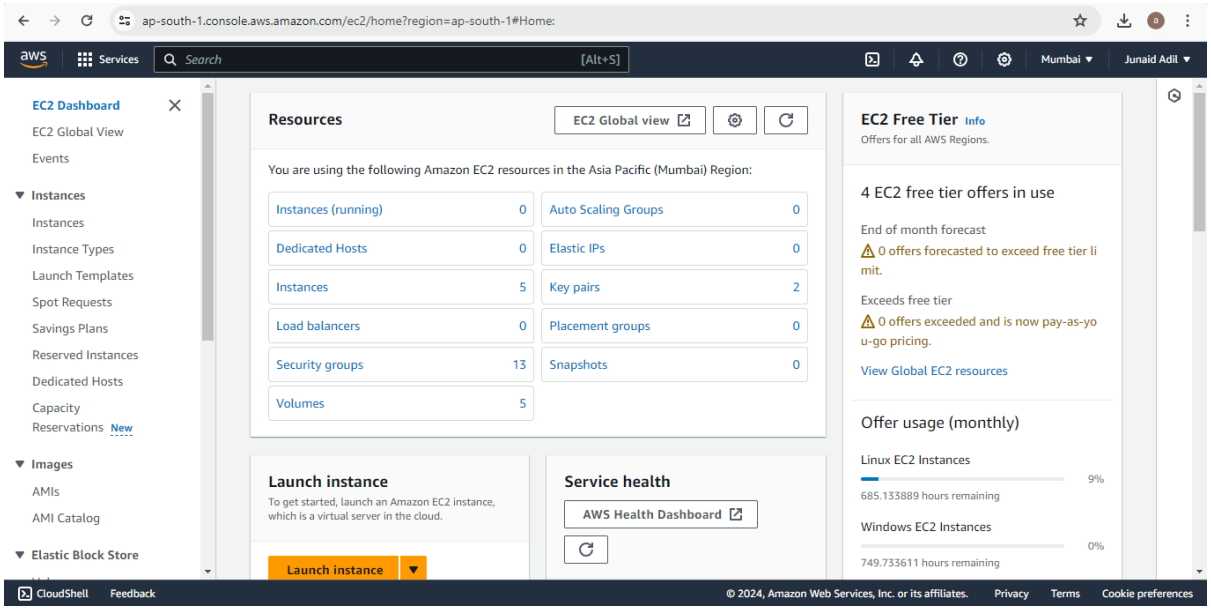


In Terminal we can see the Output. i.e, content of json file

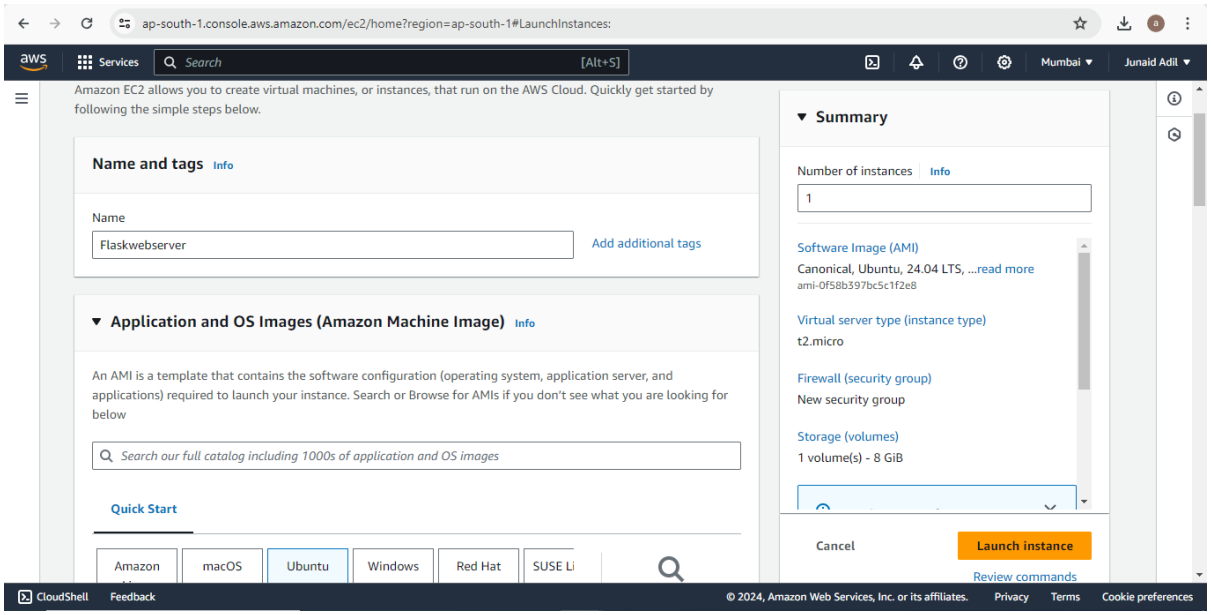


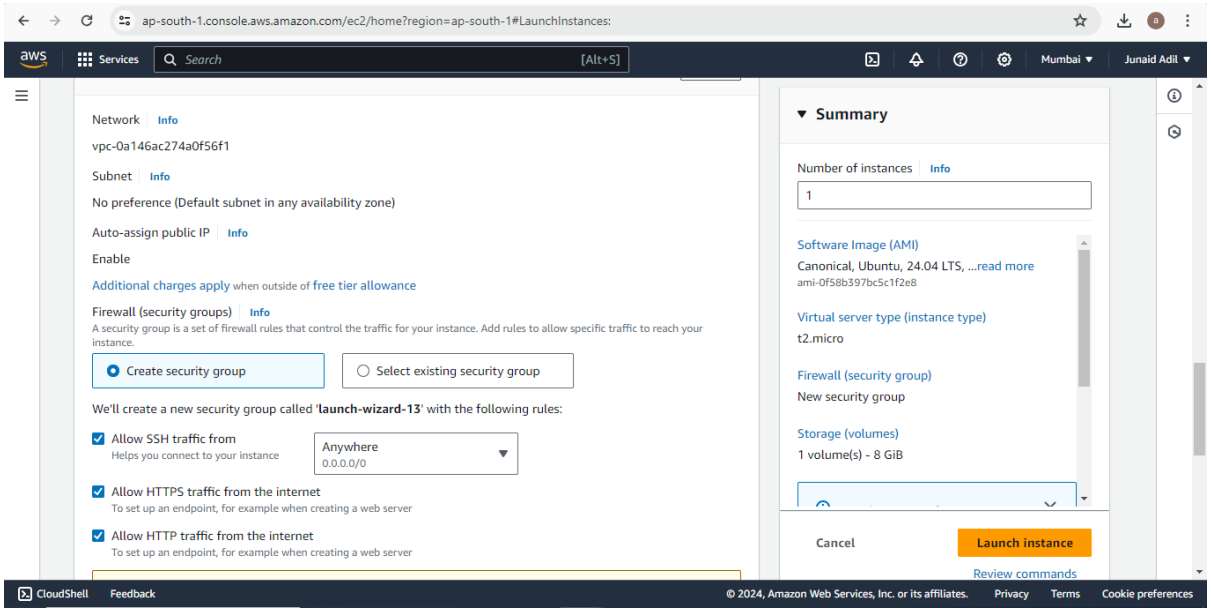
L5 - Create Python Web Application to using Flask Web Application Framework

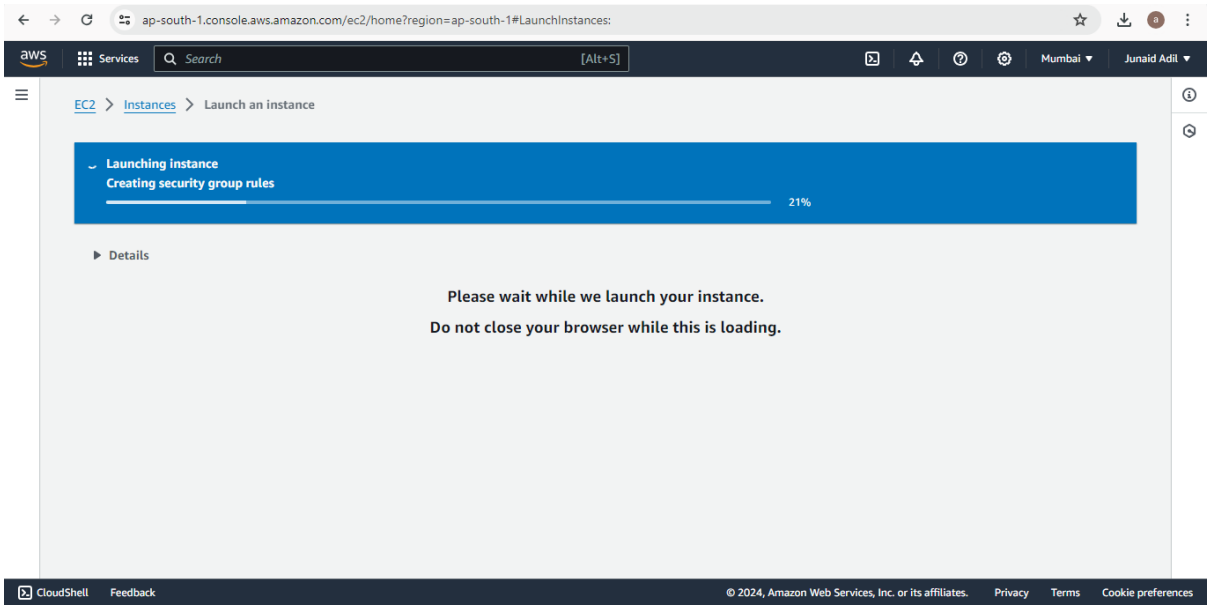
Step 1: Go to EC2 Dashboard → Launch Instance

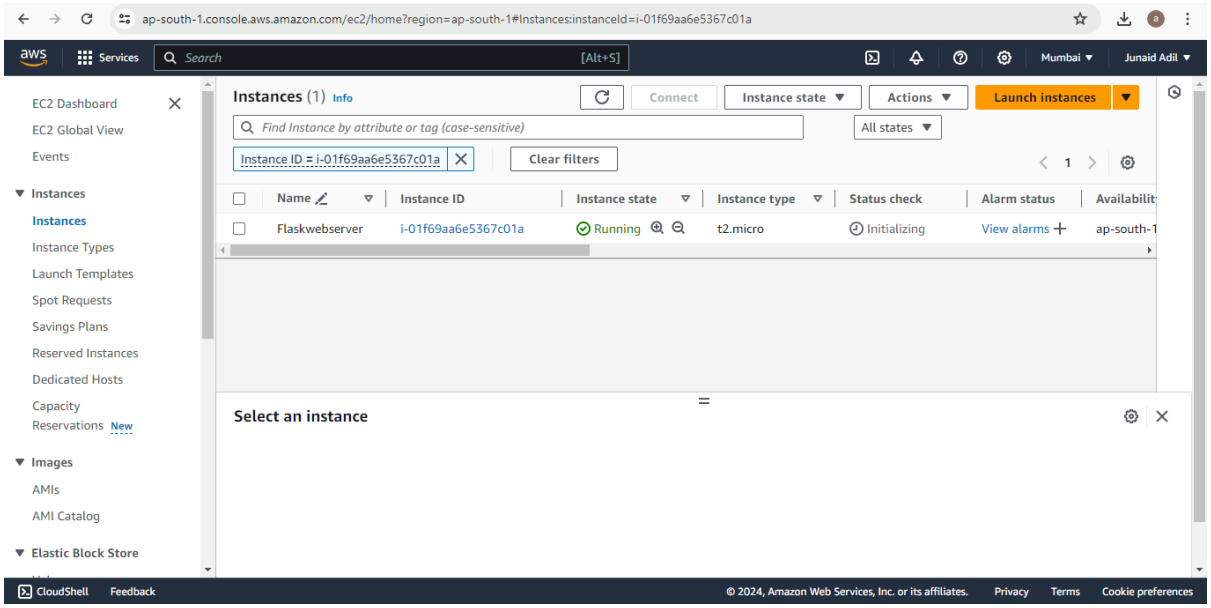


Step 2: Create an Instance " **Flaskwebserver** " and enable "Allow HTTPS traffic from the internet " and "Allow HTTP traffic from the internet " in Network settings. Then launch the instance.

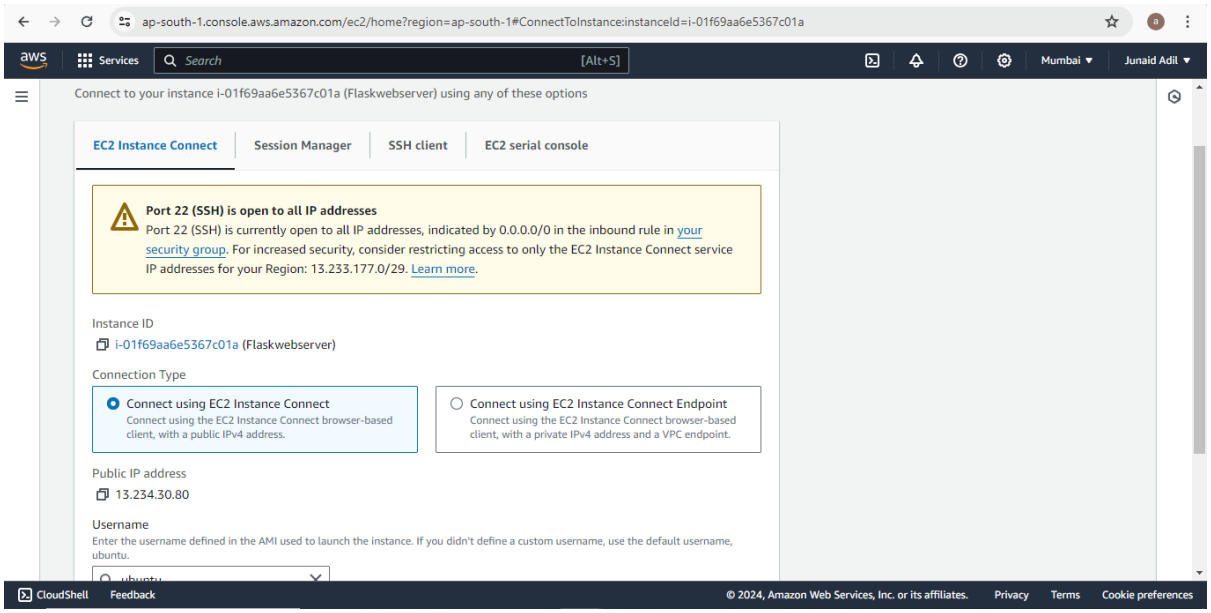


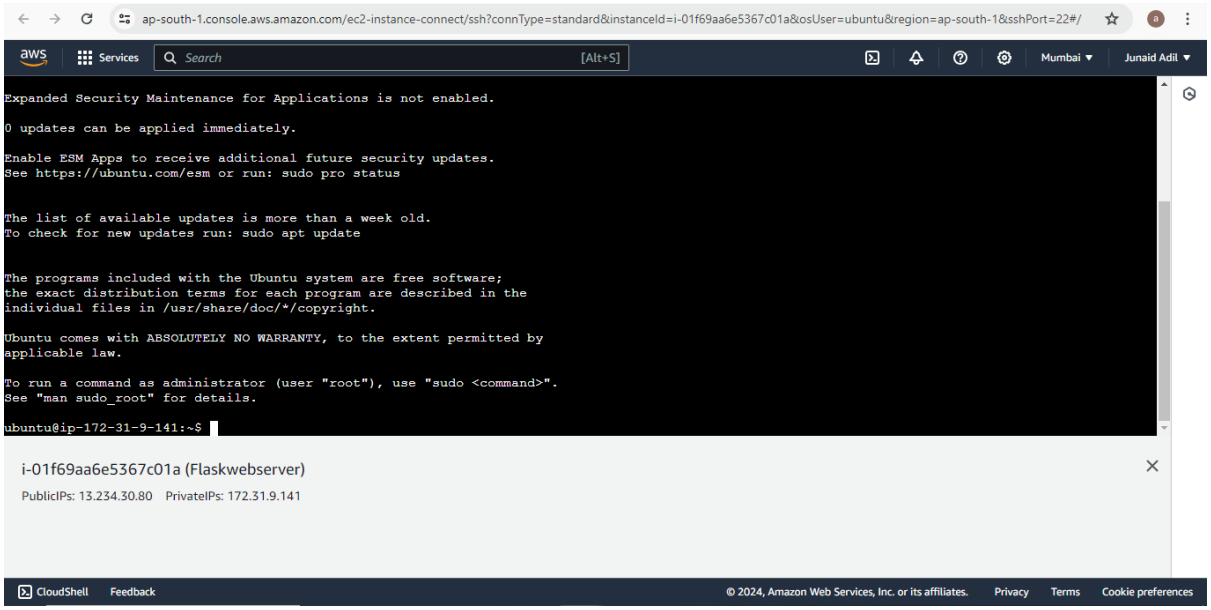

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Services
Search
[Alt+S]
Mumbai
Junaid Adil
Network Info
vpc-0a146ac274a0f56f1
Subnet Info
No preference (Default subnet in any availability zone)
Auto-assign public IP Info
Enable
Additional charges apply when outside of free tier allowance
Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.
Create security group
Select existing security group
We'll create a new security group called 'launch-wizard-13' with the following rules:
Allow SSH traffic from
Helps you connect to your instance
Anywhere
0.0.0.0/0
Allow HTTPS traffic from the internet
To set up an endpoint, for example when creating a web server
Allow HTTP traffic from the internet
To set up an endpoint, for example when creating a web server
Summary
Number of instances Info
1
Software Image (AMI)
Canonical, Ubuntu, 24.04 LTS, ...read more
ami-0f58b397bc5c1f2e8
Virtual server type (instance type)
t2.micro
Firewall (security group)
New security group
Storage (volumes)
1 volume(s) - 8 GiB
Cancel
Launch instance
Review commands
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#LaunchInstances:
Services
Search
[Alt+S]
Mumbai
Junaid Adil
EC2
Instances
Launch an instance
Launching instance
Creating security group rules
21%
Details
Please wait while we launch your instance.
Do not close your browser while this is loading.
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

