

Deploy a Sample Python Application Using Flask Module

Case Study: **Deploy a Sample Python Application Using Flask Module**

Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991. It is used for:

- web development (server-side),
- software development,
- mathematics,
- system scripting.

Syntax:

Python syntax can be executed by writing directly in the Command Line:

```
Print('hello world')
```

Flask Module:

It is the latest and comprehensive guide designed for beginners and professionals to learn **Python Web Framework Flaskone**. It is one of the most popular Python-based web frameworks.

Flask, a Python web application framework, was created by Armin Ronacher. Known for its lightweight and efficient nature, Flask is designed for quick starts and accommodates complex applications. It is based on the Werkzeug WSGI toolkit and Jinja2 template engine.

In this application we have to use –

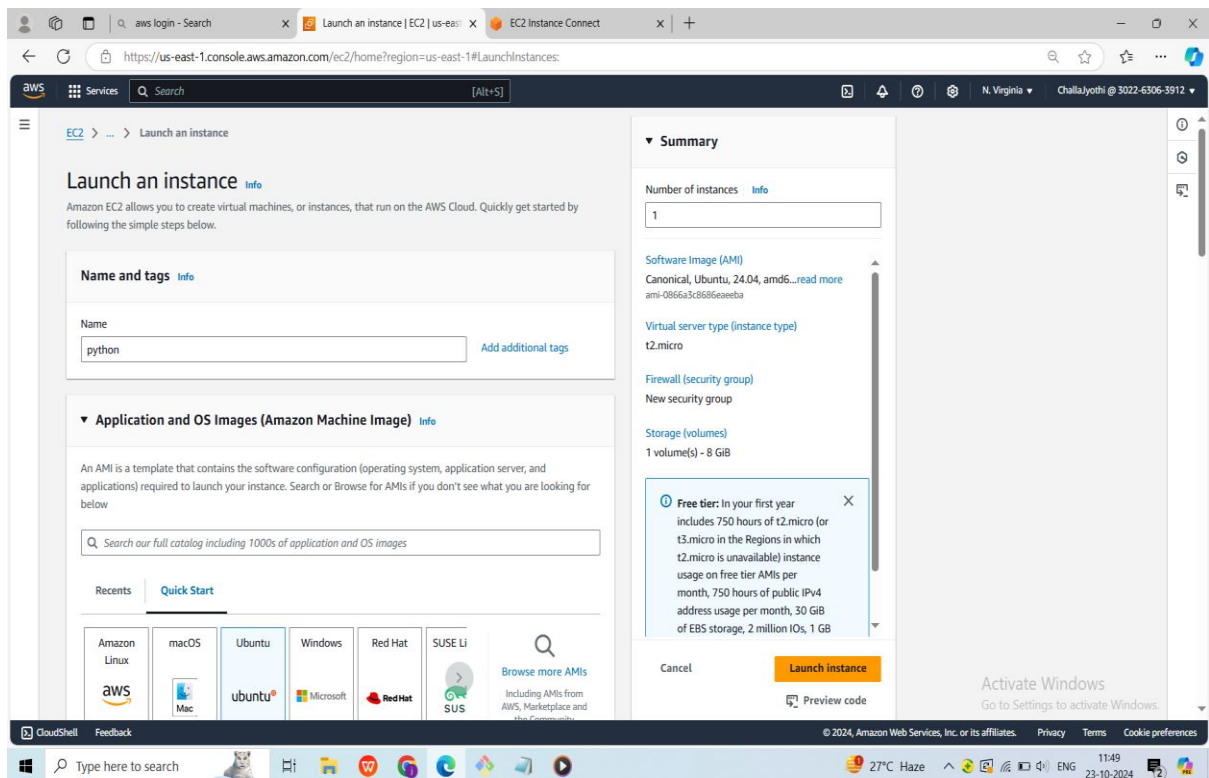
Launch an ec2 instance

Install python3 and pip

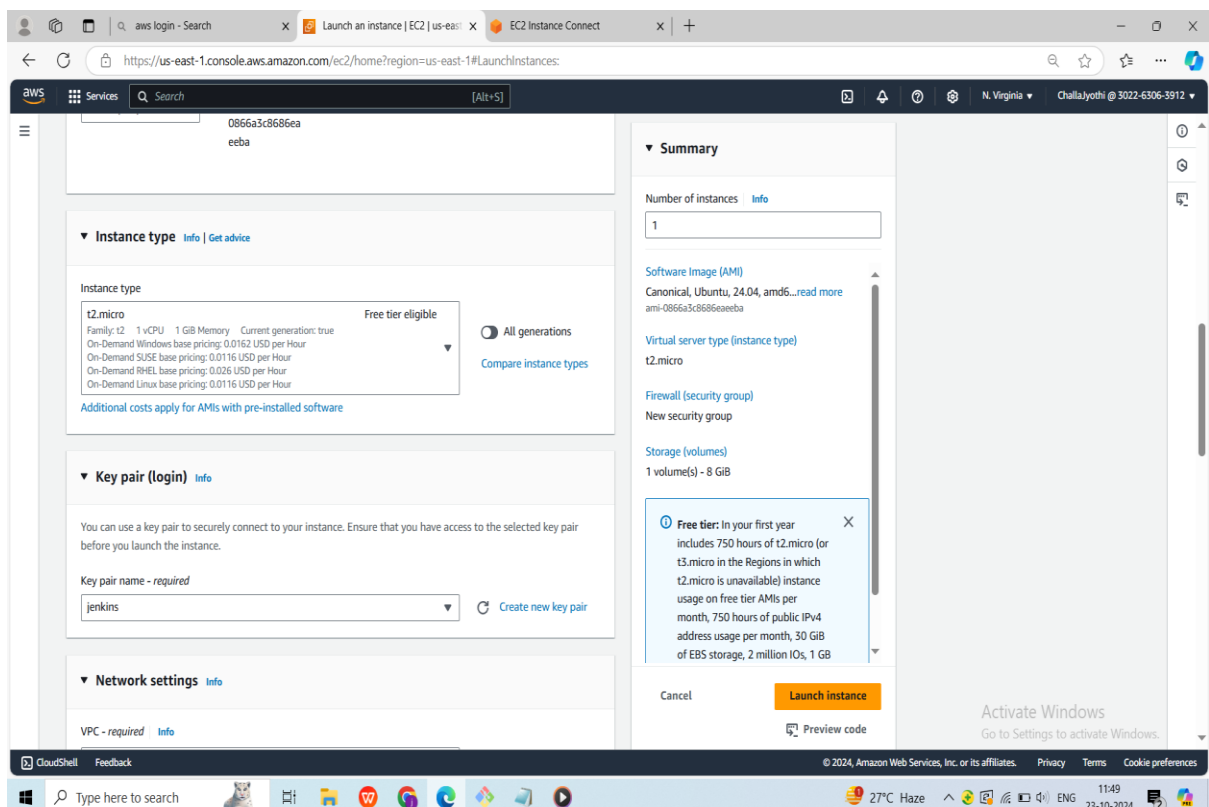
Install flask module

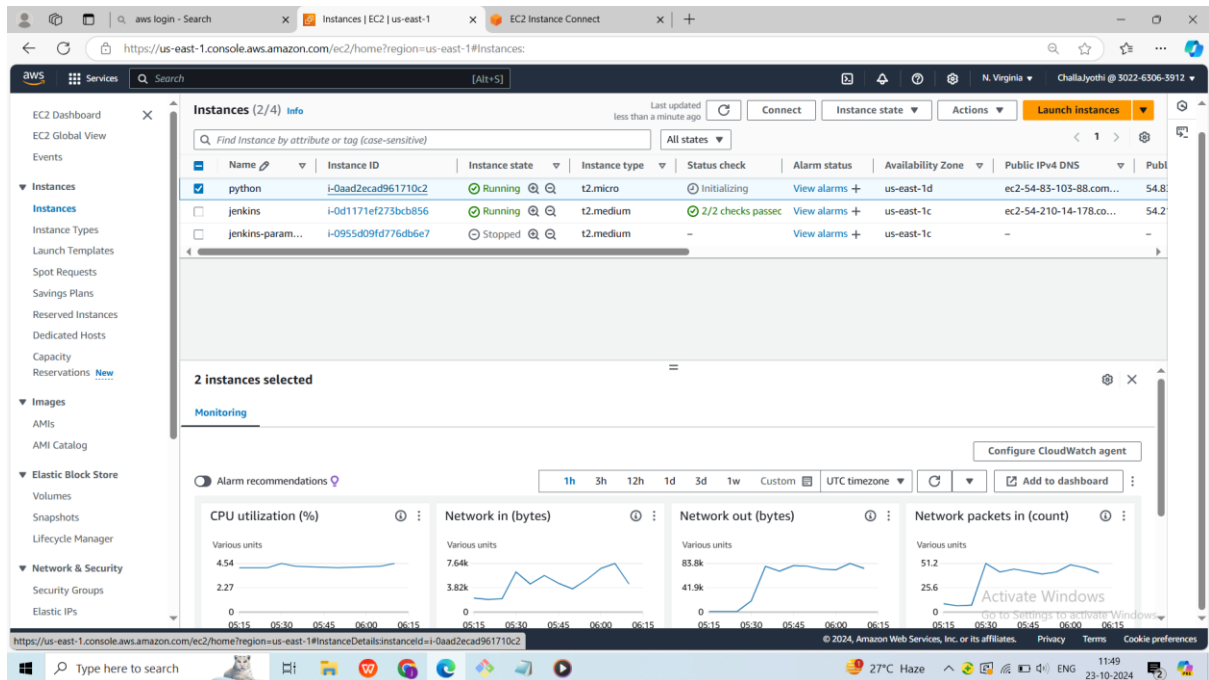
In app.py we have to import the code .

We have to Launch an ec2 instance in Ubuntu or Linux server.

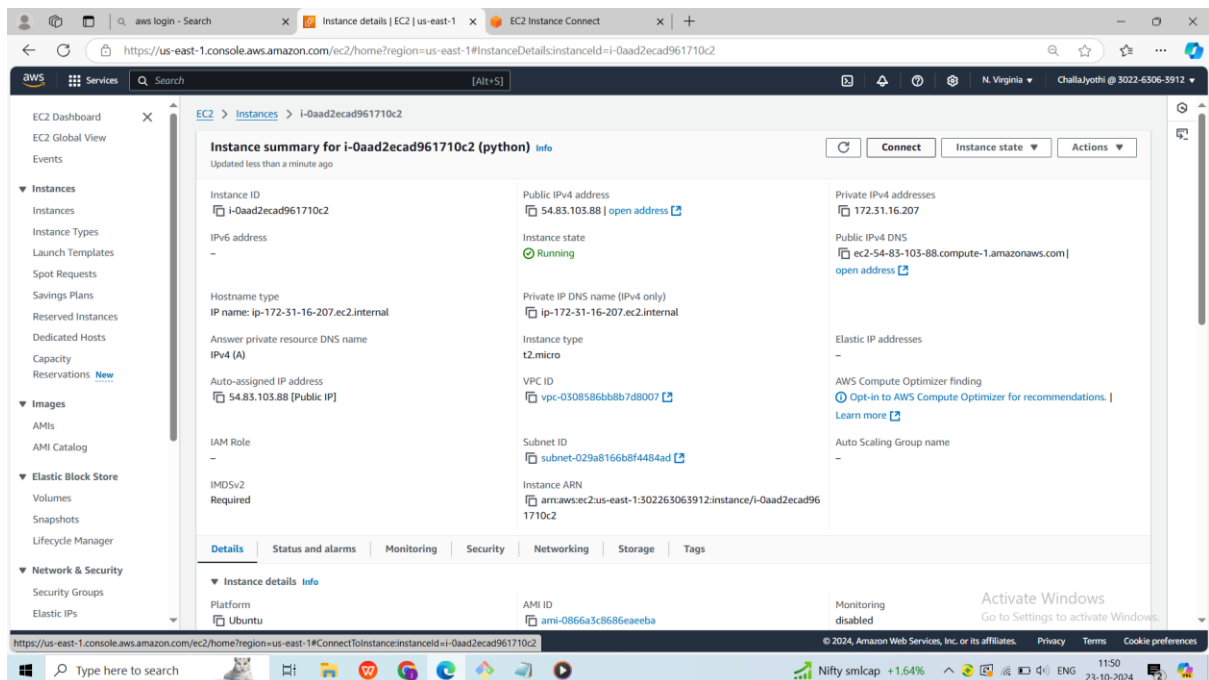


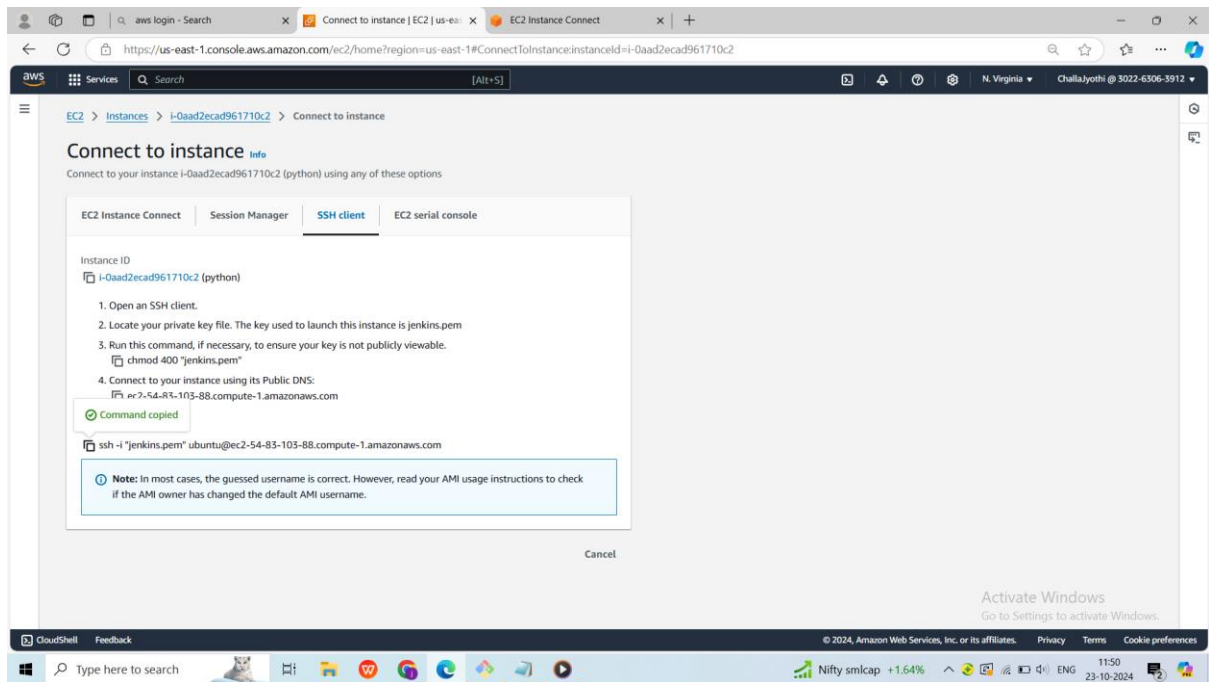
We have to create key pair and edit the security port as 5001