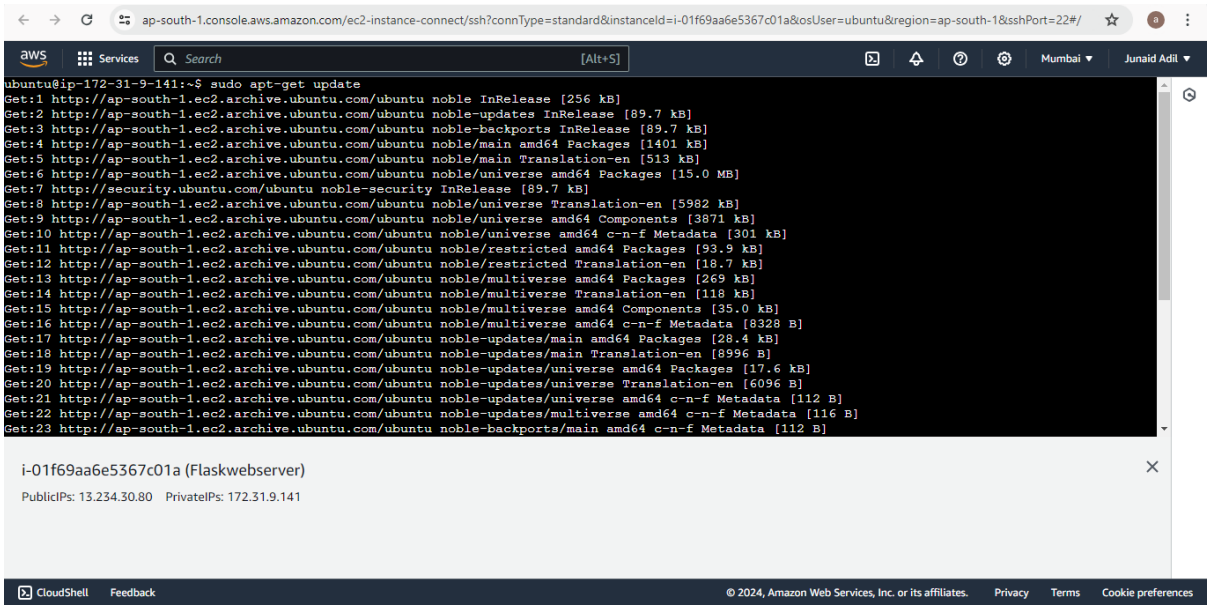


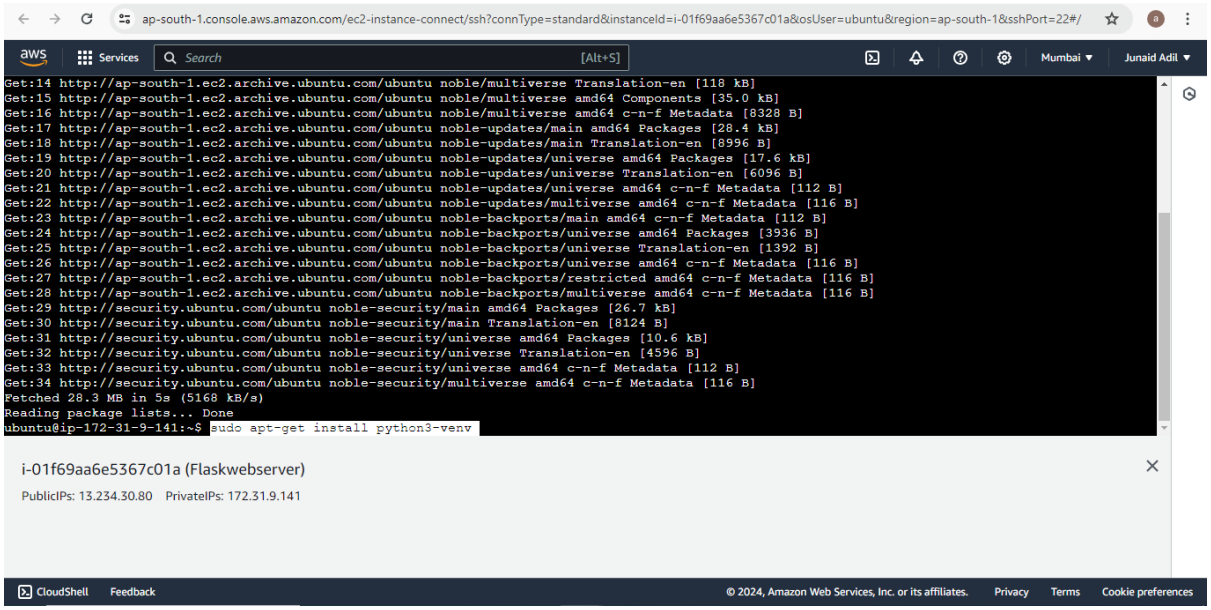
Step 3: Connect to Instance

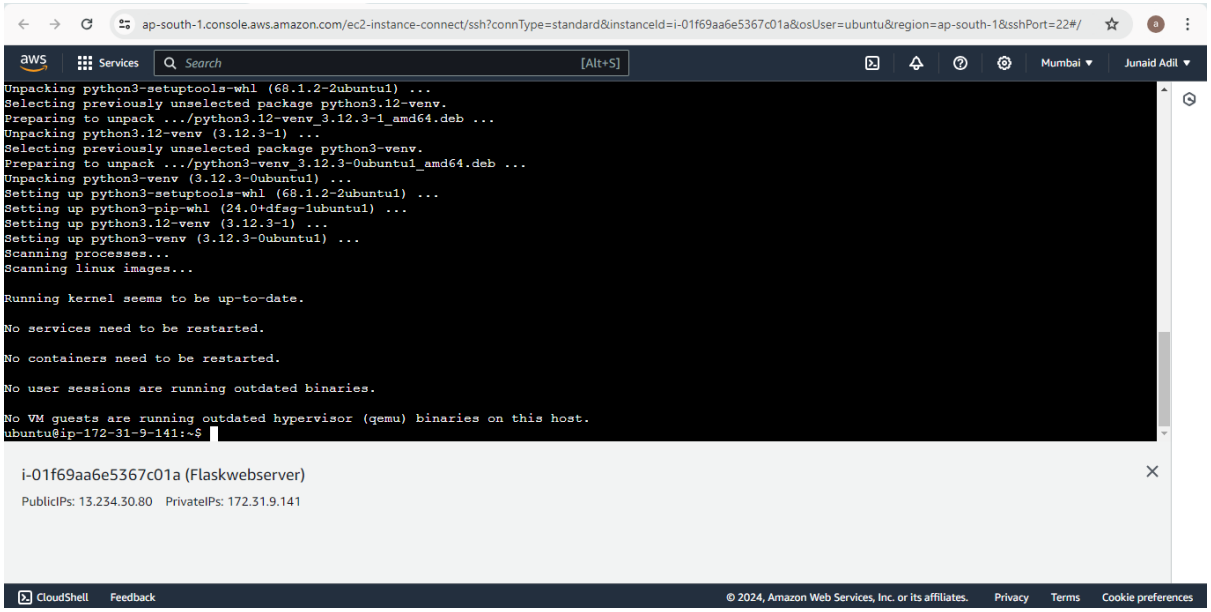




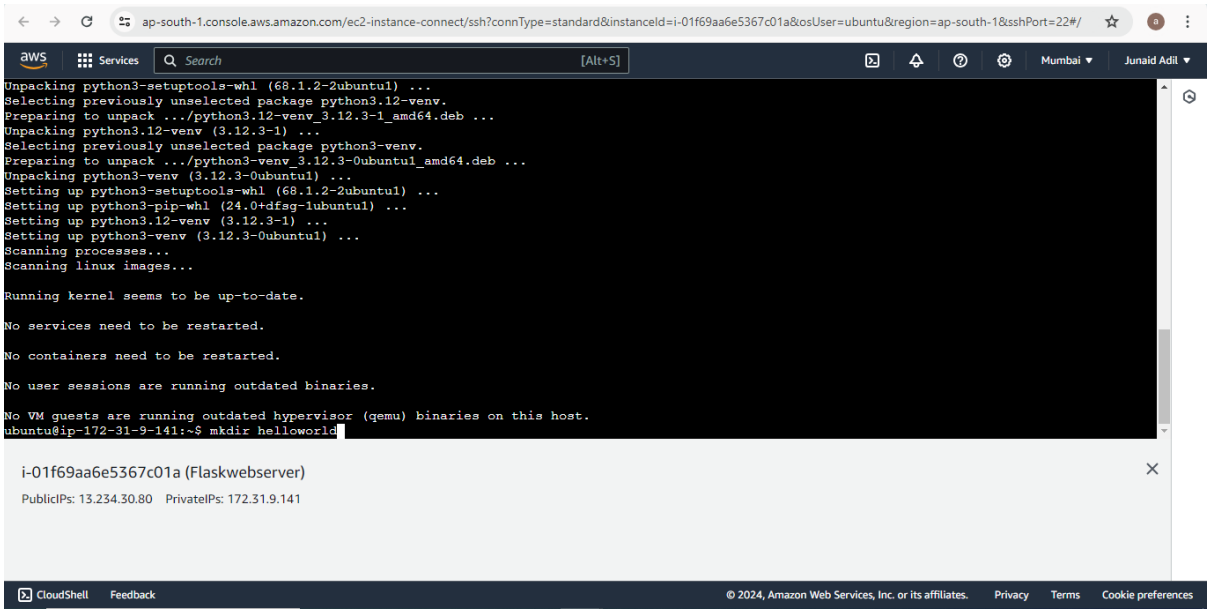
Step 4: Run the command "**sudo apt-get update**". To install Python Virtual environment run command "**sudo apt-get install python3-venv**".

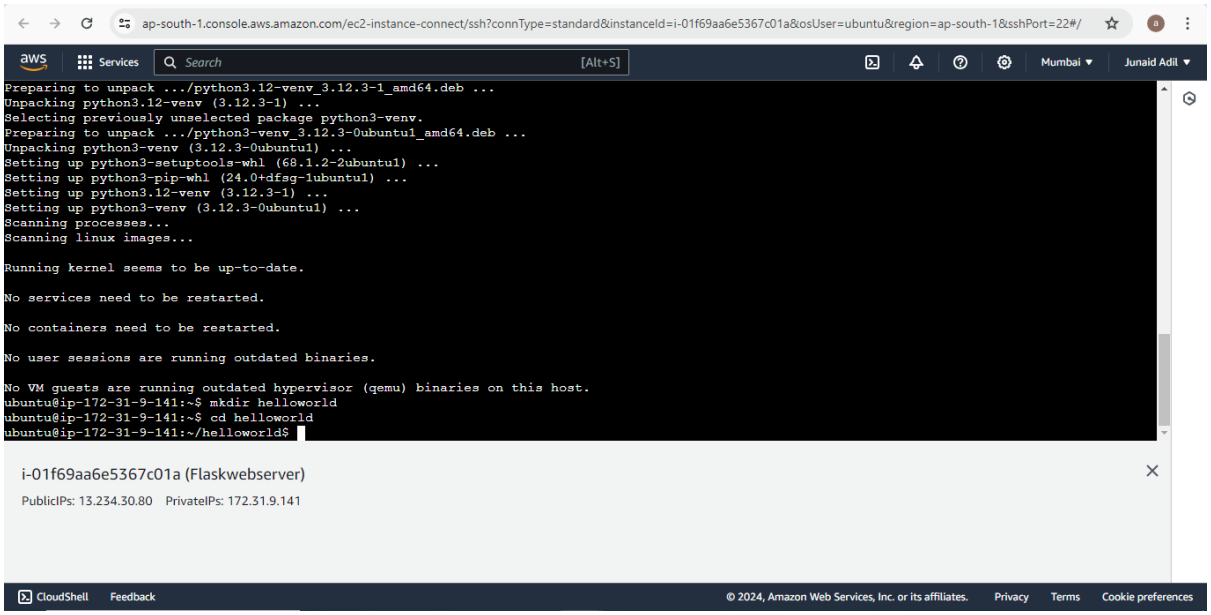



ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Junaid Adil
Get:14 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse Translation-en [118 kB]
Get:15 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 Components [35.0 kB]
Get:16 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble/multiverse amd64 c-n-f Metadata [8328 B]
Get:17 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main amd64 Packages [28.4 kB]
Get:18 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/main Translation-en [8996 B]
Get:19 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 Packages [17.6 kB]
Get:20 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe Translation-en [6096 B]
Get:21 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/universe amd64 c-n-f Metadata [112 B]
Get:22 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-updates/multiverse amd64 c-n-f Metadata [116 B]
Get:23 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/main amd64 c-n-f Metadata [112 B]
Get:24 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 Packages [3936 B]
Get:25 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe Translation-en [1392 B]
Get:26 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/universe amd64 c-n-f Metadata [116 B]
Get:27 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/restricted amd64 c-n-f Metadata [116 B]
Get:28 http://ap-south-1.ec2.archive.ubuntu.com/ubuntu noble-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:29 http://security.ubuntu.com/ubuntu noble-security/main amd64 Packages [26.7 kB]
Get:30 http://security.ubuntu.com/ubuntu noble-security/main Translation-en [8124 B]
Get:31 http://security.ubuntu.com/ubuntu noble-security/universe amd64 Packages [10.6 kB]
Get:32 http://security.ubuntu.com/ubuntu noble-security/universe Translation-en [4596 B]
Get:33 http://security.ubuntu.com/ubuntu noble-security/universe amd64 c-n-f Metadata [112 B]
Get:34 http://security.ubuntu.com/ubuntu noble-security/multiverse amd64 c-n-f Metadata [116 B]
Fetched 28.3 MB in 5s (5168 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-9-141:~$ sudo apt-get install python3-venv
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

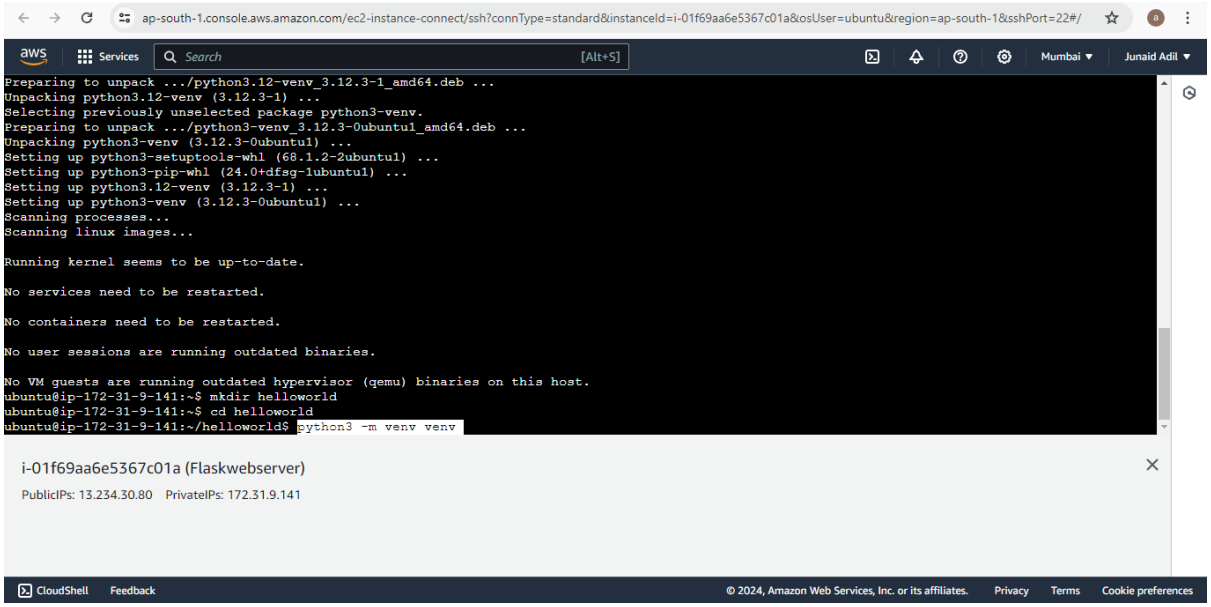

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Junaid Adil
Unpacking python3-setuptools-whl (68.1.2-2ubuntu1) ...
Selecting previously unselected package python3.12-venv.
Preparing to unpack .../python3.12-venv_3.12.3-1_amd64.deb ...
Unpacking python3.12-venv (3.12.3-1) ...
Selecting previously unselected package python3-venv.
Preparing to unpack .../python3-venv_3.12.3-0ubuntu1_amd64.deb ...
Unpacking python3-venv (3.12.3-0ubuntu1) ...
Setting up python3-setuptools-whl (68.1.2-2ubuntu1) ...
Setting up python3-pip-whl (24.0+dfsg-1ubuntu1) ...
Setting up python3.12-venv (3.12.3-1) ...
Setting up python3-venv (3.12.3-0ubuntu1) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-9-141:~$
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

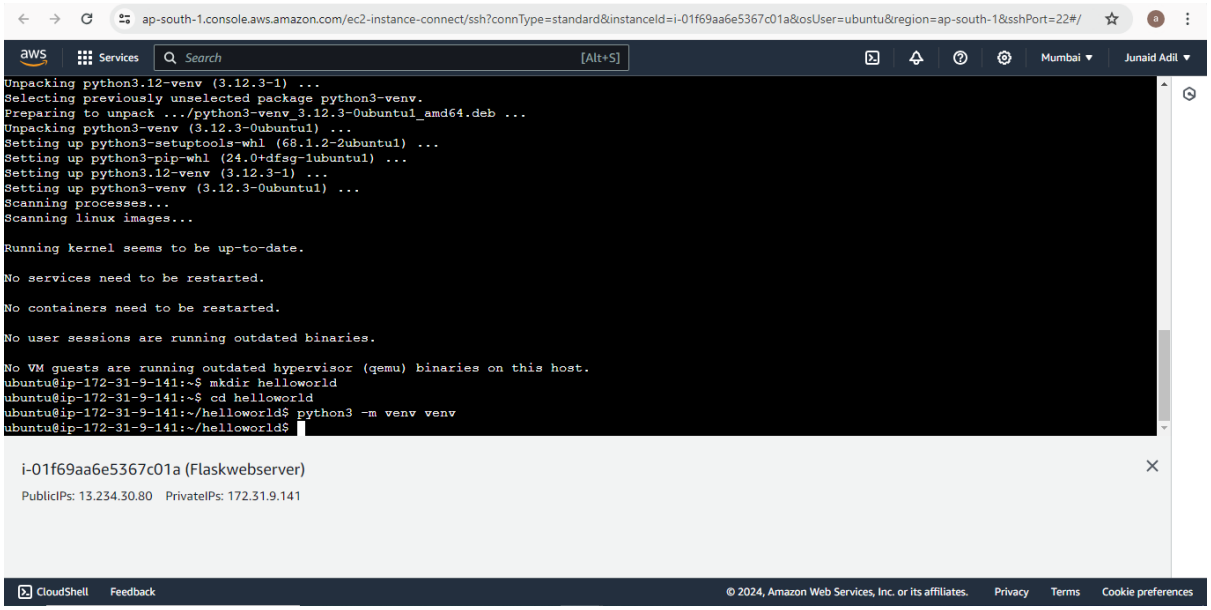
Step 5: Create the Directory “helloworld“ .



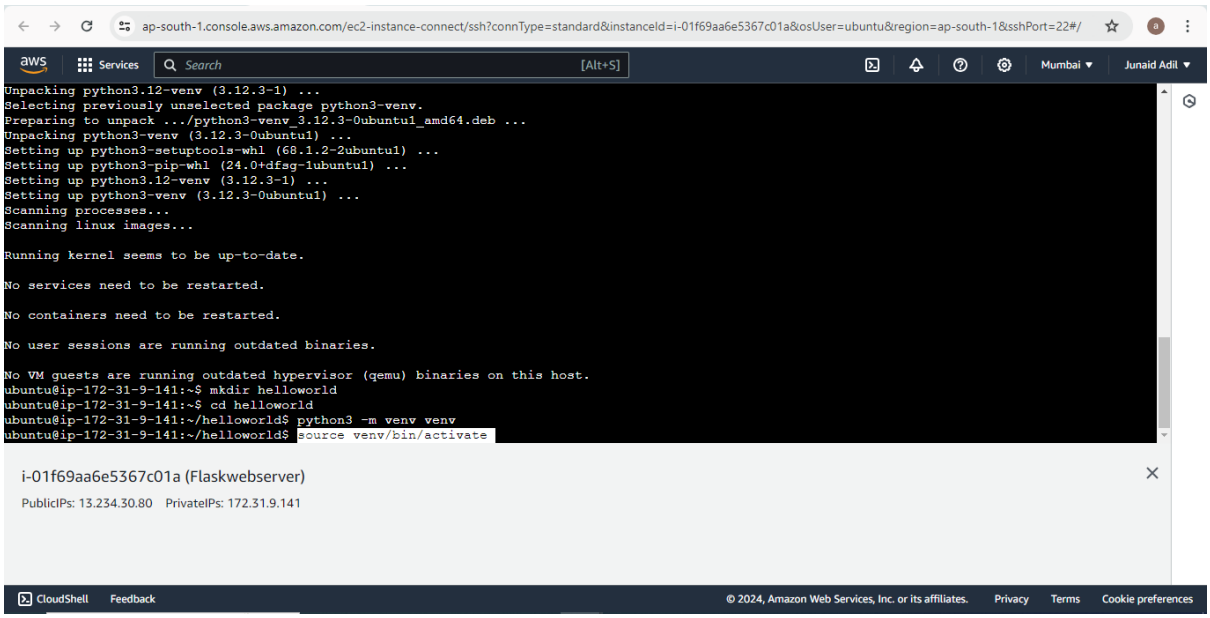


Step 6: Inside the directory create the Virtual Environment using command "**python3 -m venv venv**"

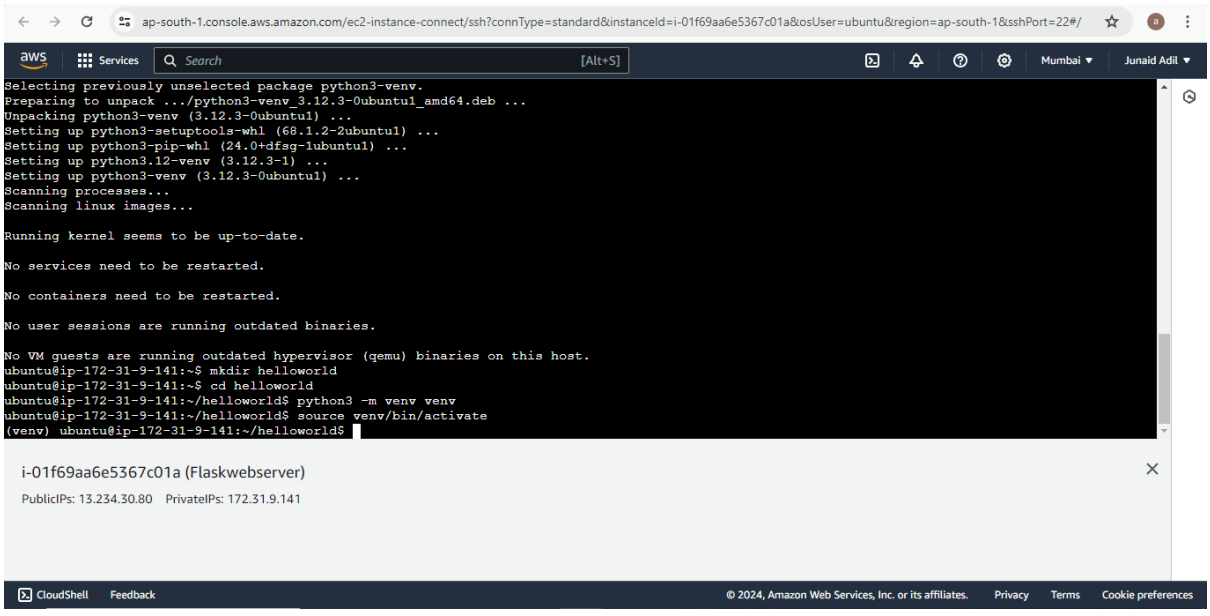




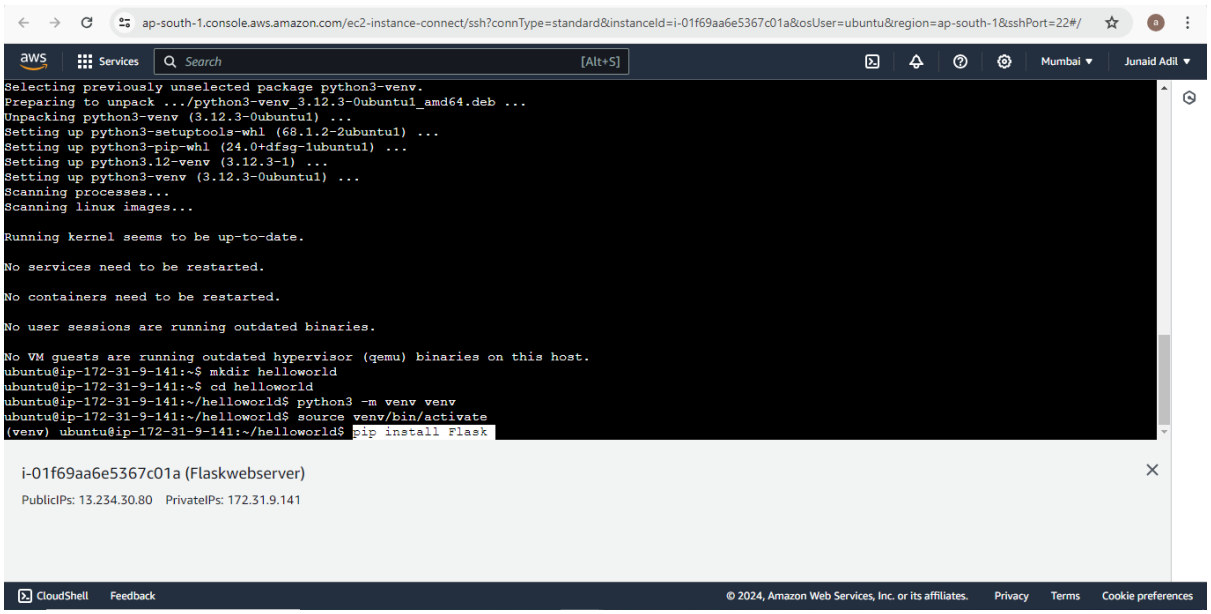
Step 7: Activate the Virtual environment using command "**source venv/bin/activate**"

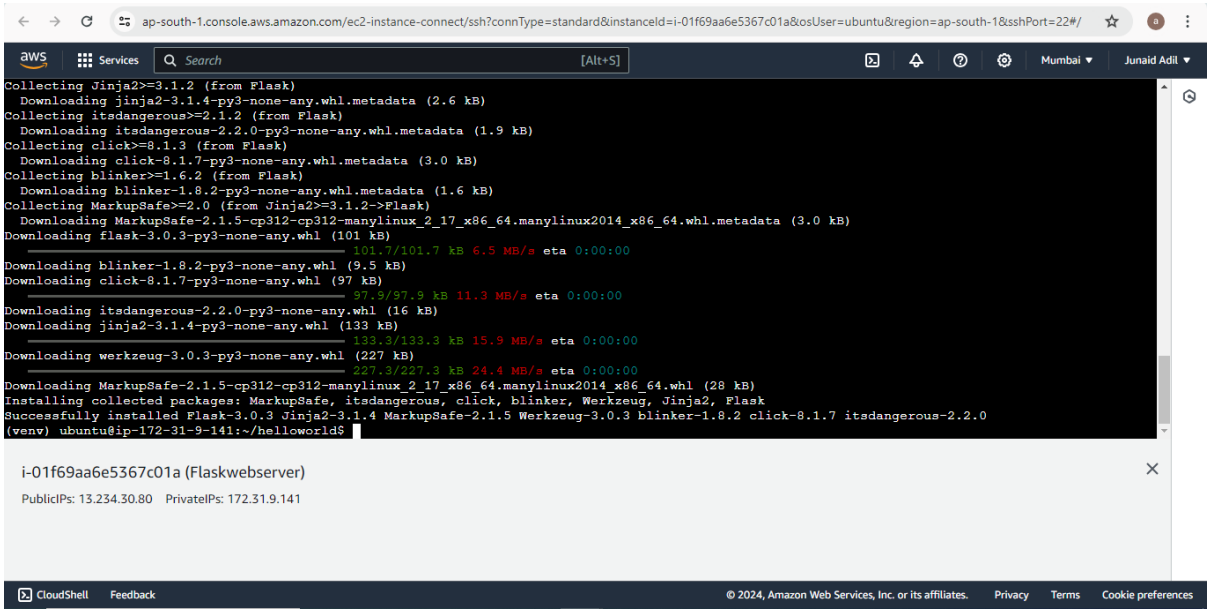


We can see the venv is activated

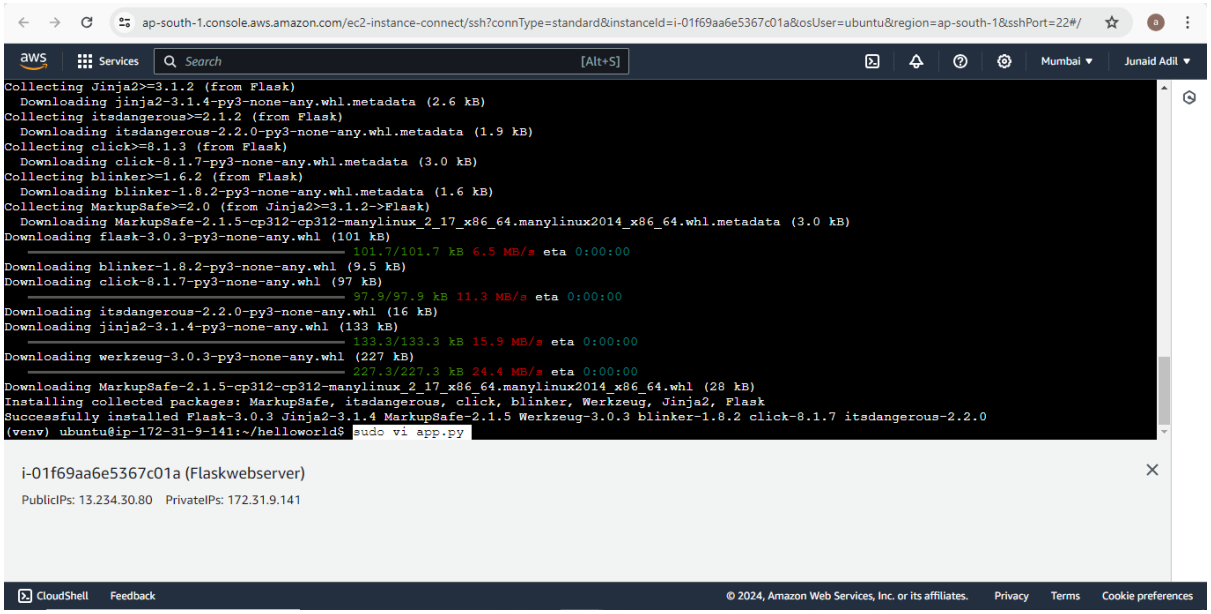


Step 8: Install the Flask using command "**pip install Flask**"