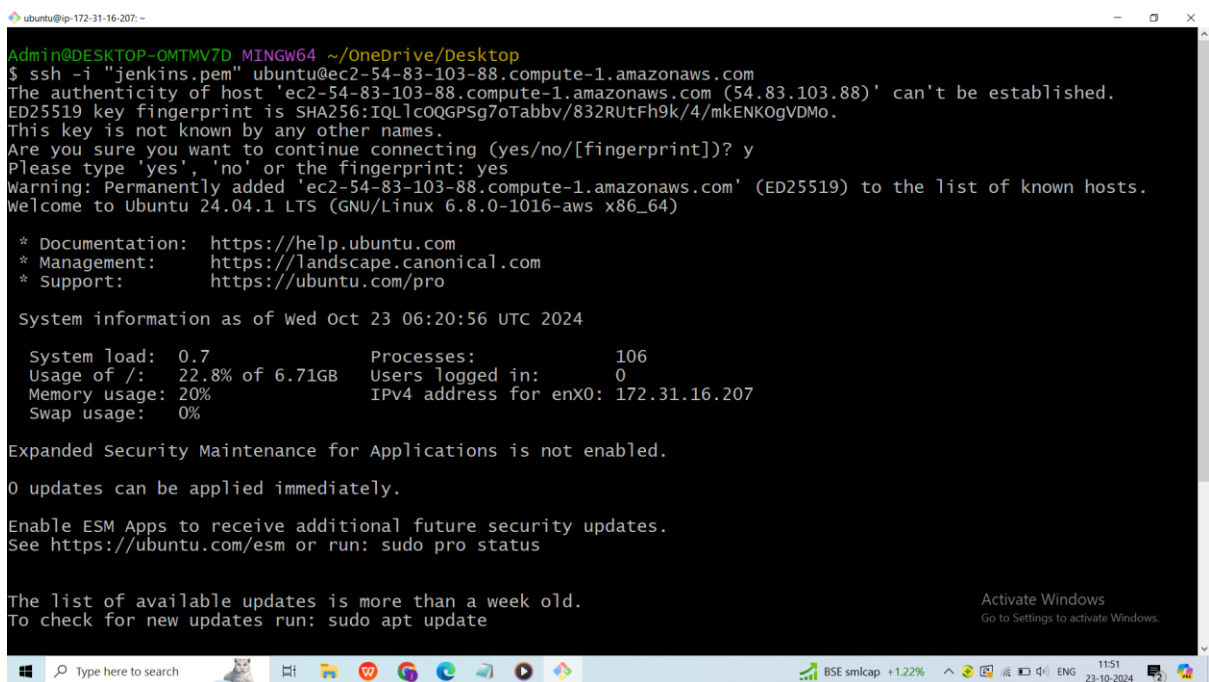


Once, the instance is launched we have to connect to server

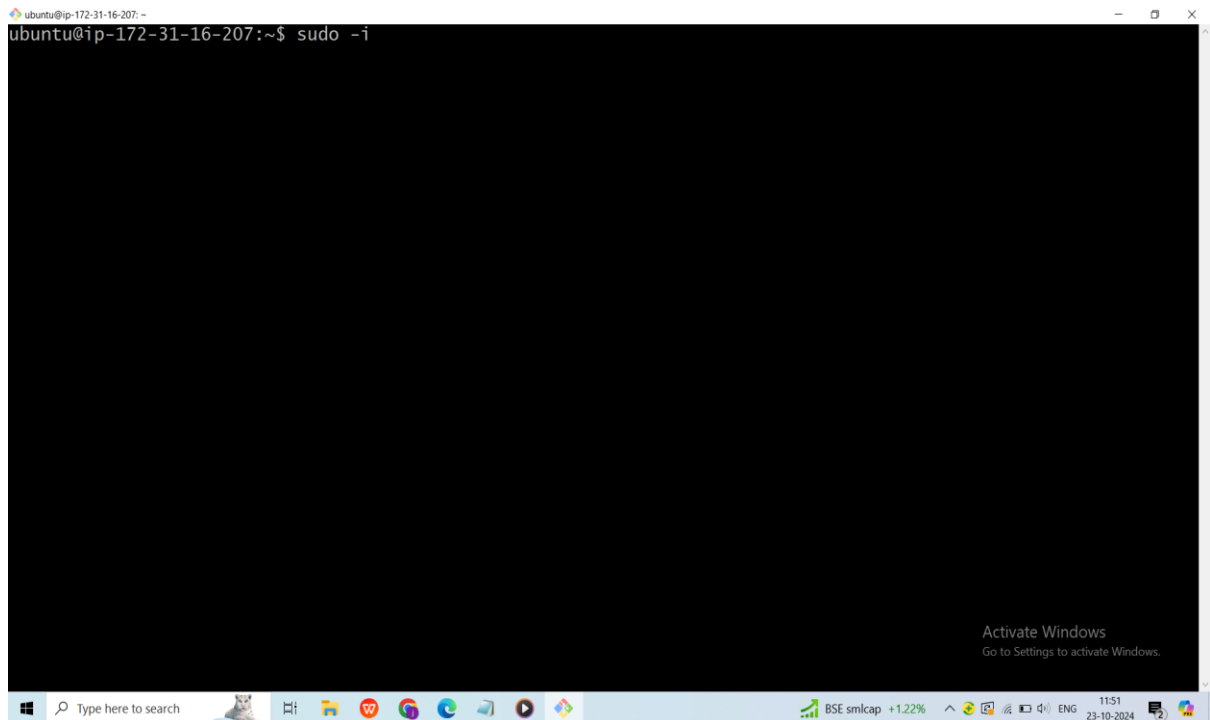




Connecting to the server



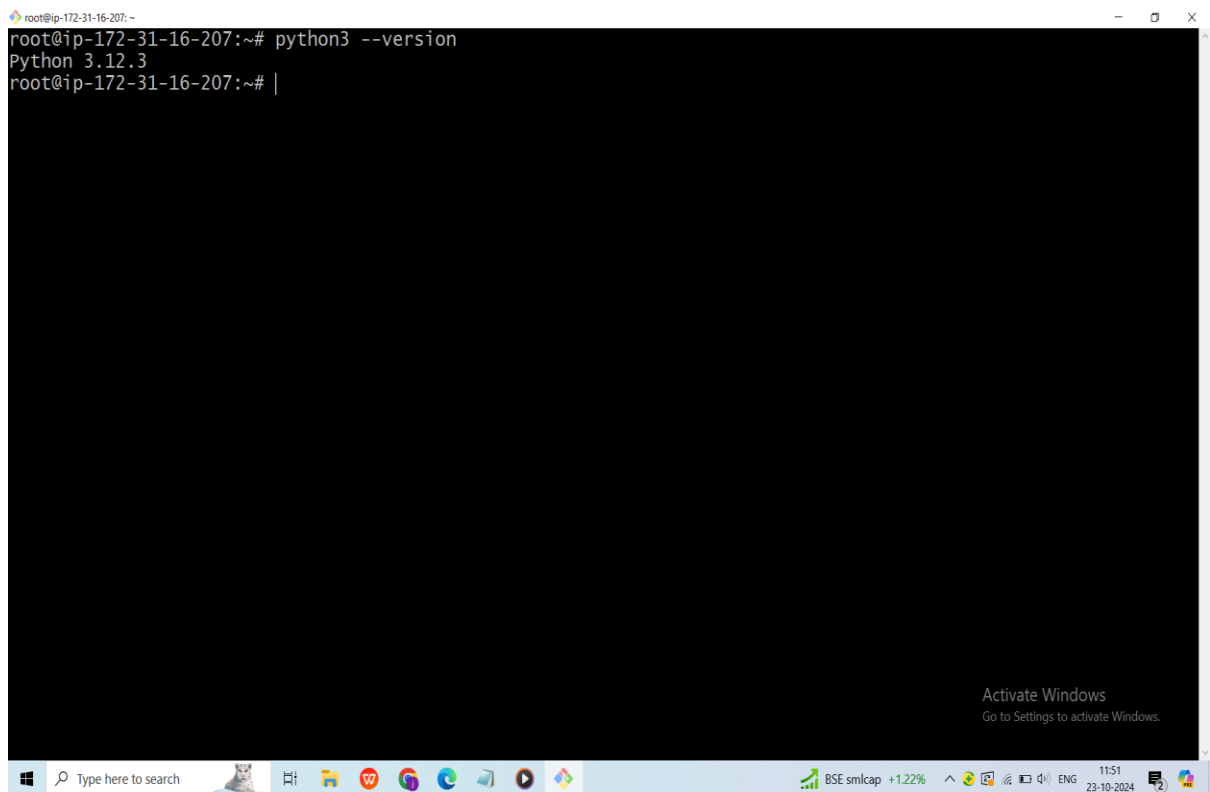
To switch normal user to root user we use **sudo -i**



```
ubuntu@ip-172-31-16-207: ~  
ubuntu@ip-172-31-16-207:~$ sudo -i
```

The terminal window shows the command `sudo -i` being executed. The prompt changes from `ubuntu@ip-172-31-16-207:~$` to `root@ip-172-31-16-207:~#`. The window title is `ubuntu@ip-172-31-16-207: ~`. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right shows the BSE smicap stock price at +1.22%, system icons, and the date/time: 11:51 on 23-10-2024.

We have to install python3 version(**apt install python3**) ,if it already exist we have to check version (**python3 --version**)