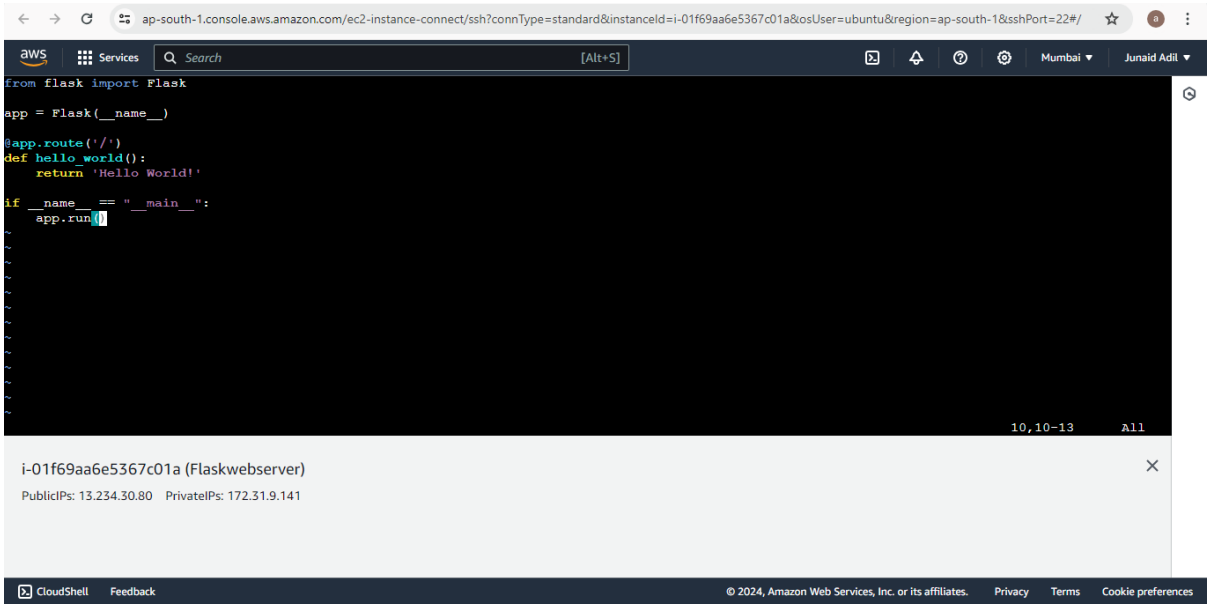




Step 9: Create the application app.py using command "**sudo vi app.py**"

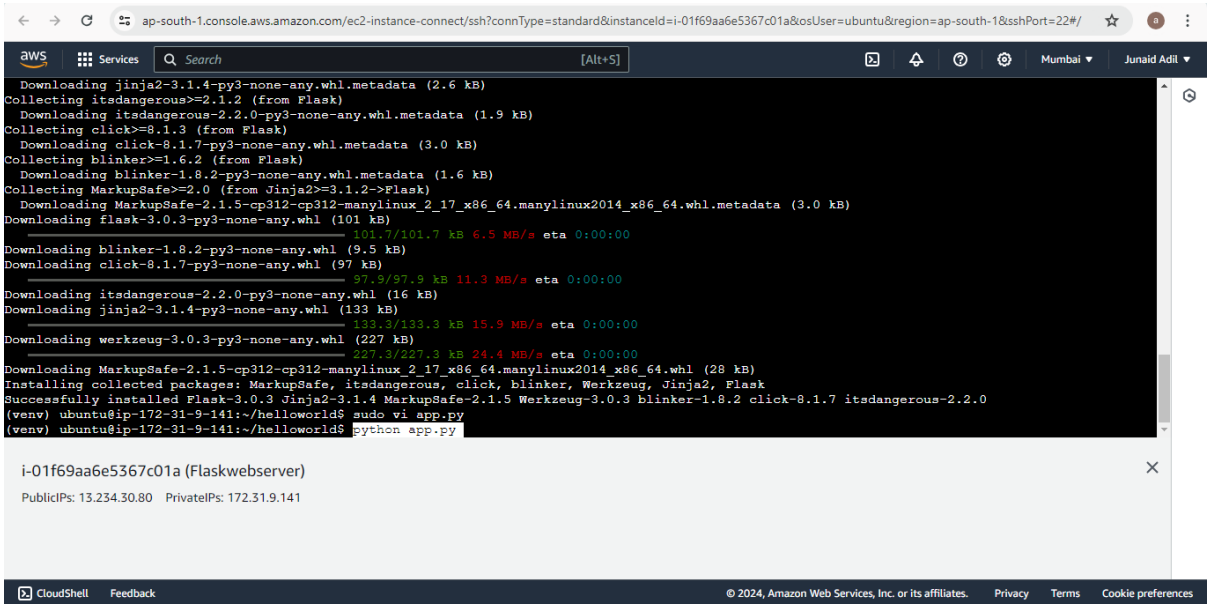


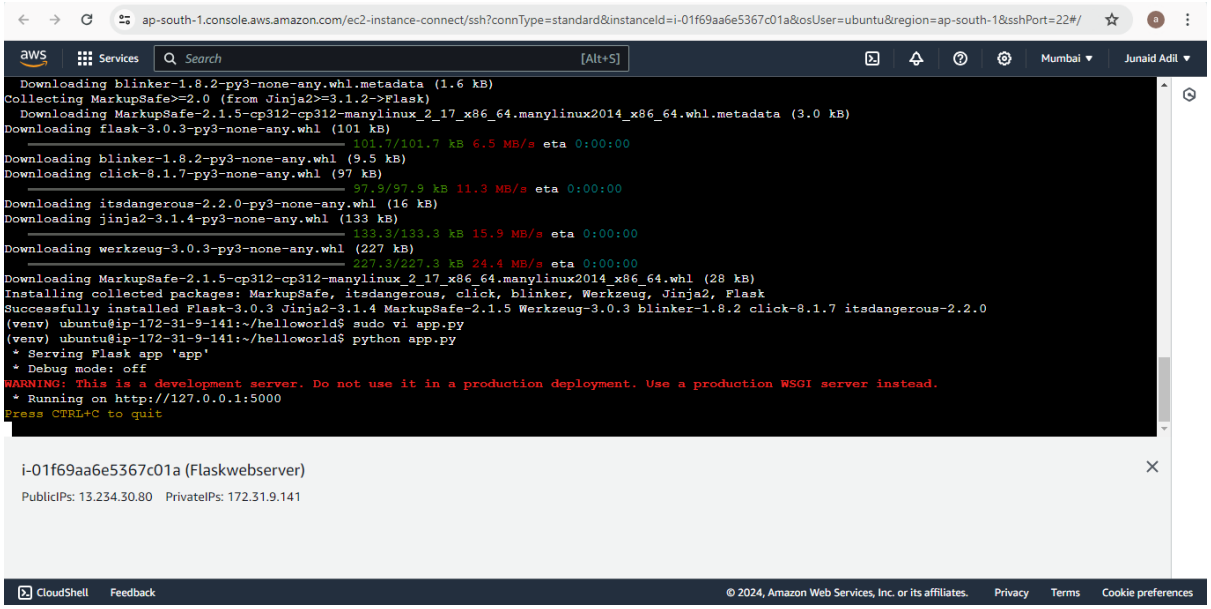
Add the commands in the file



Then save the file

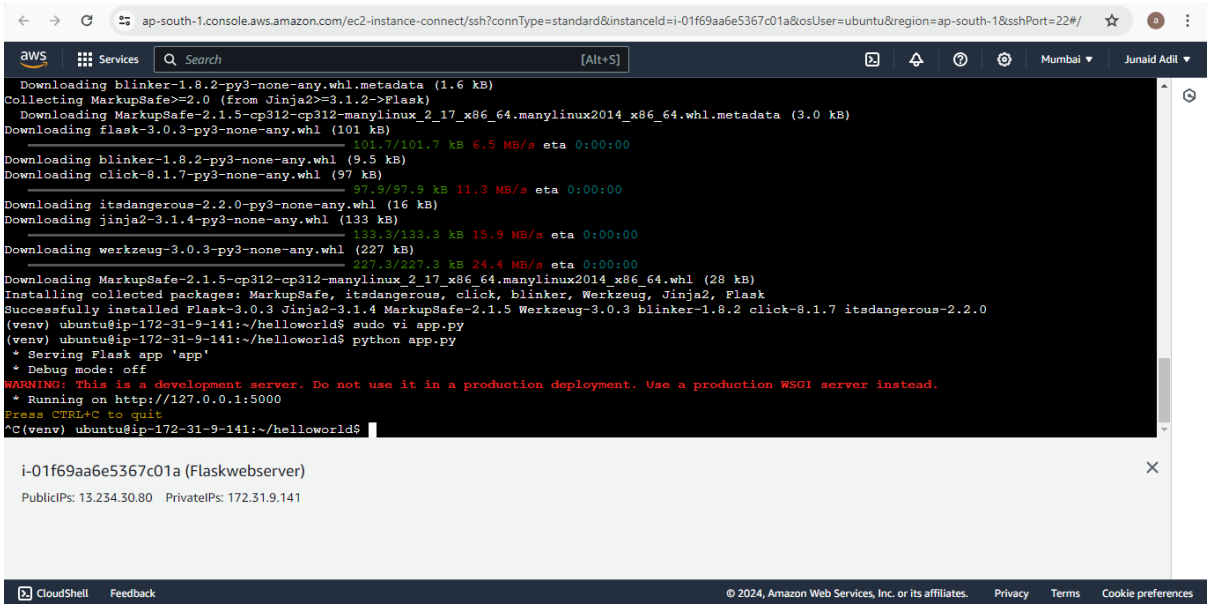
Step 10: Run the file using command "**python app.py**"



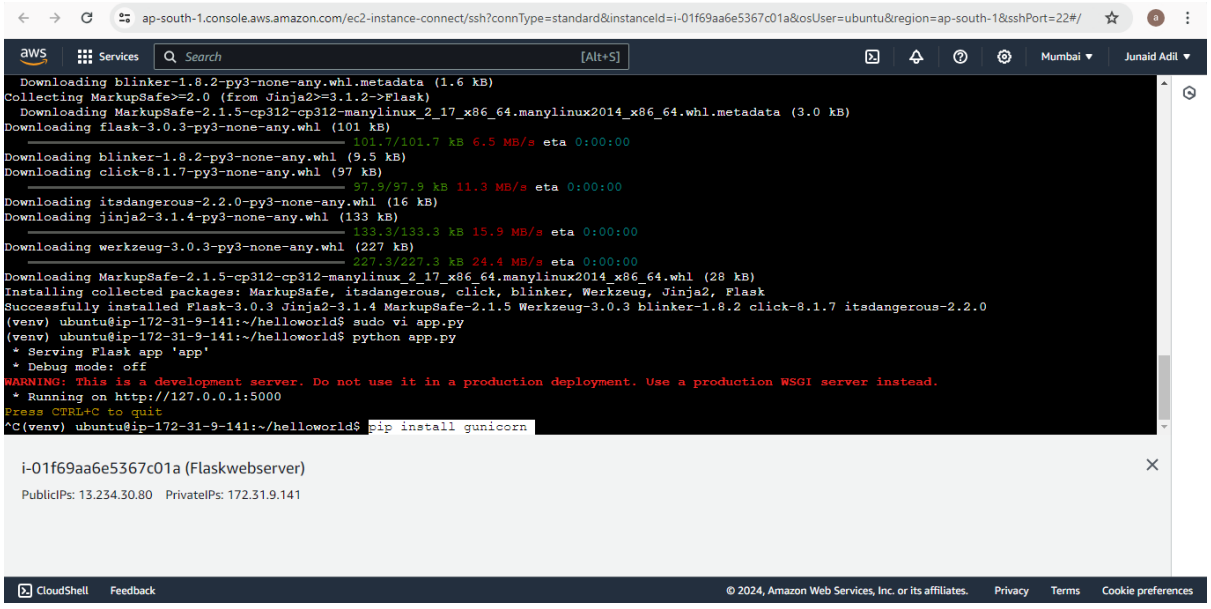


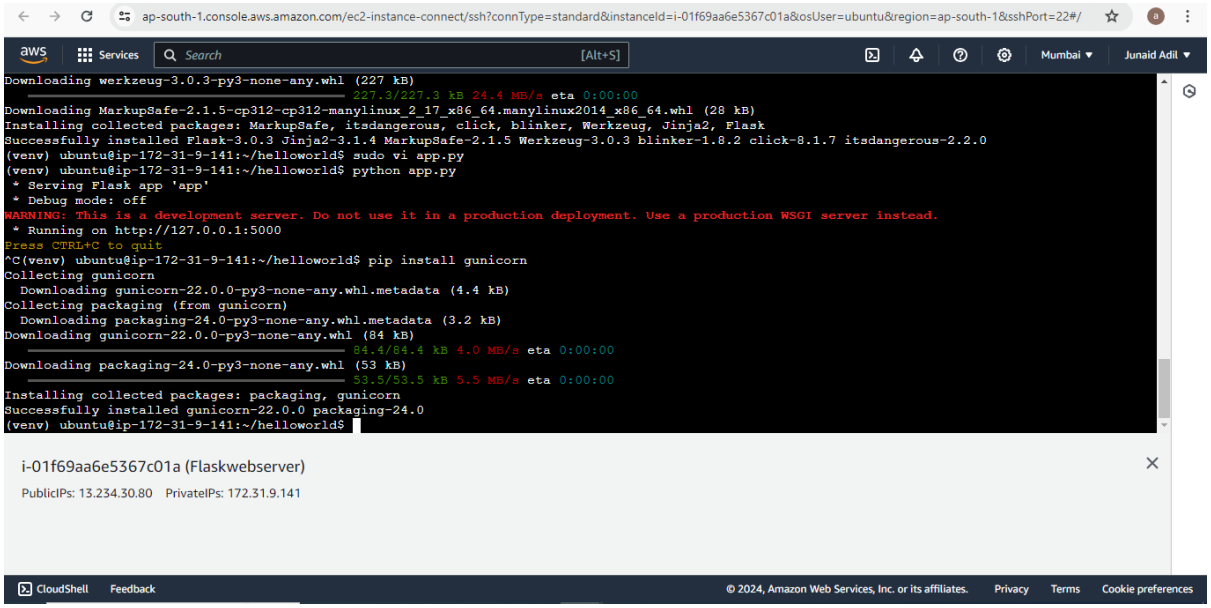
There we can see the port on which it is running Localhost:5000

Step 11: To stop it use Ctrl+c

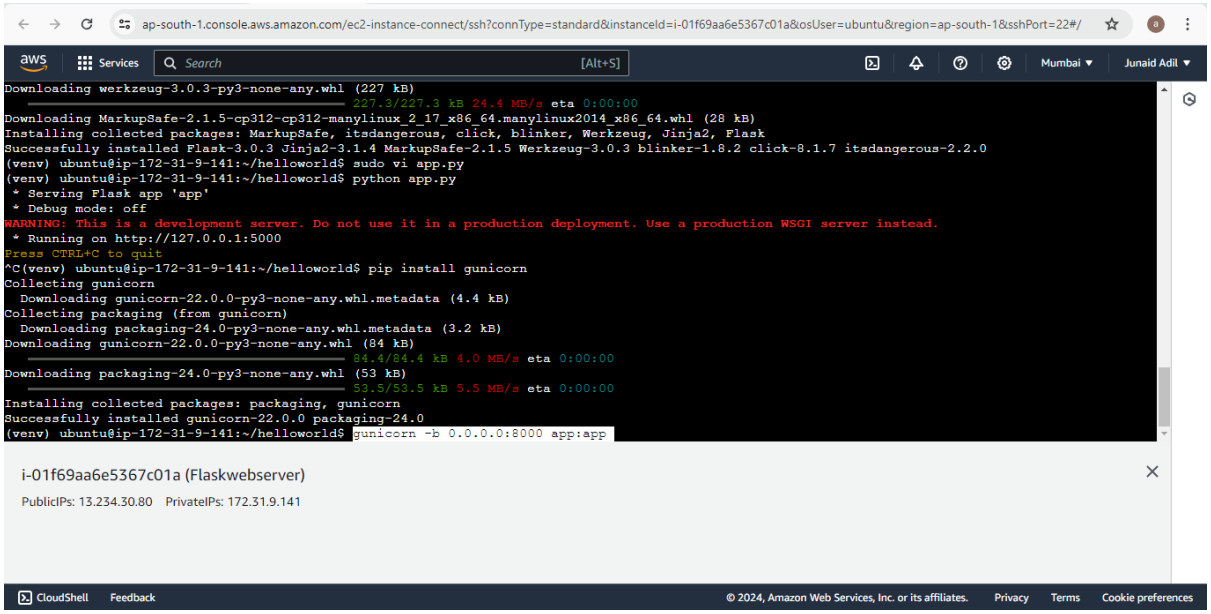


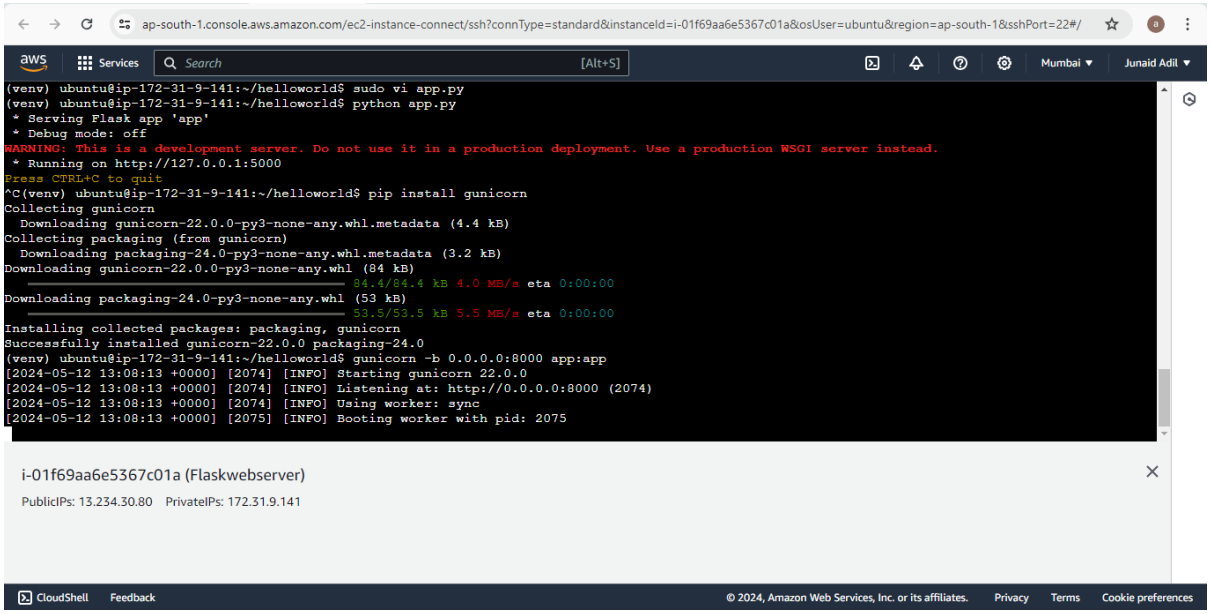
Step 12: Install gunicorn server using command "**pip install gunicorn**"

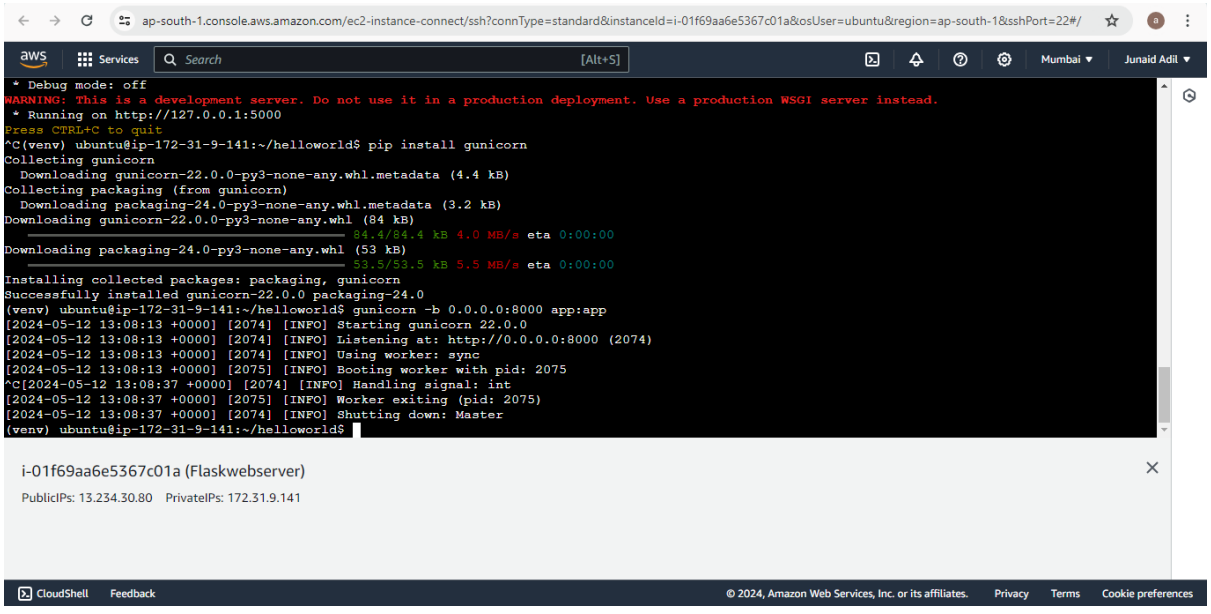




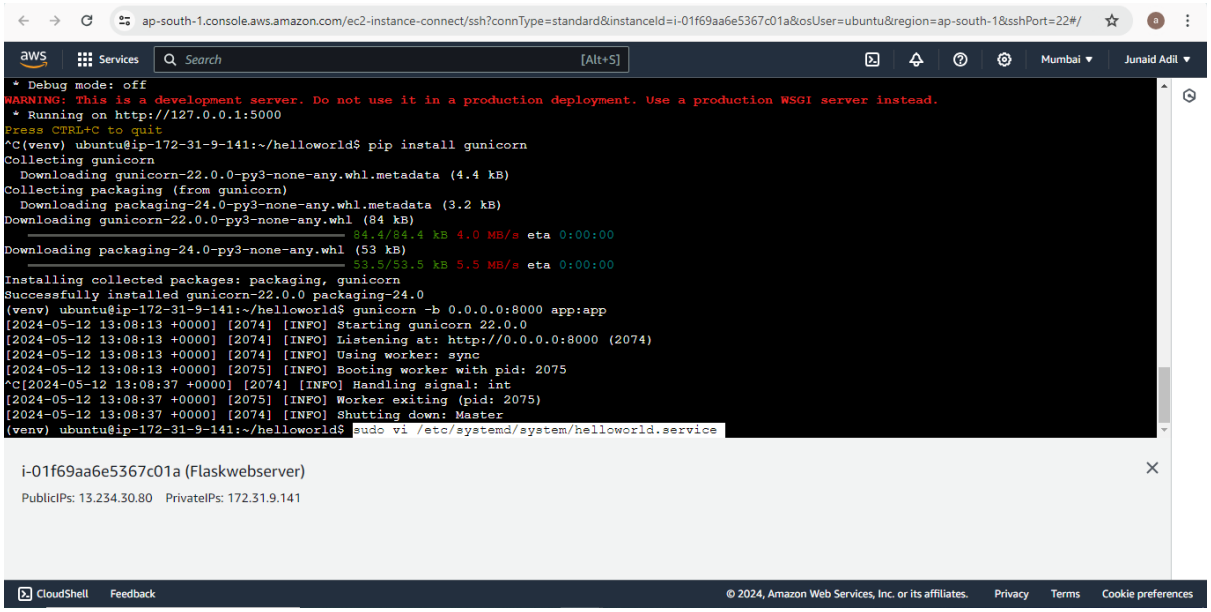
Step 13: Then run the server using command "**gunicorn -b 0.0.0.0:8000 app:app**"



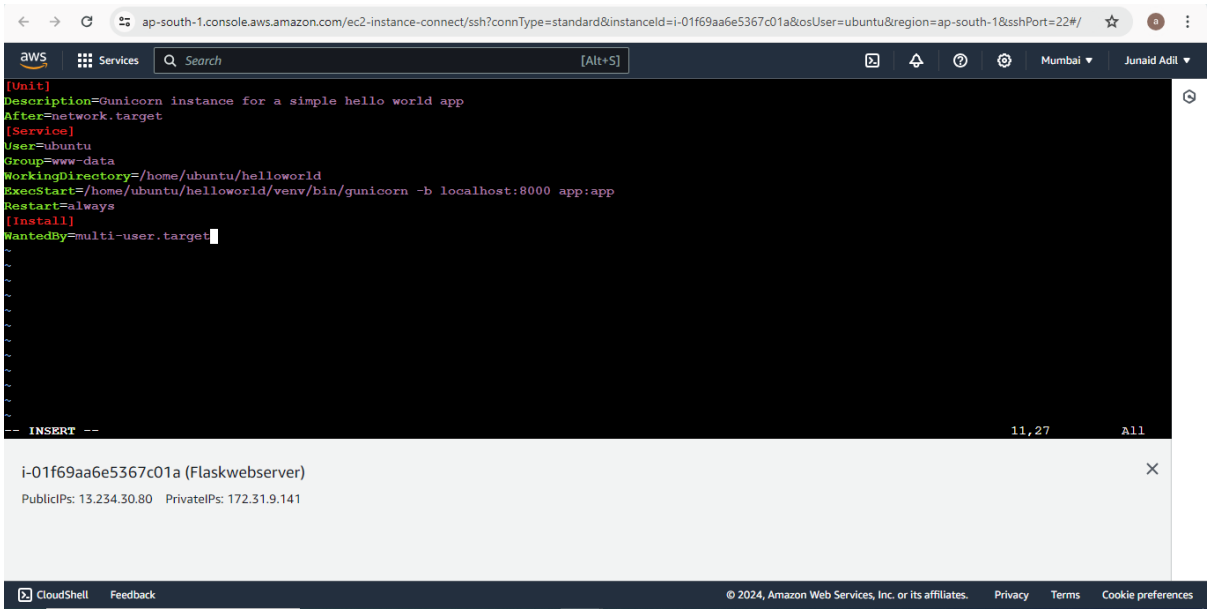

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Junaid Adil
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ sudo vi app.py
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ python app.py
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
^C(venv) ubuntu@ip-172-31-9-141:~/helloworld$ pip install gunicorn
Collecting gunicorn
Downloading gunicorn-22.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting packaging (from gunicorn)
Downloading packaging-24.0-py3-none-any.whl.metadata (3.2 kB)
Downloading gunicorn-22.0.0-py3-none-any.whl (84 kB)
84.4/84.4 kB 4.0 MB/s eta 0:00:00
Downloading packaging-24.0-py3-none-any.whl (53 kB)
53.5/53.5 kB 5.5 MB/s eta 0:00:00
Installing collected packages: packaging, gunicorn
Successfully installed gunicorn-22.0.0 packaging-24.0
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ gunicorn -b 0.0.0.0:8000 app:app
[2024-05-12 13:08:13 +0000] [2074] [INFO] Starting gunicorn 22.0.0
[2024-05-12 13:08:13 +0000] [2074] [INFO] Listening at: http://0.0.0.0:8000 (2074)
[2024-05-12 13:08:13 +0000] [2074] [INFO] Using worker: sync
[2024-05-12 13:08:13 +0000] [2075] [INFO] Booting worker with pid: 2075
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
aws
Services
Search
[Alt+S]
Mumbai
Junaid Adil
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
^C(venv) ubuntu@ip-172-31-9-141:~/helloworld$ pip install gunicorn
Collecting gunicorn
Downloading gunicorn-22.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting packaging (from gunicorn)
Downloading packaging-24.0-py3-none-any.whl.metadata (3.2 kB)
Downloading gunicorn-22.0.0-py3-none-any.whl (84 kB)
84.4/84.4 kB 4.0 MB/s eta 0:00:00
Downloading packaging-24.0-py3-none-any.whl (53 kB)
53.5/53.5 kB 5.5 MB/s eta 0:00:00
Installing collected packages: packaging, gunicorn
Successfully installed gunicorn-22.0.0 packaging-24.0
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ gunicorn -b 0.0.0.0:8000 app:app
[2024-05-12 13:08:13 +0000] [2074] [INFO] Starting gunicorn 22.0.0
[2024-05-12 13:08:13 +0000] [2074] [INFO] Listening at: http://0.0.0.0:8000 (2074)
[2024-05-12 13:08:13 +0000] [2074] [INFO] Using worker: sync
[2024-05-12 13:08:13 +0000] [2075] [INFO] Booting worker with pid: 2075
^C[2024-05-12 13:08:37 +0000] [2074] [INFO] Handling signal: int
[2024-05-12 13:08:37 +0000] [2075] [INFO] Worker exiting (pid: 2075)
[2024-05-12 13:08:37 +0000] [2074] [INFO] Shutting down: Master
(venv) ubuntu@ip-172-31-9-141:~/helloworld$
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

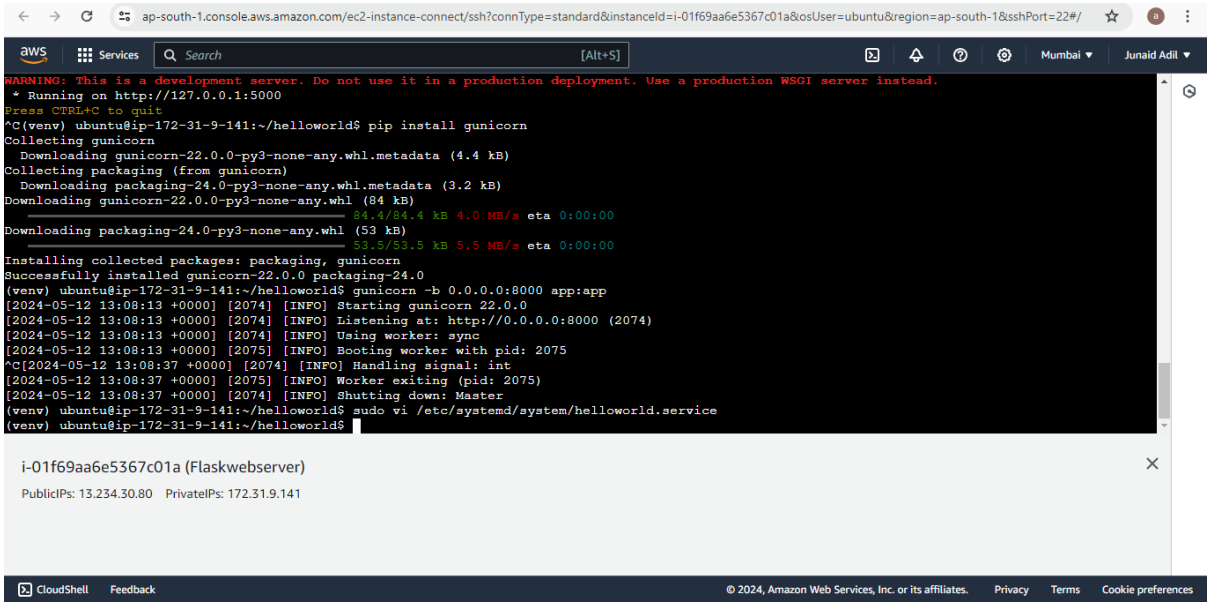
Step 14: We have to create the helloworld.service file for the server using command "**sudo vi /etc/systemd/system/helloworld.service**"



Then add the commands in the file

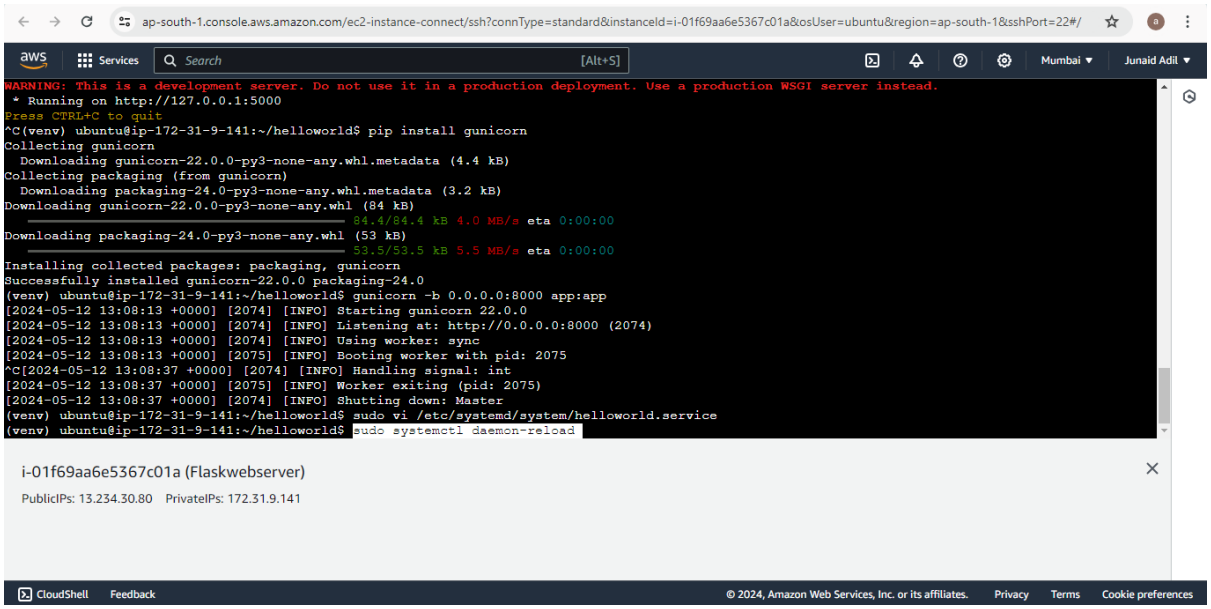


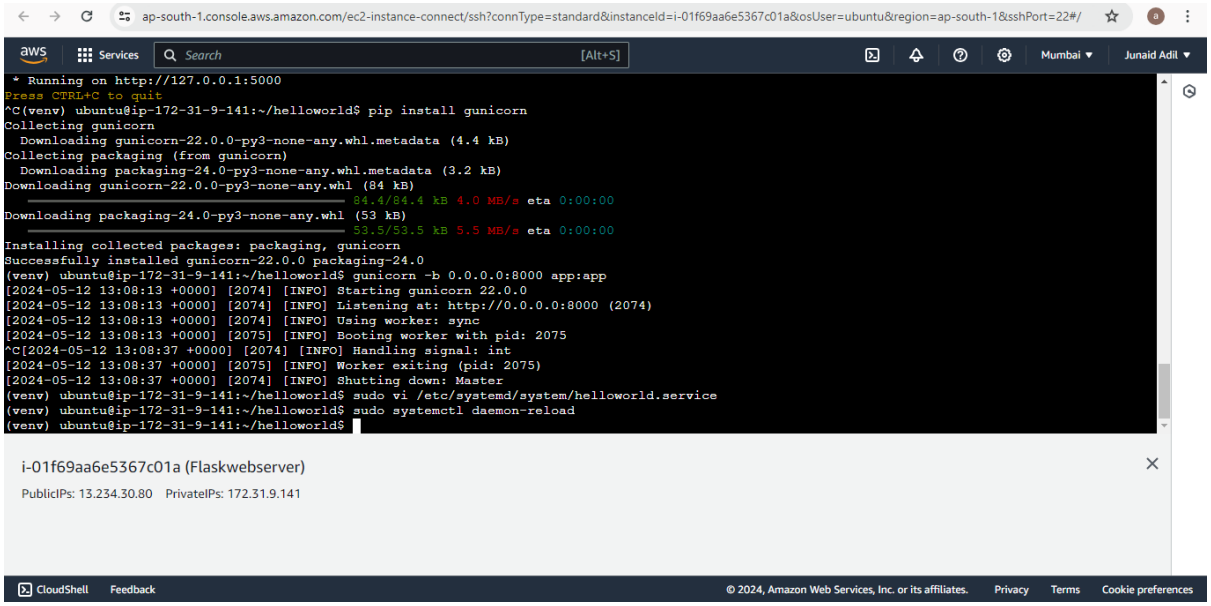
Save the file :wq!