```
root@ip-172-31-16-207: ~  
root@ip-172-31-16-207:~# python3 --version  
Python 3.12.3  
root@ip-172-31-16-207:~# |
```

The terminal window shows the command `python3 --version` being executed. The output is `Python 3.12.3`. The prompt changes from `root@ip-172-31-16-207:~#` to `root@ip-172-31-16-207:~# |`. The window title is `root@ip-172-31-16-207: ~`. The Windows taskbar at the bottom shows the search bar, task view, and several application icons. The system tray on the right shows the BSE smicap stock price at +1.22%, system icons, and the date/time: 11:51 on 23-10-2024.

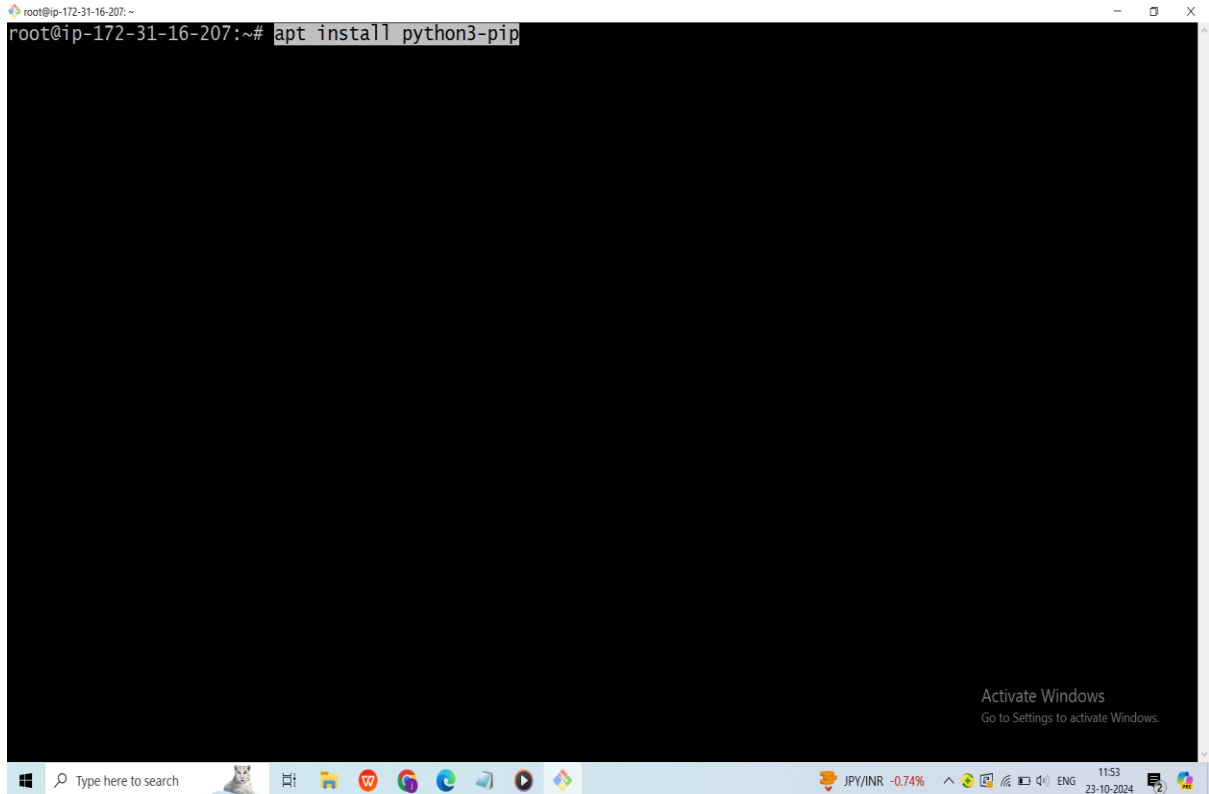
We have to update the server (**apt update -y**)

```
root@ip-172-31-16-207: ~  
root@ip-172-31-16-207:~# python3 --version  
Python 3.12.3  
root@ip-172-31-16-207:~# apt update -y
```