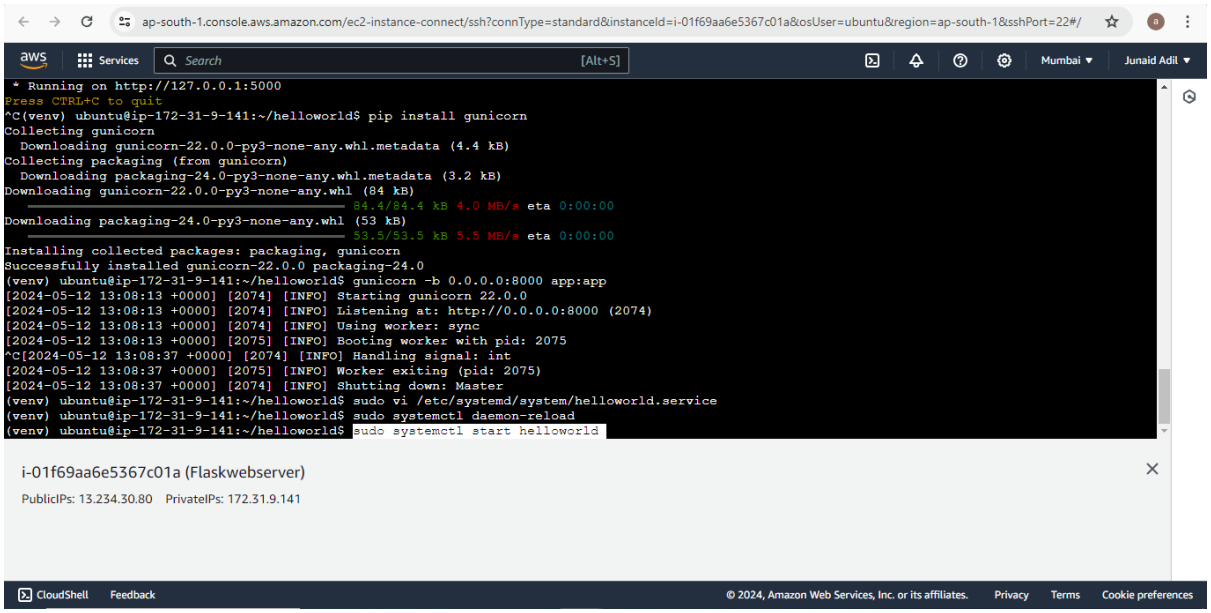
<u>Step 15:</u> We have to start and enable the server. So run the commands "**sudo systemctl daemon-reload**", "**sudo systemctl start helloworld**", "**sudo systemctl enable helloworld**"

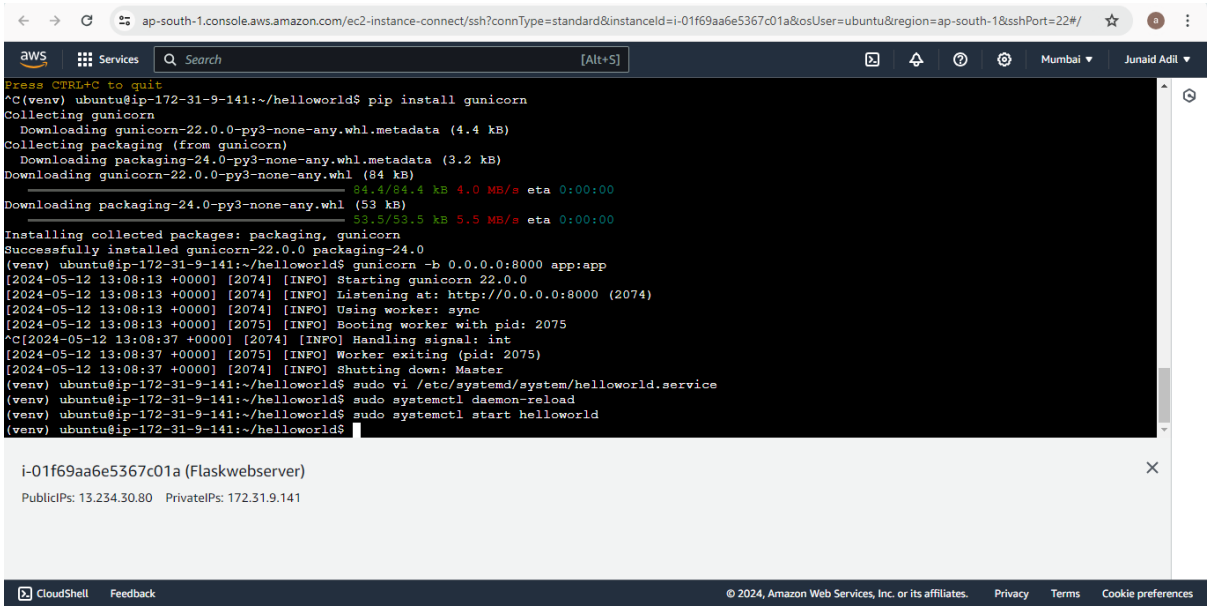
Run command : "**sudo systemctl daemon-reload**".



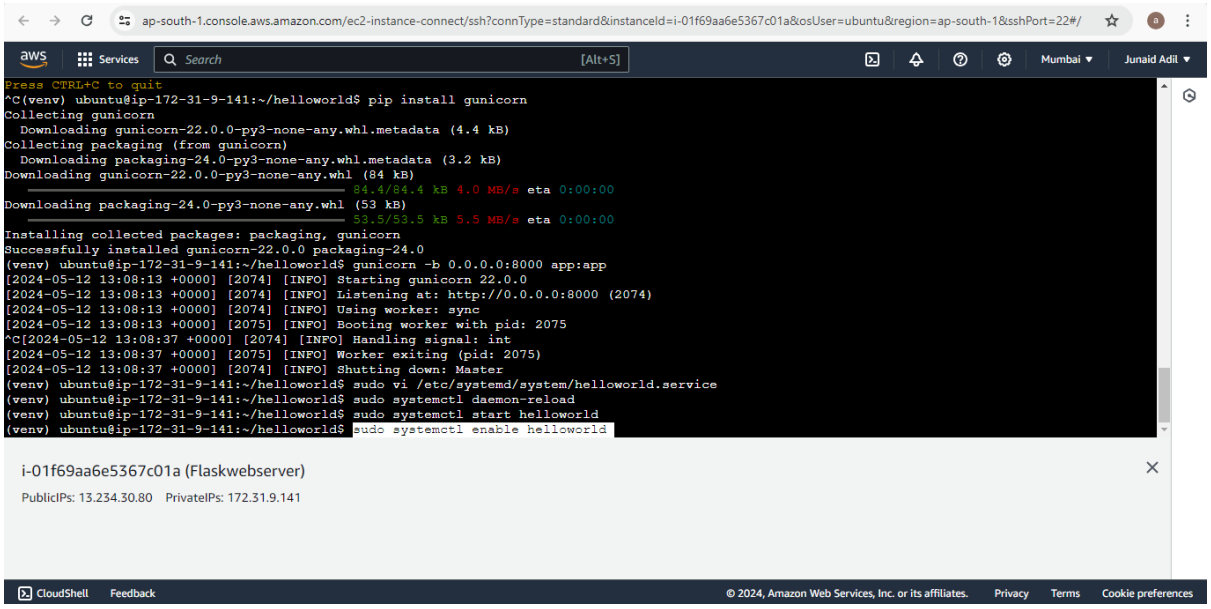


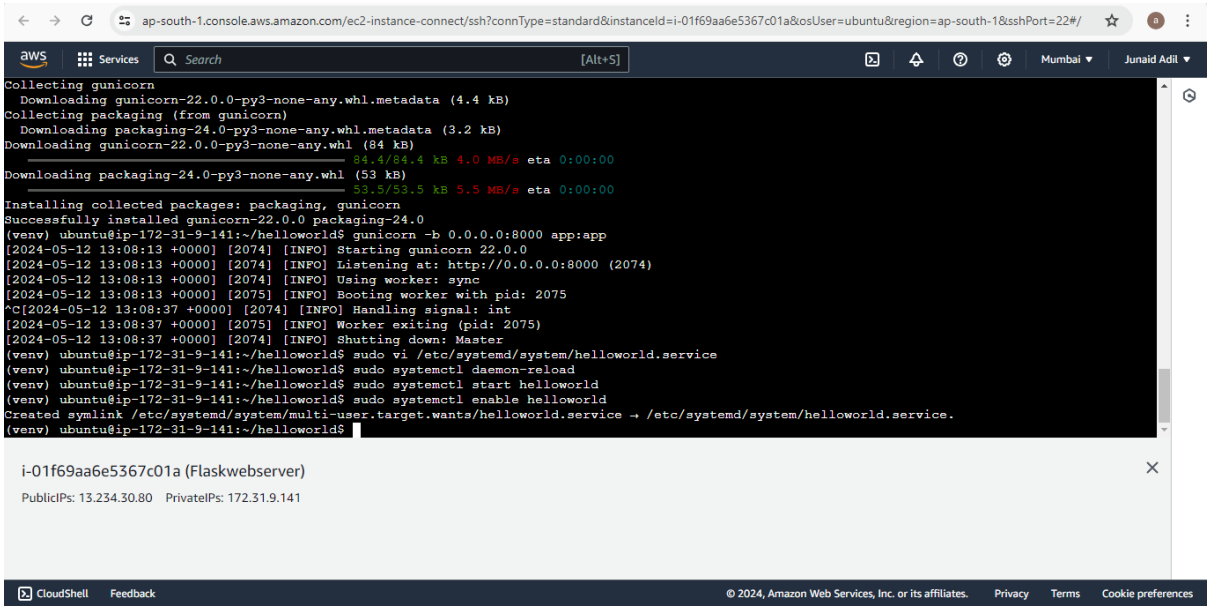
Run the command : “**sudo systemctl start helloworld**”



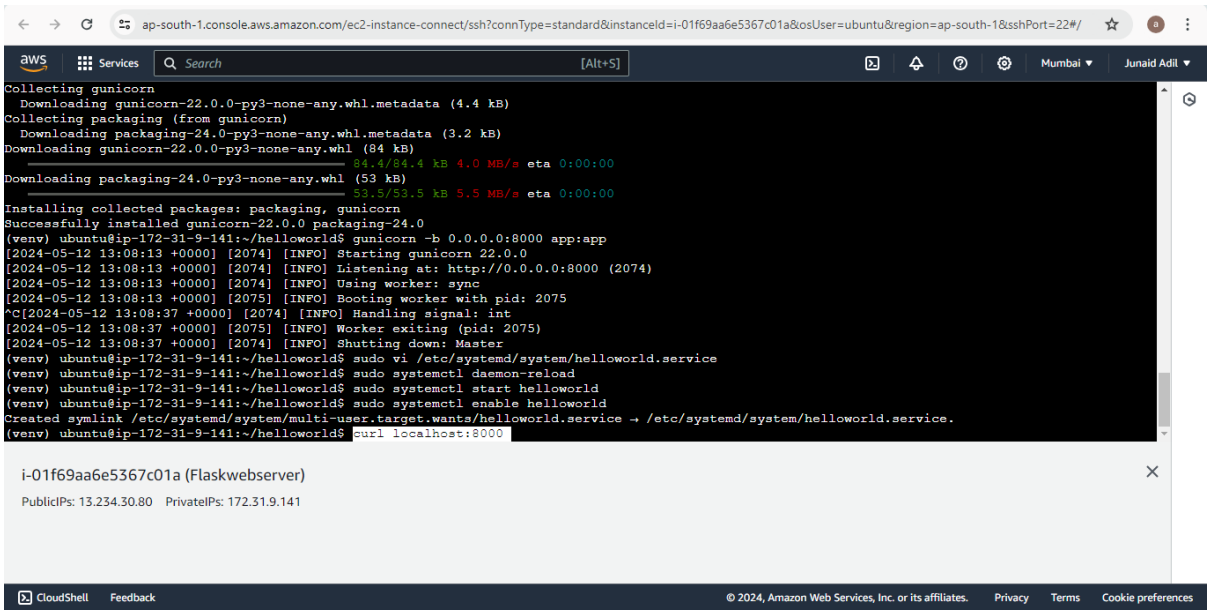


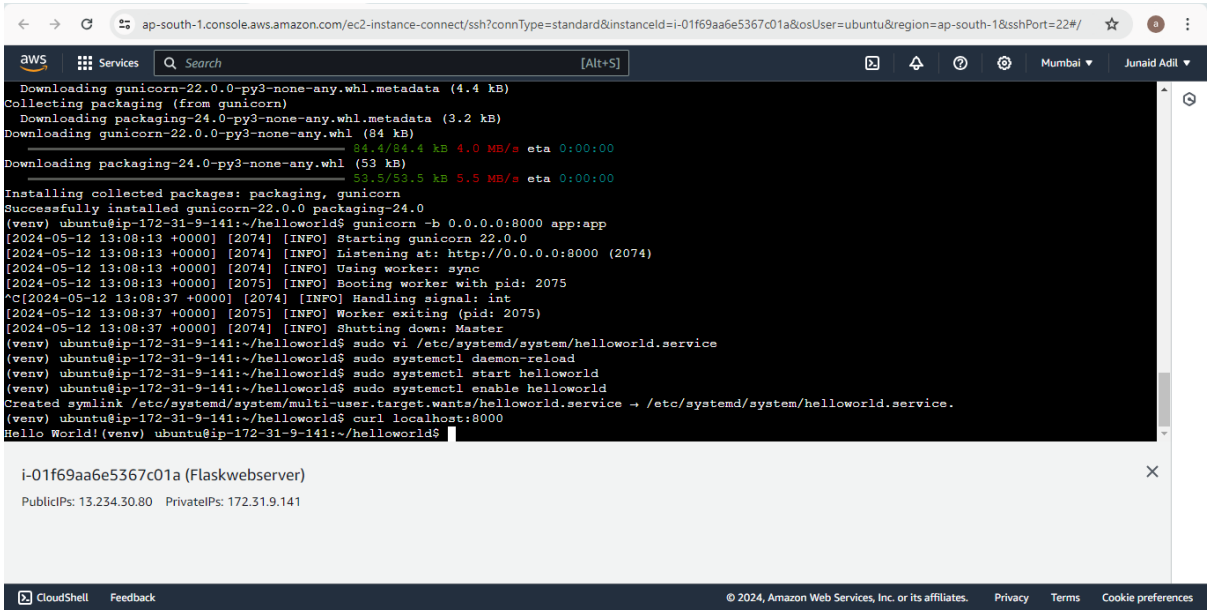
Run the command: “**sudo systemctl enable helloworld**”



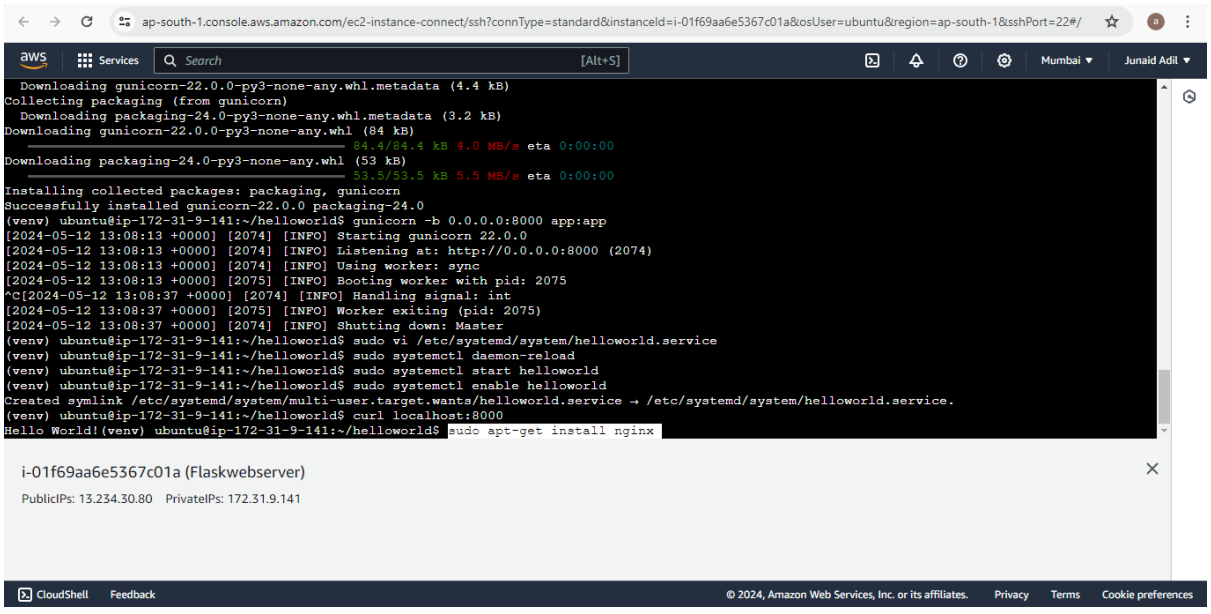


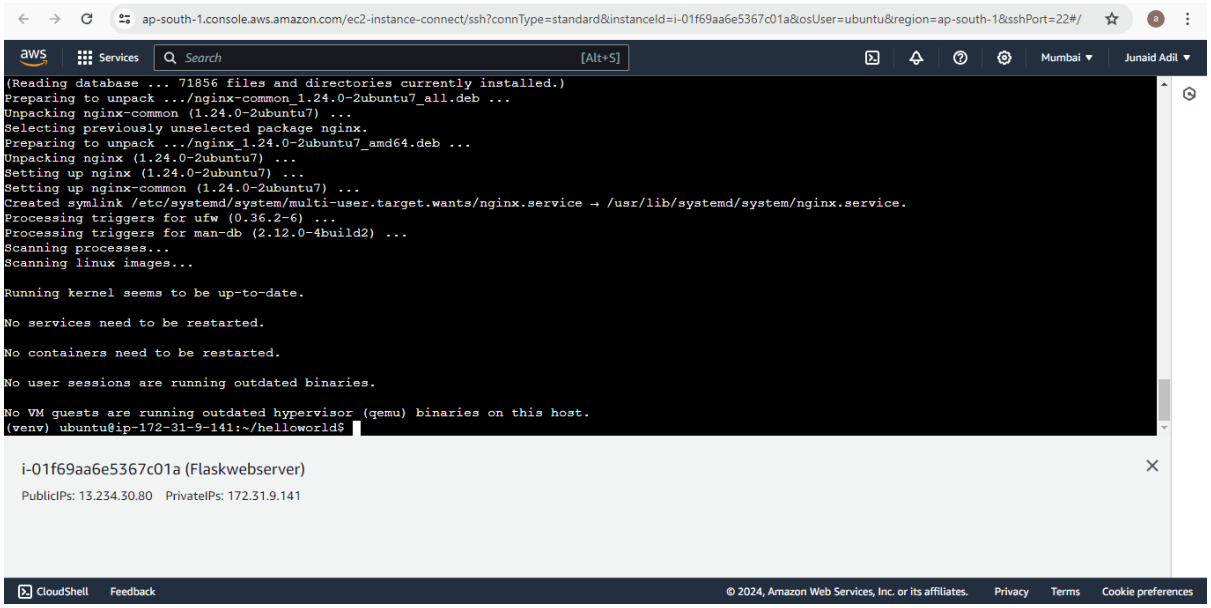
Step 16: To check the server is running, use command "**curl localhost:8000**"



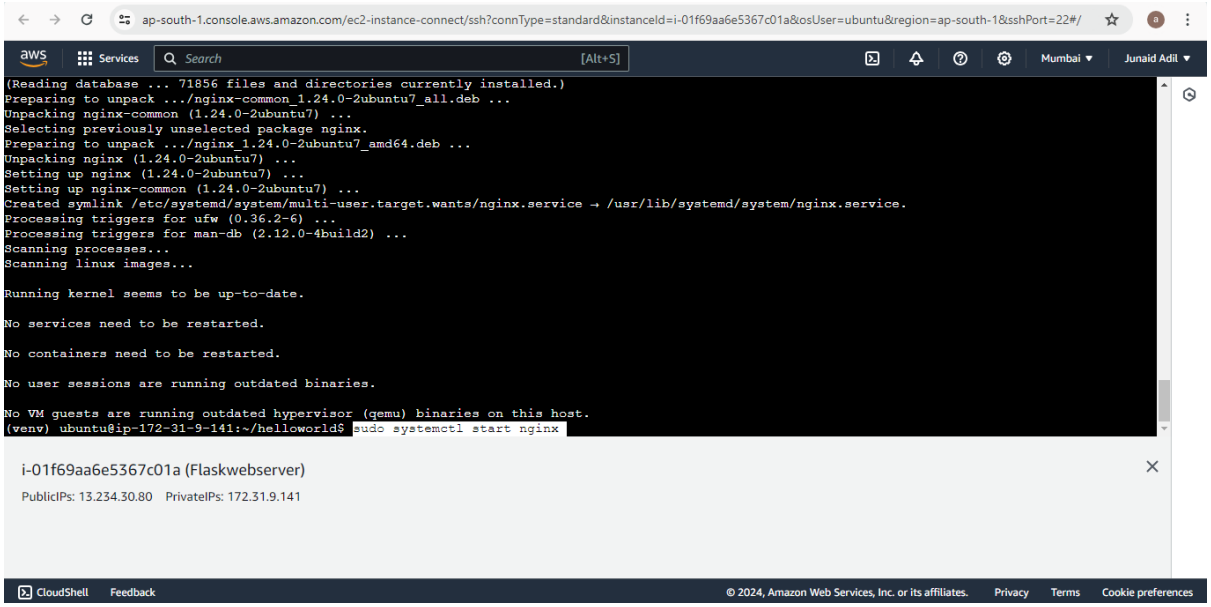


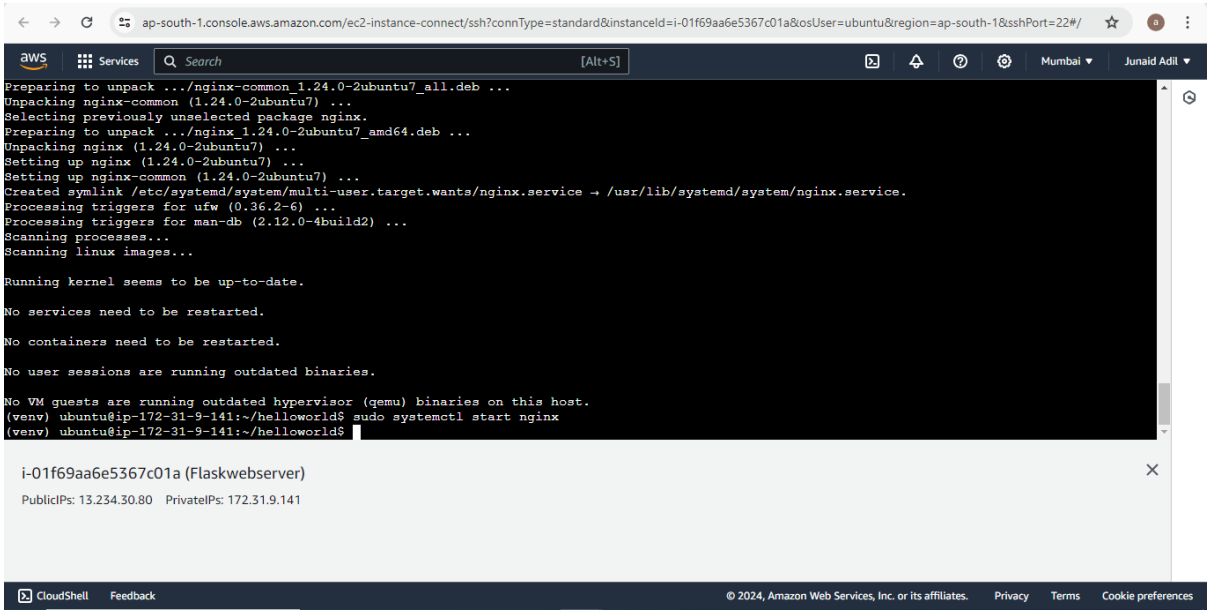
Step 17: Install the Nginx server using command "**sudo apt-get install nginx**"

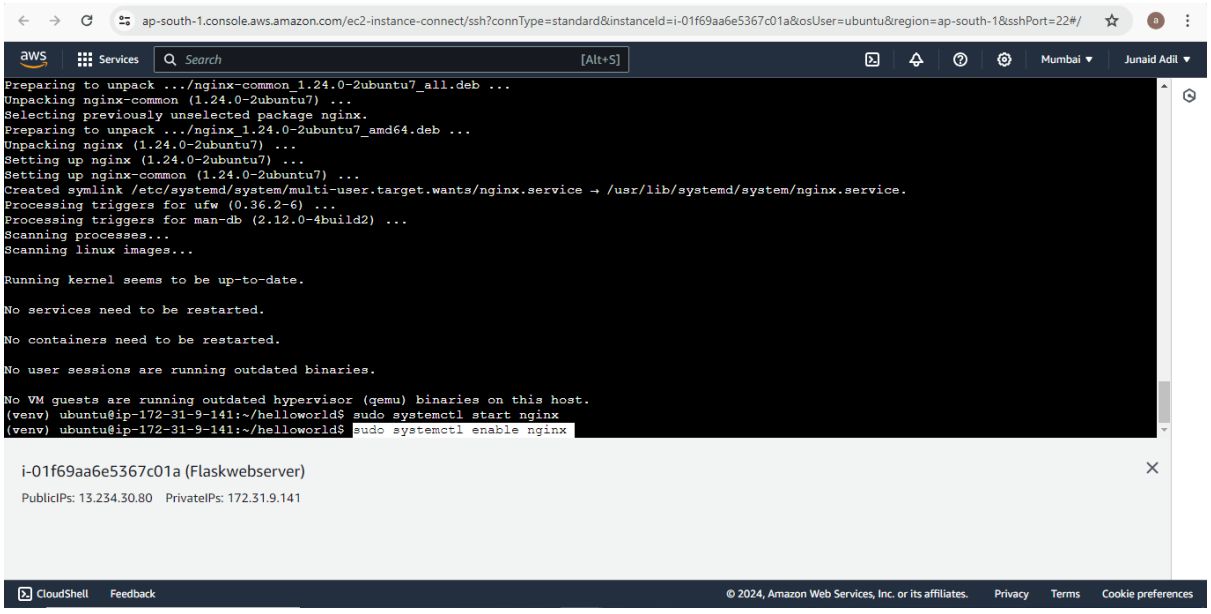


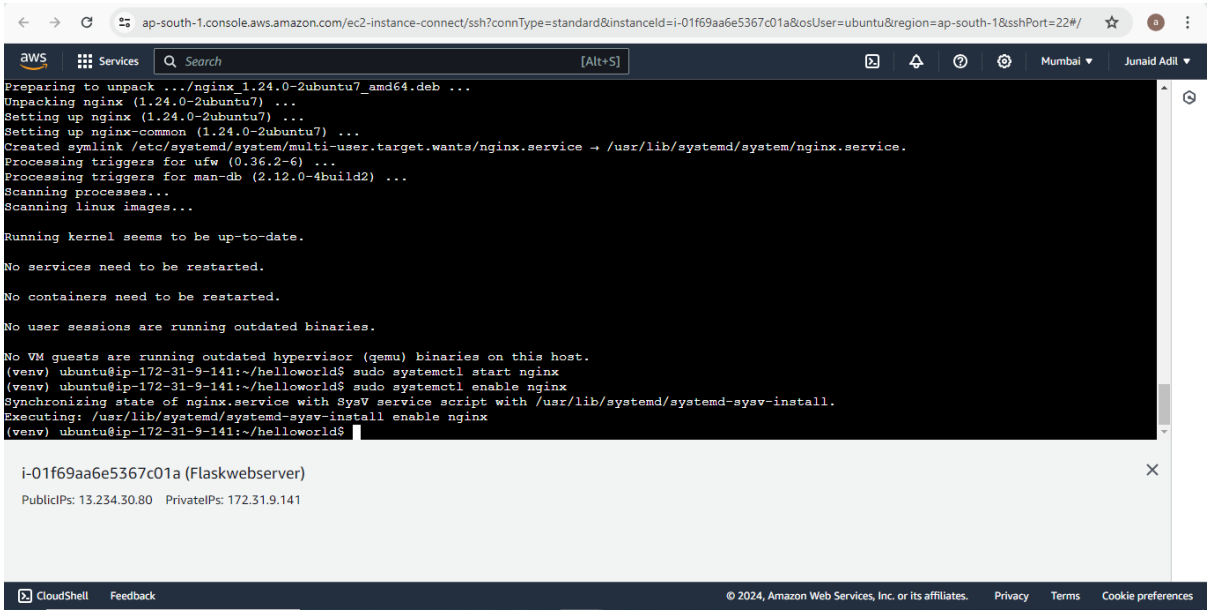


Step 18: To start and enable the Nginx server, run the commands "**sudo systemctl start nginx**" and to enable "**sudo systemctl enable nginx**"

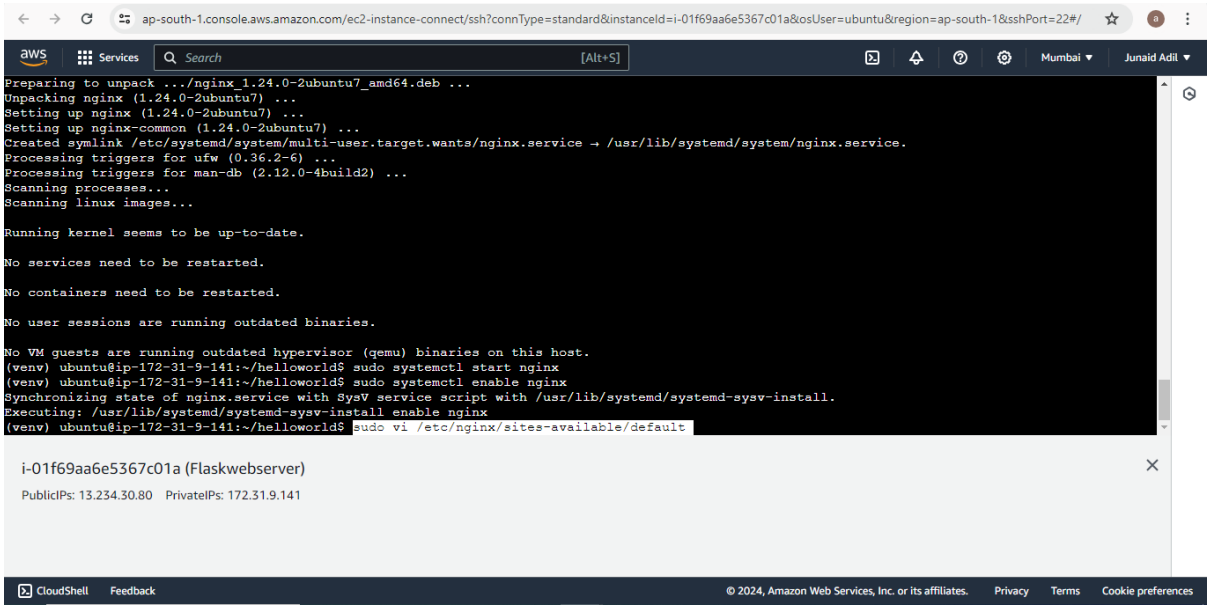


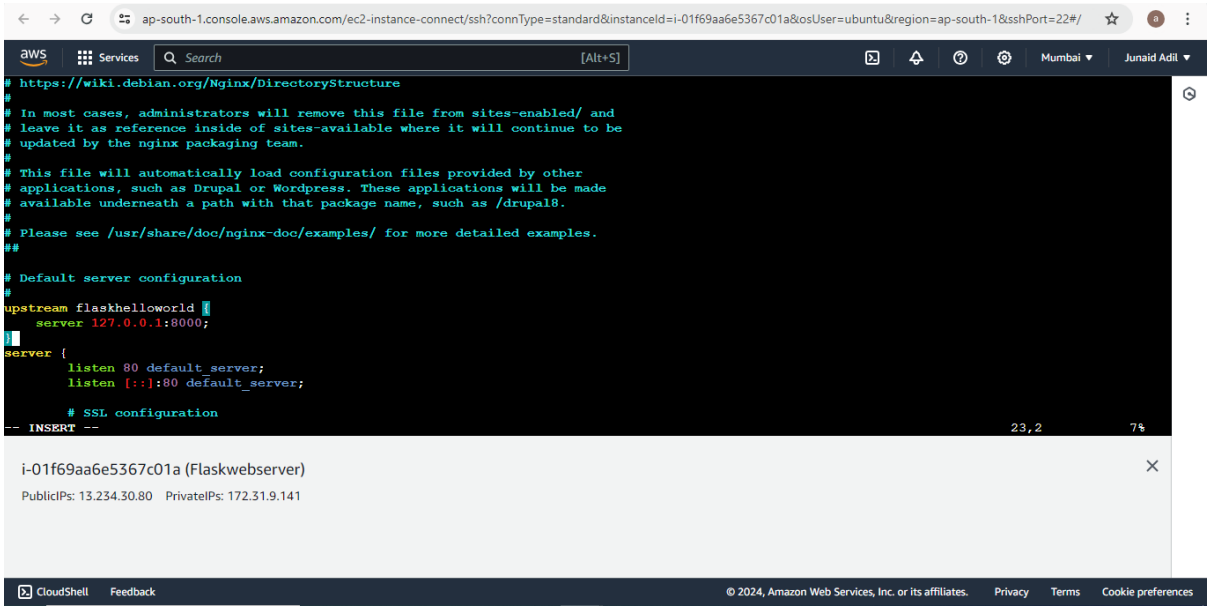

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
Services
Search
[Alt+S]
Mumbai
Junaid Adil
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7) ...
Setting up nginx (1.24.0-2ubuntu7) ...
Setting up nginx-common (1.24.0-2ubuntu7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ sudo systemctl start nginx
(venv) ubuntu@ip-172-31-9-141:~/helloworld$
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences

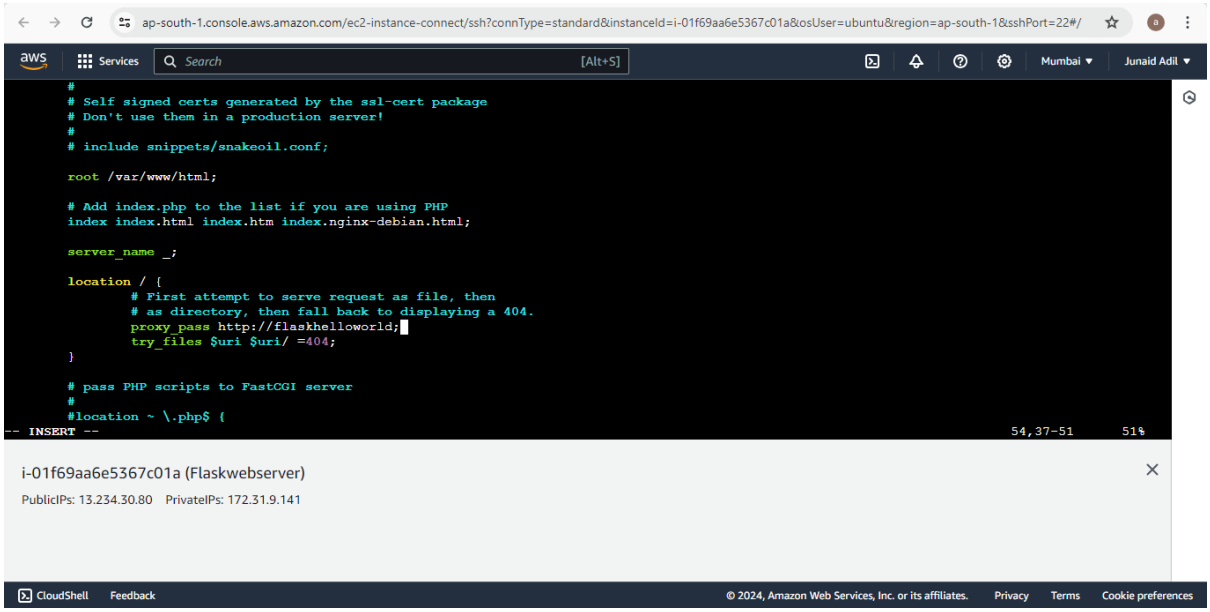

ap-south-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-01f69aa6e5367c01a&osUser=ubuntu&region=ap-south-1&sshPort=22#/
Services
Search
[Alt+S]
Mumbai
Junaid Adil
Preparing to unpack .../nginx-common_1.24.0-2ubuntu7_all.deb ...
Unpacking nginx-common (1.24.0-2ubuntu7) ...
Selecting previously unselected package nginx.
Preparing to unpack .../nginx_1.24.0-2ubuntu7_amd64.deb ...
Unpacking nginx (1.24.0-2ubuntu7) ...
Setting up nginx (1.24.0-2ubuntu7) ...
Setting up nginx-common (1.24.0-2ubuntu7) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nginx.service → /usr/lib/systemd/system/nginx.service.
Processing triggers for ufw (0.36.2-6) ...
Processing triggers for man-db (2.12.0-4build2) ...
Scanning processes...
Scanning linux images...
Running kernel seems to be up-to-date.
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.
No VM guests are running outdated hypervisor (qemu) binaries on this host.
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ sudo systemctl start nginx
(venv) ubuntu@ip-172-31-9-141:~/helloworld$ sudo systemctl enable nginx
i-01f69aa6e5367c01a (Flaskwebserver)
PublicIPs: 13.234.30.80 PrivateIPs: 172.31.9.141
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences



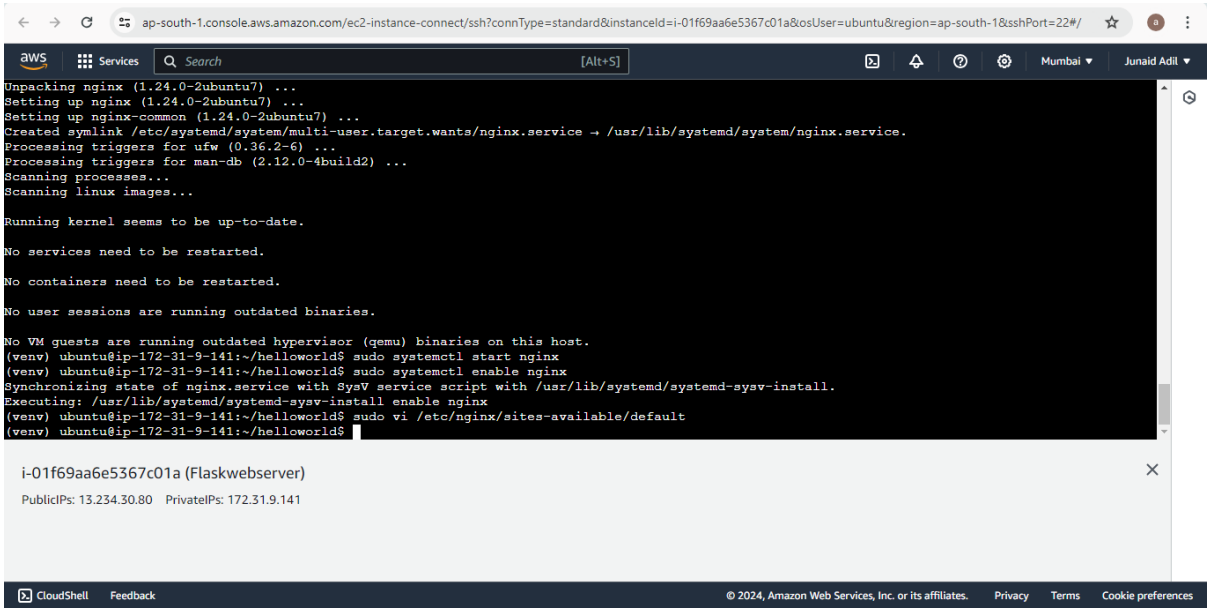
Step 19: To make some changes in the Nginx Server we need to edit the default file using the command "**sudo vi /etc/nginx/sites-available/default**" and add the code at the top of the file and add a proxy pass to flaskhelloworld at location below



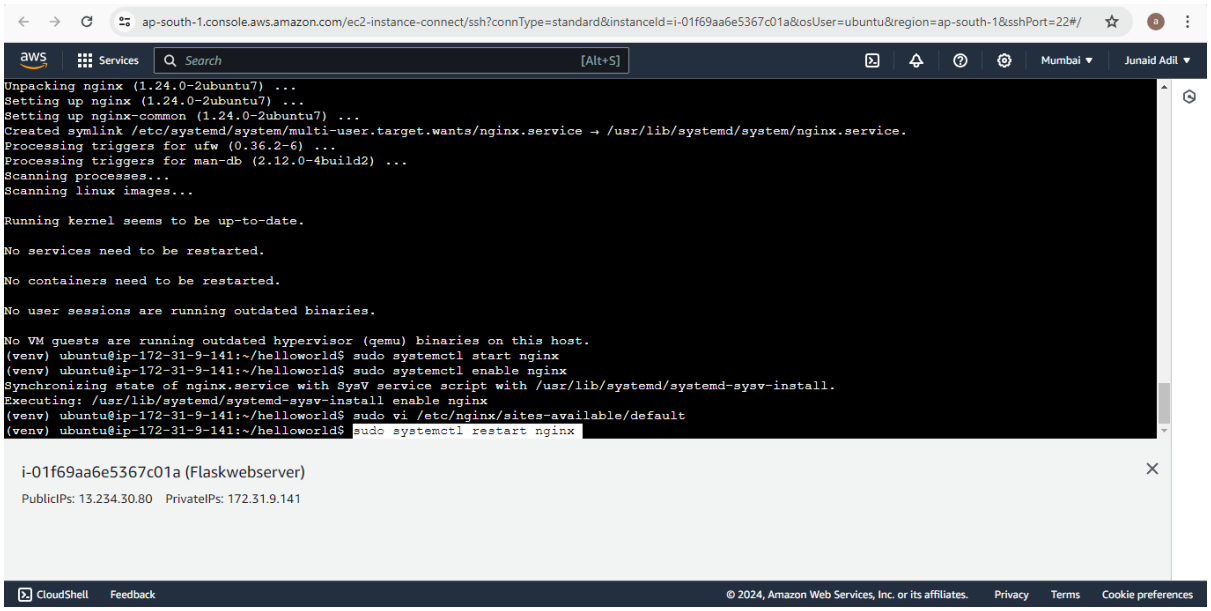




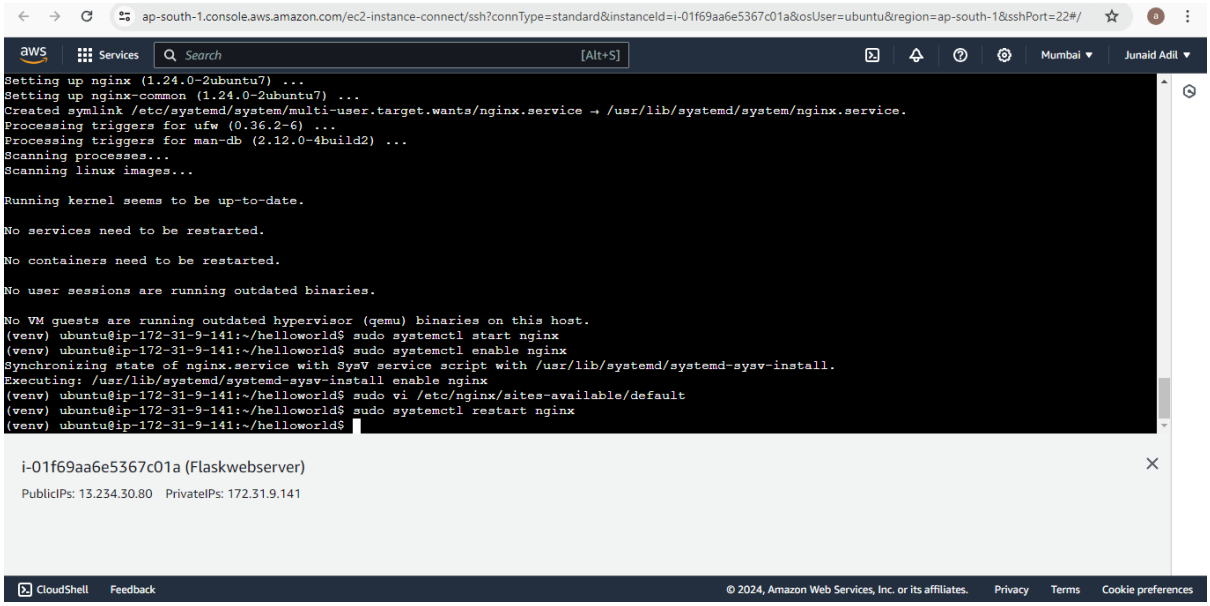
Then save the file :wq!



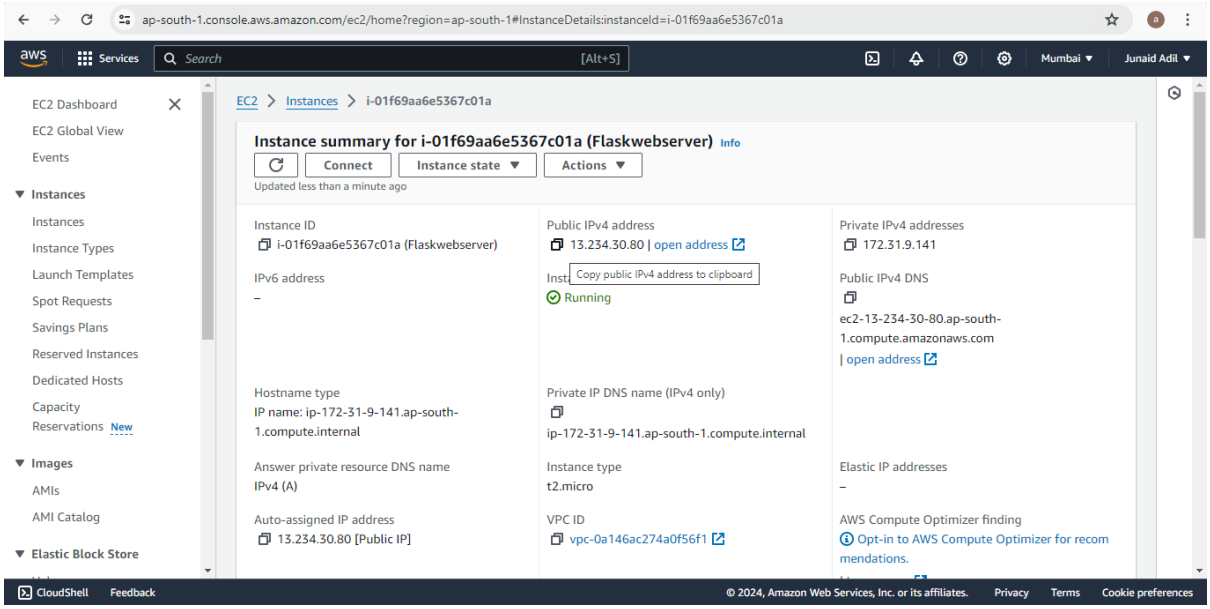
Step 20: To restart the Nginx server run the command "**sudo systemctl restart nginx**"



We can see we didn't get any error, so the server is working fine.



Step 21: To check go to the instance and copy the Public IP and paste it in the browser and press enter




13.234.30.80
Not secure
13.234.30.80
Hello World!