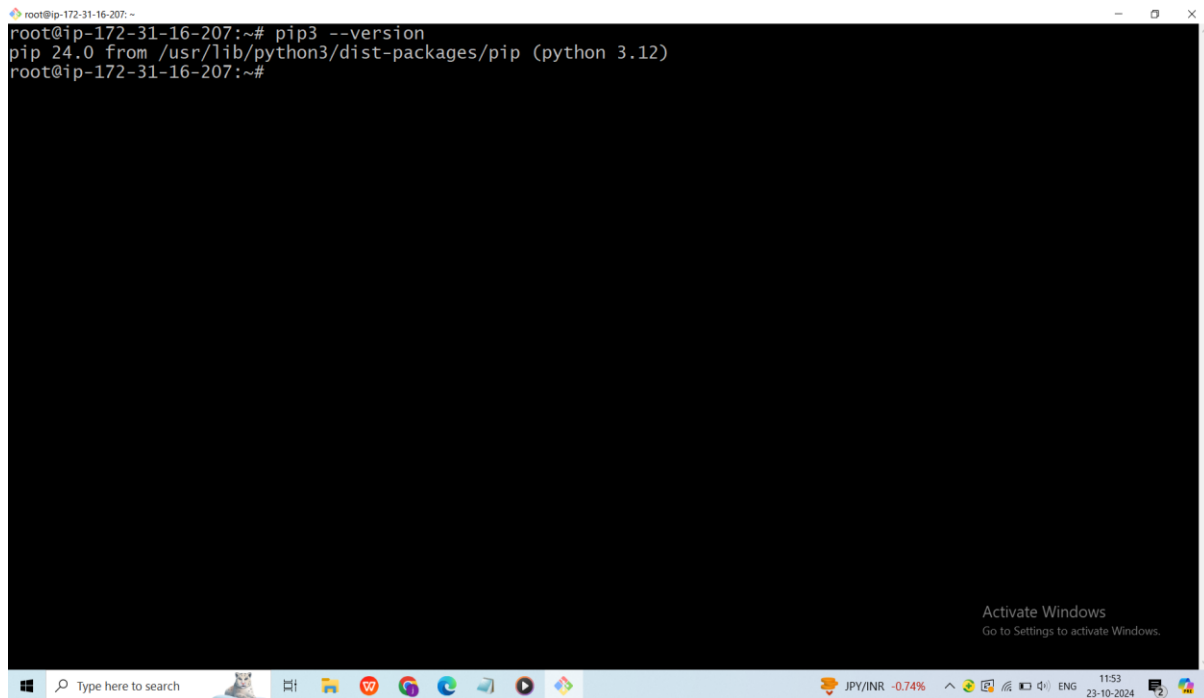
After updating the server we have to install the pip (**apt install python3-pip**)

```
root@ip-172-31-16-207: ~  
root@ip-172-31-16-207:~# apt install python3-pip
```



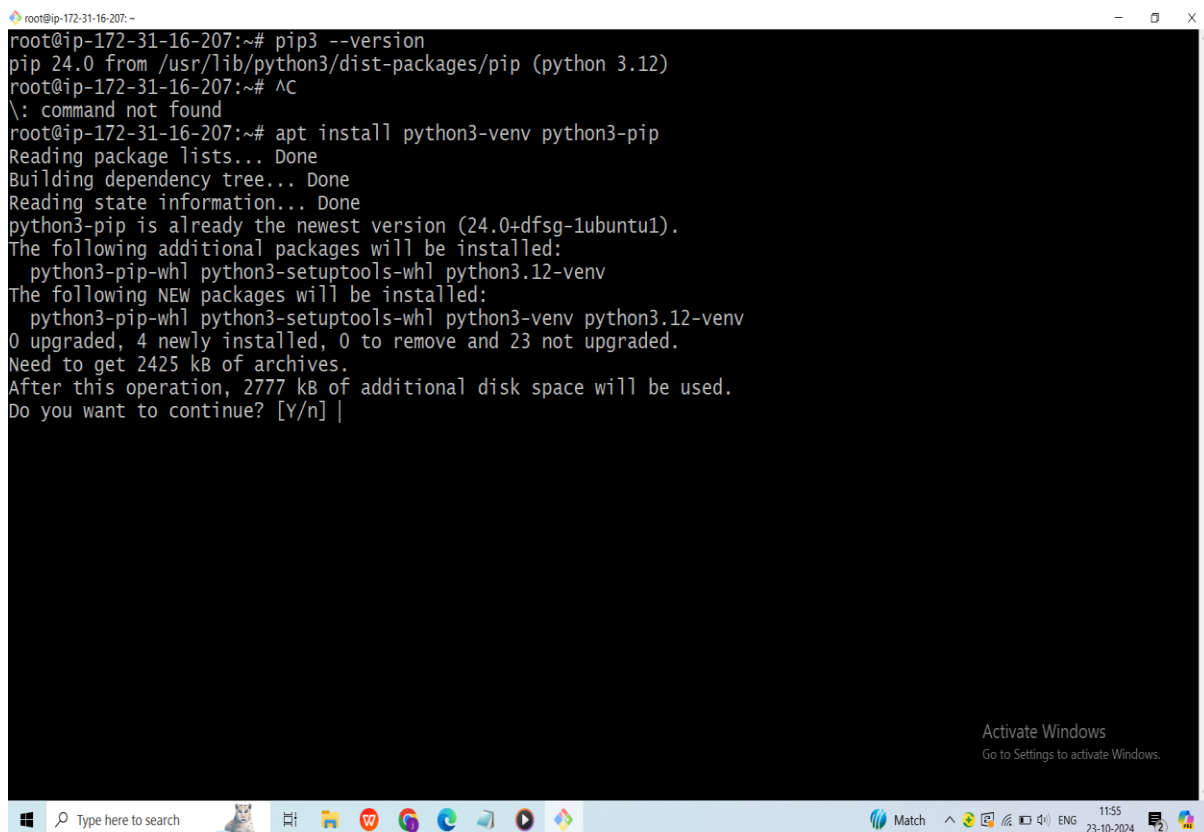
After installing the pip we have to check the version of pip (**pip3 --version**)

```
root@ip-172-31-16-207: ~  
root@ip-172-31-16-207:~# pip3 --version  
pip 24.0 from /usr/lib/python3/dist-packages/pip (python 3.12)  
root@ip-172-31-16-207:~#
```



Let us install python3-venv by using the command (**apt install python3-venv python3-pip**)

```
root@ip-172-31-16-207: ~  
root@ip-172-31-16-207:~# pip3 --version  
pip 24.0 from /usr/lib/python3/dist-packages/pip (python 3.12)  
root@ip-172-31-16-207:~# ^C  
\: command not found  
root@ip-172-31-16-207:~# apt install python3-venv python3-pip  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3-pip is already the newest version (24.0+dfsg-1ubuntu1).  
The following additional packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3.12-venv  
The following NEW packages will be installed:  
  python3-pip-whl python3-setuptools-whl python3-venv python3.12-venv  
0 upgraded, 4 newly installed, 0 to remove and 23 not upgraded.  
Need to get 2425 kB of archives.  
After this operation, 2777 kB of additional disk space will be used.  
Do you want to continue? [Y/n] |
```



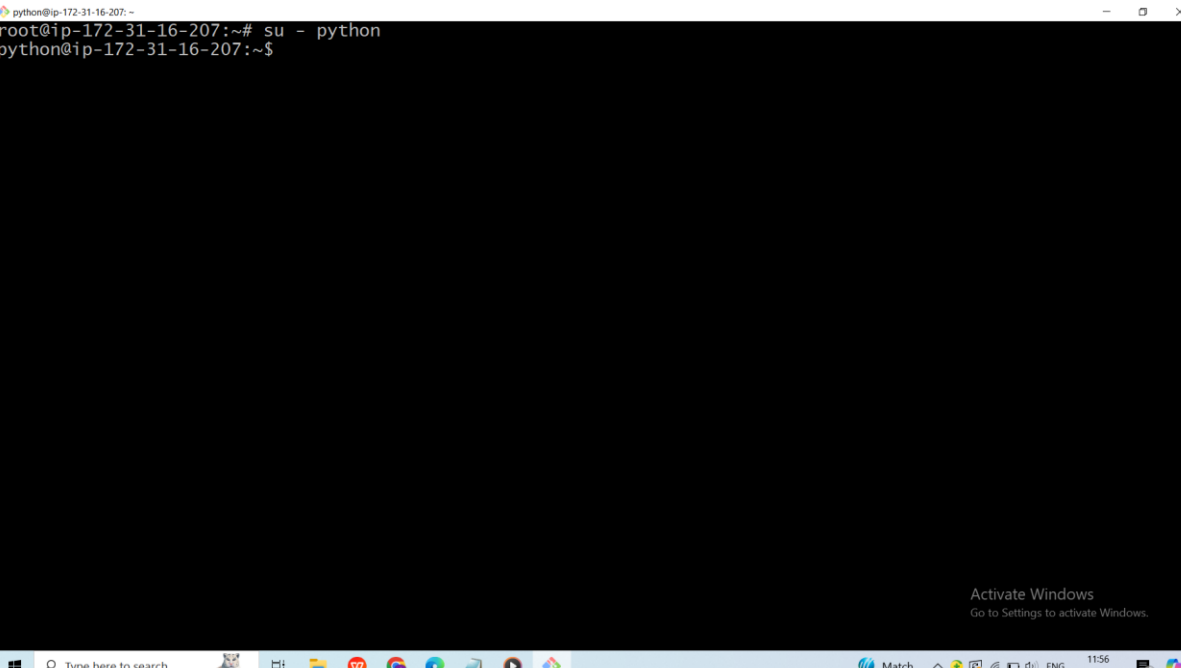
You can install Flask globally with the command `pip3 install flask`, but it's recommended to create a virtual environment and install the Flask application there. Let's create a new user and install the Flask application in a new virtual environment: (**adduser python**)

The screenshot shows a Windows terminal window with a black background and white text. The command prompt shows the user is root at ip-172-31-16-207. The command executed is `adduser python`. The output shows the following steps:

- info: Adding user 'python' ...
- info: Selecting UID/GID from range 1000 to 59999 ...
- info: Adding new group 'python' (1001) ...
- info: Adding new user 'python' (1001) with group 'python (1001)' ...
- info: Creating home directory '/home/python' ...
- info: Copying files from '/etc/skel' ...

The prompt then asks for a new password, with a cursor visible after the text "New password:". The Windows taskbar is visible at the bottom, showing the Start button, a search bar, and several application icons. The system tray on the right shows the date and time as 11:55 on 23-10-2024.

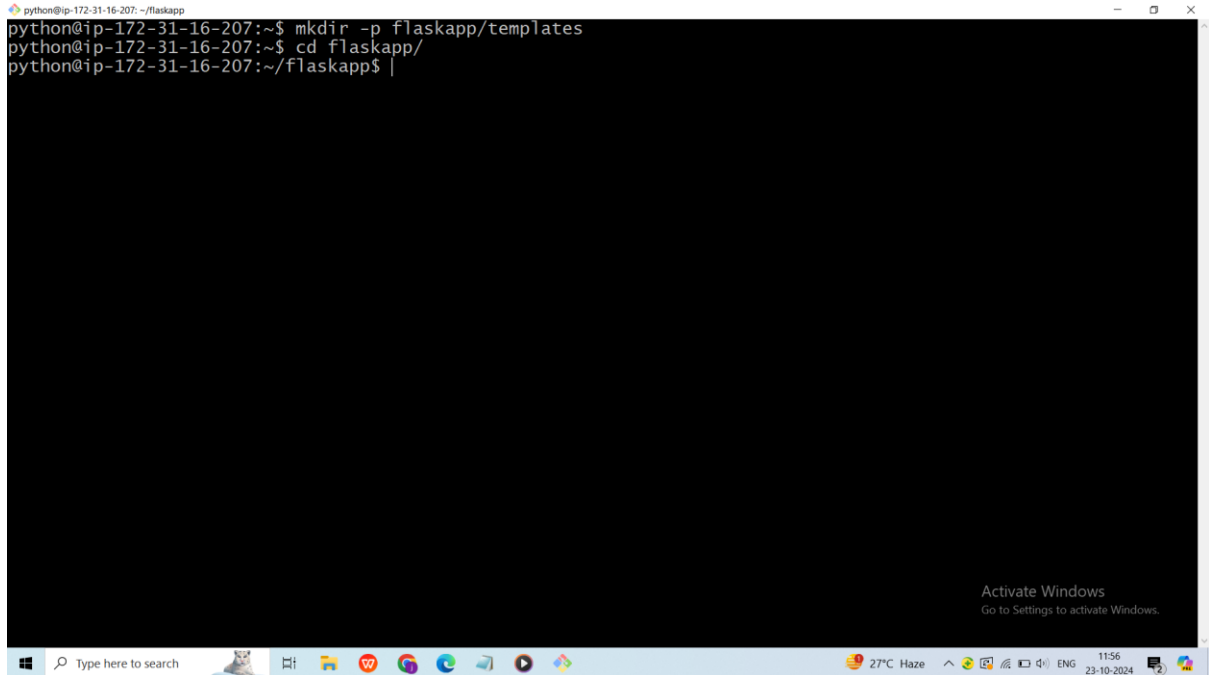
Now, you can log in as the user john with the above command. (**su – Python**)

The image shows a Windows 10 desktop environment. A terminal window is open, displaying the command prompt. The prompt shows the user is root on a machine with IP 172.31.16.207, having switched from python to root using 'su - python'. The terminal output shows the root prompt and the user's name 'python'. The taskbar at the bottom contains the Start button, a search bar, and several application icons including File Explorer, Word, Chrome, and a game. The system tray on the right shows the date and time as 11:56 on 23-10-2024, along with network and volume icons. A watermark 'Activate Windows' is visible in the bottom right corner of the desktop area.

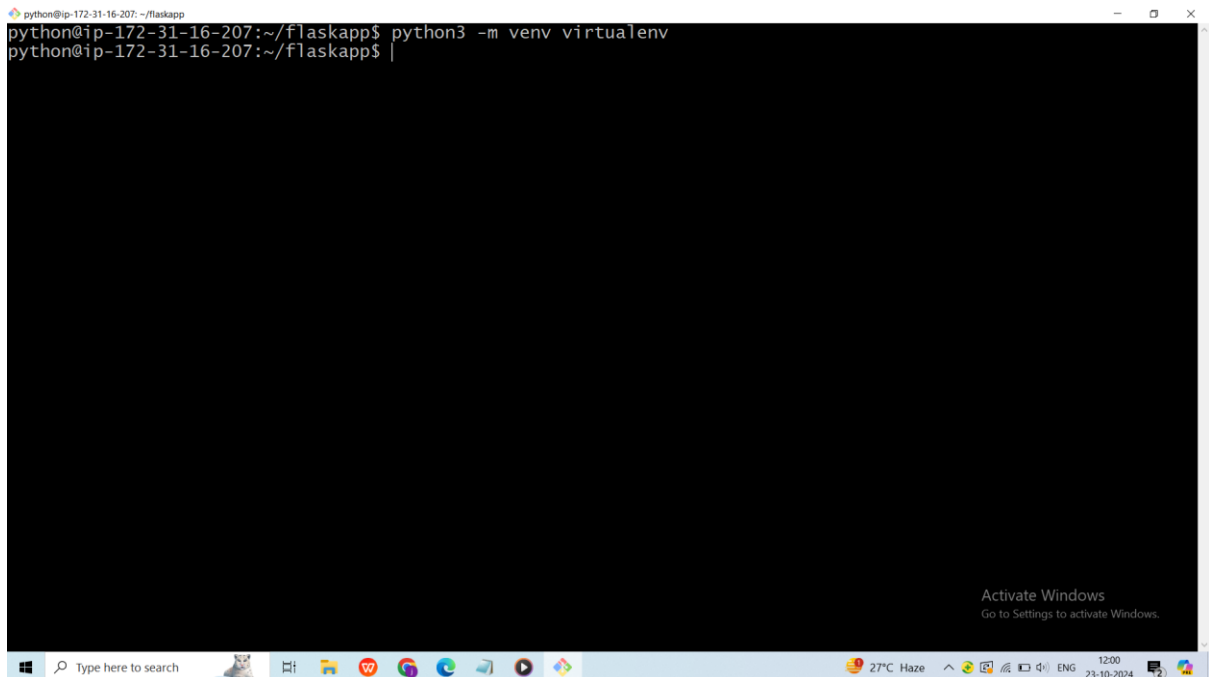
After logging in, create the directories flaskapp/templates (**mkdir -p flaskapp/templates**)

Enter the directory called **flaskapp** (**cd flaskapp**)– you can now go ahead and create the virtual environment. (**python3 -m venv virtualenv**)

```
python@ip-172-31-16-207: ~/flaskapp
python@ip-172-31-16-207:~$ mkdir -p flaskapp/templates
python@ip-172-31-16-207:~$ cd flaskapp/
python@ip-172-31-16-207:~/flaskapp$ |
```

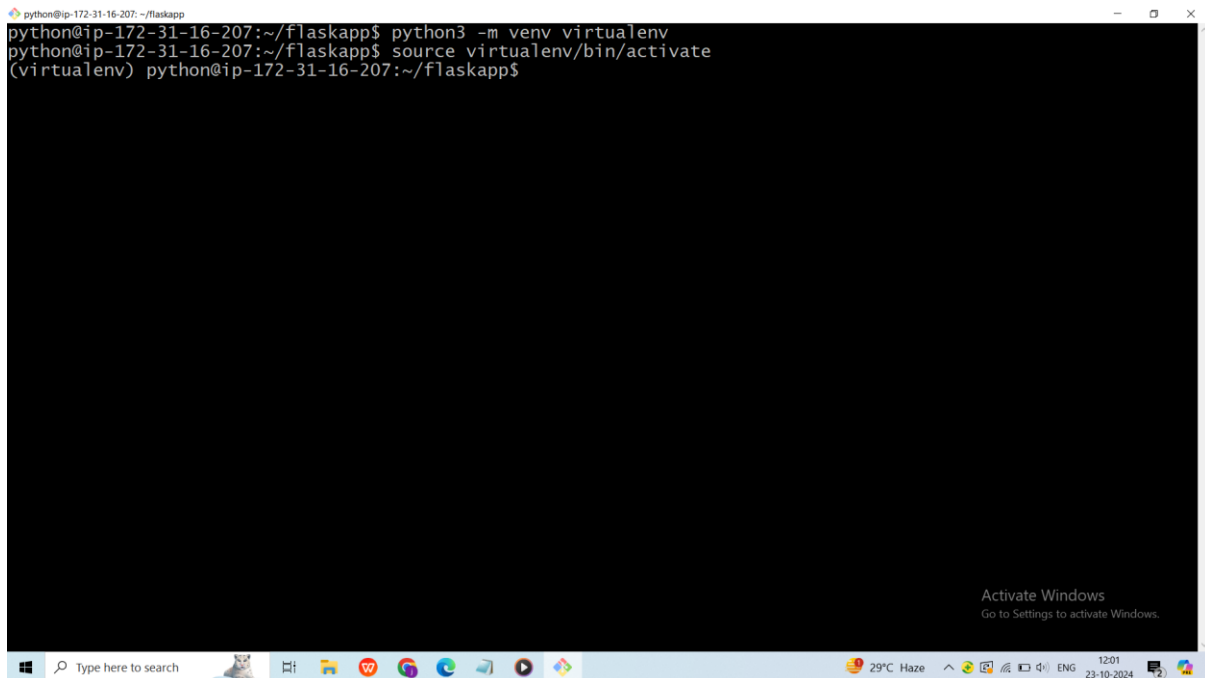
A terminal window with a black background and white text. The prompt is 'python@ip-172-31-16-207: ~/flaskapp'. The first command is 'python@ip-172-31-16-207:~\$ mkdir -p flaskapp/templates'. The second command is 'python@ip-172-31-16-207:~\$ cd flaskapp/'. The third line shows the prompt 'python@ip-172-31-16-207:~/flaskapp\$' followed by a vertical bar. The window has a Windows taskbar at the bottom with various icons and a system tray showing '27°C Haze' and '11:56 23-10-2024'. An 'Activate Windows' watermark is visible in the bottom right corner of the terminal area.

```
python@ip-172-31-16-207: ~/flaskapp
python@ip-172-31-16-207:~/flaskapp$ python3 -m venv virtualenv
python@ip-172-31-16-207:~/flaskapp$ |
```

A terminal window with a black background and white text. The prompt is 'python@ip-172-31-16-207: ~/flaskapp'. The command is 'python@ip-172-31-16-207:~/flaskapp\$ python3 -m venv virtualenv'. The next line shows the prompt 'python@ip-172-31-16-207:~/flaskapp\$' followed by a vertical bar. The window has a Windows taskbar at the bottom with various icons and a system tray showing '27°C Haze' and '12:00 23-10-2024'. An 'Activate Windows' watermark is visible in the bottom right corner of the terminal area.

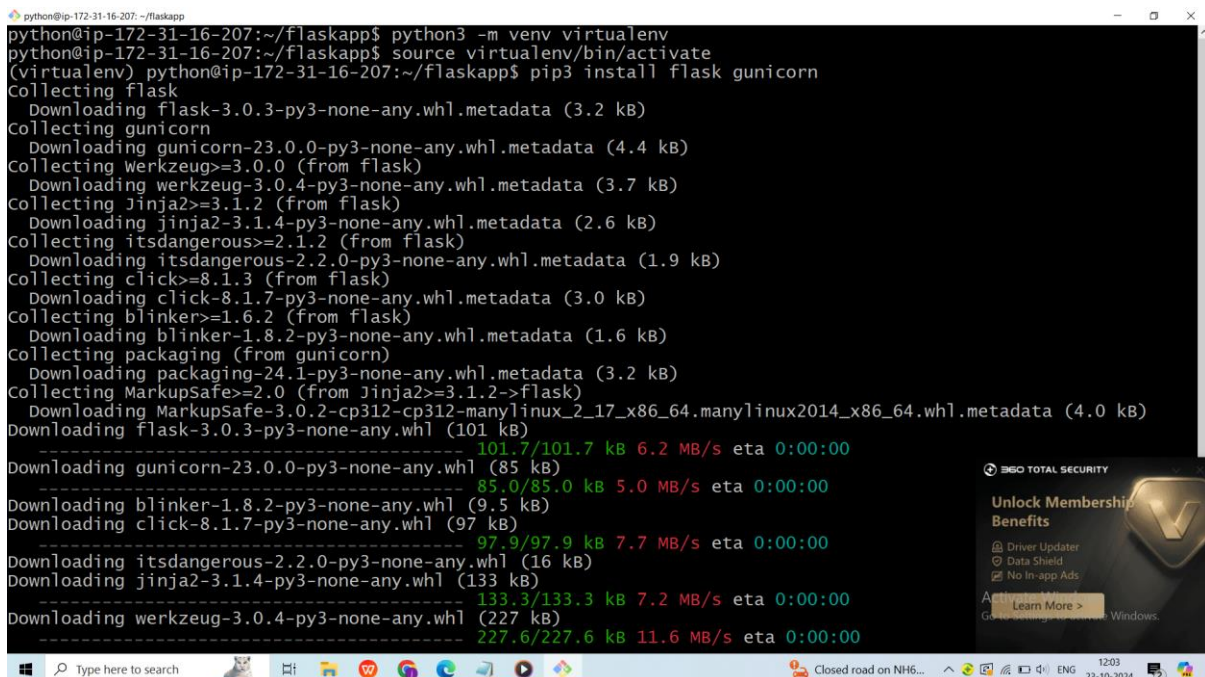
Now enter the virtual environment with: **source virtualenv/bin/activate**

```
python@ip-172-31-16-207: ~/flaskapp
python@ip-172-31-16-207:~/flaskapp$ python3 -m venv virtualenv
python@ip-172-31-16-207:~/flaskapp$ source virtualenv/bin/activate
(virtualenv) python@ip-172-31-16-207:~/flaskapp$
```



You have now activated the virtual environment, where we can start our installation. Let's install Flask and Gunicorn: (**pip3 install flask gunicorn**)

```
python@ip-172-31-16-207: ~/flaskapp
python@ip-172-31-16-207:~/flaskapp$ python3 -m venv virtualenv
python@ip-172-31-16-207:~/flaskapp$ source virtualenv/bin/activate
(virtualenv) python@ip-172-31-16-207:~/flaskapp$ pip3 install flask gunicorn
Collecting flask
  Downloading flask-3.0.3-py3-none-any.whl.metadata (3.2 kB)
Collecting gunicorn
  Downloading gunicorn-23.0.0-py3-none-any.whl.metadata (4.4 kB)
Collecting Werkzeug>=3.0.0 (from flask)
  Downloading werkzeug-3.0.4-py3-none-any.whl.metadata (3.7 kB)
Collecting Jinja2>=3.1.2 (from flask)
  Downloading jinja2-3.1.4-py3-none-any.whl.metadata (2.6 kB)
Collecting itsdangerous>=2.1.2 (from flask)
  Downloading itsdangerous-2.2.0-py3-none-any.whl.metadata (1.9 kB)
Collecting click>=8.1.3 (from flask)
  Downloading click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Collecting blinker>=1.6.2 (from flask)
  Downloading blinker-1.8.2-py3-none-any.whl.metadata (1.6 kB)
Collecting packaging (from gunicorn)
  Downloading packaging-24.1-py3-none-any.whl.metadata (3.2 kB)
Collecting MarkupSafe>=2.0 (from Jinja2>=3.1.2->flask)
  Downloading MarkupSafe-3.0.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.0 kB)
Downloading flask-3.0.3-py3-none-any.whl (101 kB)
----- 101.7/101.7 kB 6.2 MB/s eta 0:00:00
Downloading gunicorn-23.0.0-py3-none-any.whl (85 kB)
----- 85.0/85.0 kB 5.0 MB/s eta 0:00:00
Downloading blinker-1.8.2-py3-none-any.whl (9.5 kB)
----- 9.5/9.5 kB 7.7 MB/s eta 0:00:00
Downloading click-8.1.7-py3-none-any.whl (97 kB)
----- 97.9/97.9 kB 7.7 MB/s eta 0:00:00
Downloading itsdangerous-2.2.0-py3-none-any.whl (16 kB)
----- 16.0/16.0 kB 7.2 MB/s eta 0:00:00
Downloading jinja2-3.1.4-py3-none-any.whl (133 kB)
----- 133.3/133.3 kB 7.2 MB/s eta 0:00:00
Downloading werkzeug-3.0.4-py3-none-any.whl (227 kB)
----- 227.6/227.6 kB 11.6 MB/s eta 0:00:00
```



Once Flask is installed, you can run a simple application to test if everything is working as expected. Make sure you are logged in as the user “john”. Create an **app.py** file using your preferred text editor: **vi app.py**

```
python@ip-172-31-16-207: ~/flaskapp
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ pip install flask
Requirement already satisfied: Flask in ./virtualenv/lib/python3.12/site-packages (3.0.3)
Requirement already satisfied: Werkzeug<=3.0.0 in ./virtualenv/lib/python3.12/site-packages (from flask) (3.0.4)
Requirement already satisfied: Jinja2<=3.1.2 in ./virtualenv/lib/python3.12/site-packages (from flask) (3.1.4)
Requirement already satisfied: itsdangerous<=2.1.2 in ./virtualenv/lib/python3.12/site-packages (from flask) (2.2.0)
Requirement already satisfied: click>=8.1.3 in ./virtualenv/lib/python3.12/site-packages (from flask) (8.1.7)
Requirement already satisfied: blinker>=1.6.2 in ./virtualenv/lib/python3.12/site-packages (from flask) (1.8.2)
Requirement already satisfied: MarkupSafe<=2.0 in ./virtualenv/lib/python3.12/site-packages (from Jinja2<=3.1.2->flask) (3.0.2)
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ vi app.py
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
Address already in use
Port 5000 is in use by another program. Either identify and stop that program, or start the server with a different port.
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ tnlp
tnlp: command not found
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ vi app.py
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ python3 app.py
* Serving Flask app 'app'
* Debug mode: off
Address already in use
Port 5000 is in use by another program. Either identify and stop that program, or start the server with a different port.
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ cat app
cat: app: No such file or directory
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ cat app.py
from flask import Flask
app = Flask(__name__)
@app.route("/")
def helloworld():
    return "Hello World!"
if __name__ == "__main__":
    app.run()

(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ vi app.py
(Virtualenv) python@ip-172-31-16-207:~/flaskapp$ python3 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 all addresses (0.0.0.0)
* Running on http://127.0.0.1:5001
* Running on http://172.31.16.207:5001
Press CTRL+C to quit
124.123.178.85 - - [23/Oct/2024 07:48:29] "GET / HTTP/1.1" 200 -
124.123.178.85 - - [23/Oct/2024 07:48:36] "GET /favicon.ico HTTP/1.1" 404 -

Activate Windows
Go to Settings to activate Windows.
```

Add the following code to the file:

```
python@ip-172-31-16-207: ~/flaskapp
from flask import Flask
app = Flask(__name__)
@app.route("/")
def helloworld():
    return "Hello World!"
if __name__ == "__main__":
    app.run(host='0.0.0.0',port=5001)

"app.py" 9L, 177B

Activate Windows
Go to Settings to activate Windows.
```

To run the app.py we use **python3 app.py**

Once the code is successful we have to copy the public Ip address and paste in any browser

The final output is:

