

## **PROJECT-3**

### **DEPLOYING PYTHON APPLICATION USING VARIOUS METHODS**

#### **WHAT IS PYTHON?**

It is a computer programming language often used to build websites software automate tasks and conduct data analysis.

Python is a versatile and high-level computer programming language often used for:

- Building websites and software.
- Automating tasks.
- Conducting data analysis.

It is beginner-friendly and supports multiple programming paradigms, including object-oriented, procedural, and functional programming.

#### **WHAT IS PIP?**

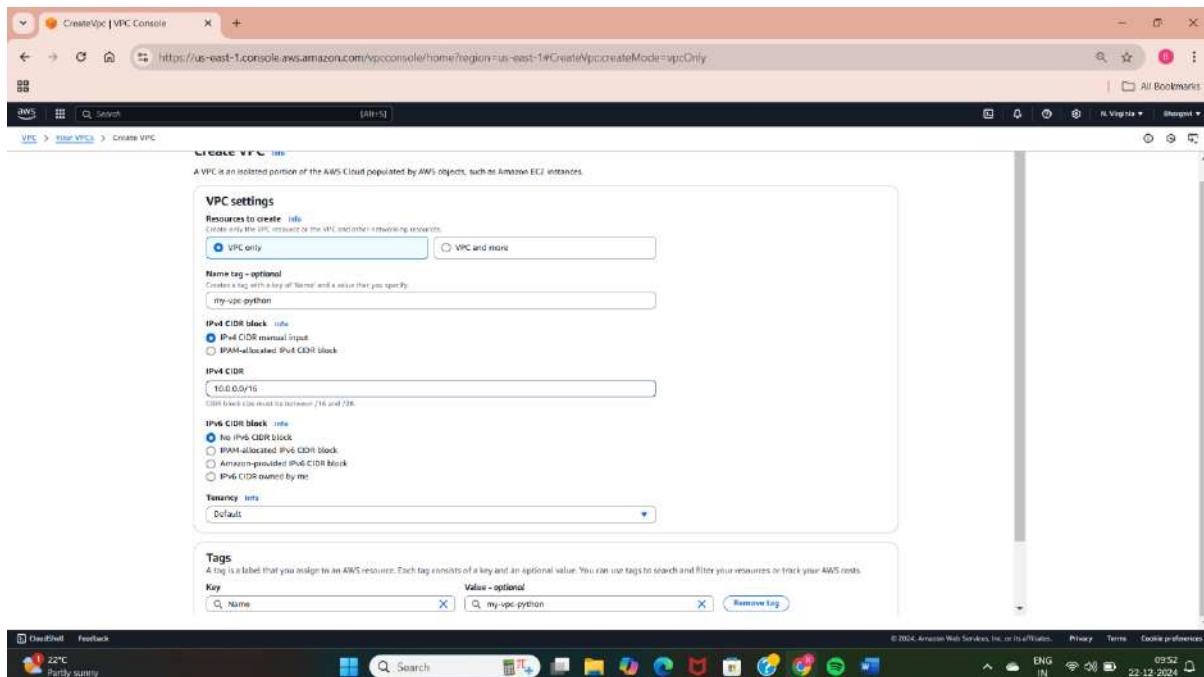
- Pip stands for **Pip Installs Packages**.
- It is the default package manager for Python, enabling the installation of Python libraries and modules from the Python Package Index (PyPI).
- Pip is a package manager python packages or modules
- NOTE: if you have python version3.4 or later PIP is including by default

#### **PRE-REQUISITES:**

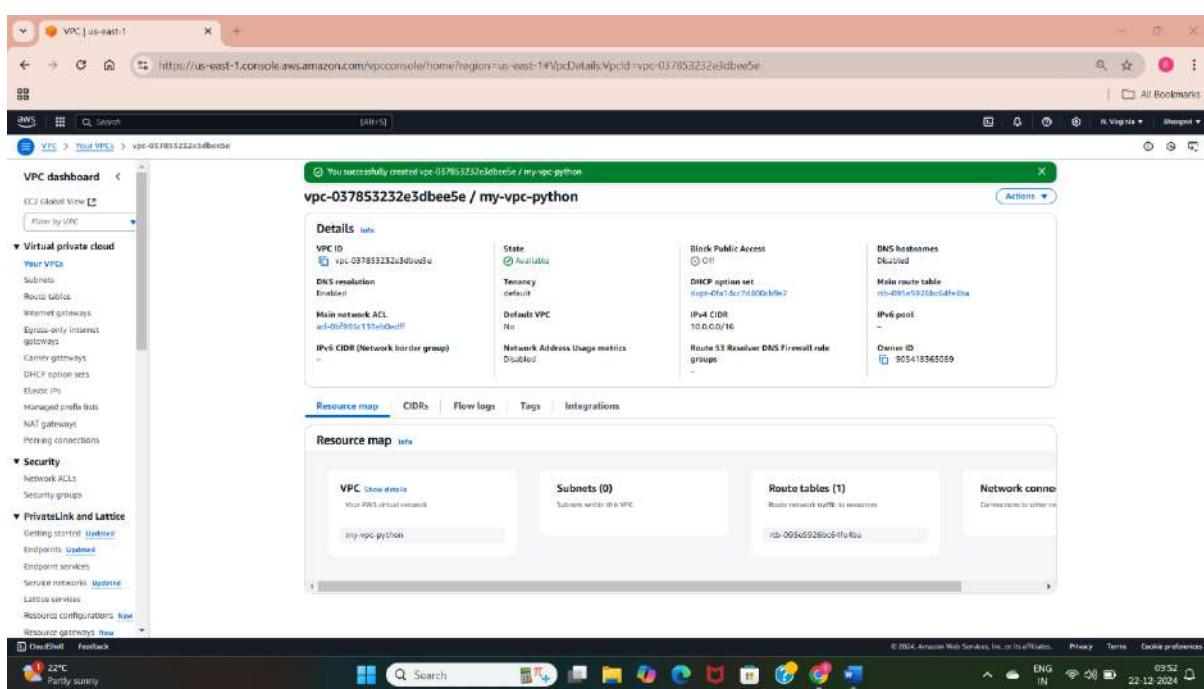
- AWS account  
Use an IAM user with the necessary permissions to interact with AWS services (e.g., EC2, S3, VPC, etc.).
- IAM user Terminal  
We care using the aws credentials i.e access keys and secret access keys to use aws service using the terraform.
- Basic understanding pf Python/flask
- Ensure Python is installed, then install Flask using pip.

## METHOD-1: TO DEPLOY A PYTHON APPLICATION USING MANUAL METHODS FIRST WE NEED TO CREATE A VIRTUAL PRIVATE NETWORK, SUBNETS, ROUTE TABLE, INTERNET GATEWAY

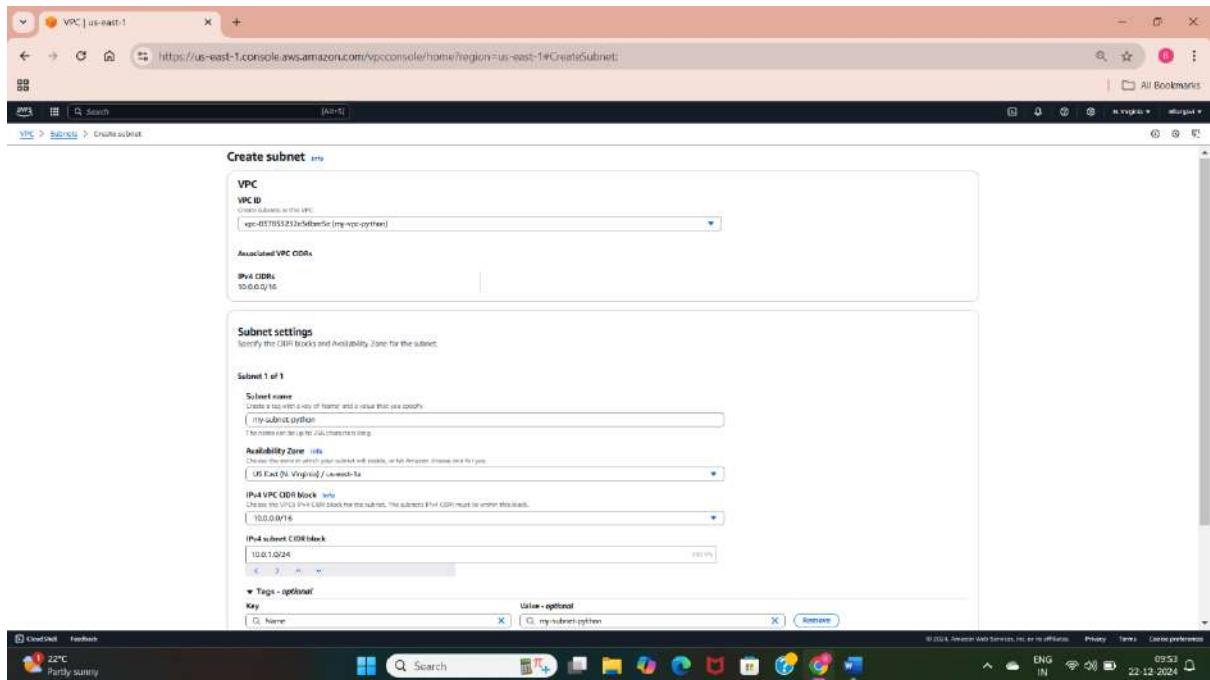
Virtual Private Cloud (VPC) is a isolated network environment with in the public cloud infrastructure. It allows users to create and manage private and secure network so that we can host resources like EC2 instances, and other AWS resources.



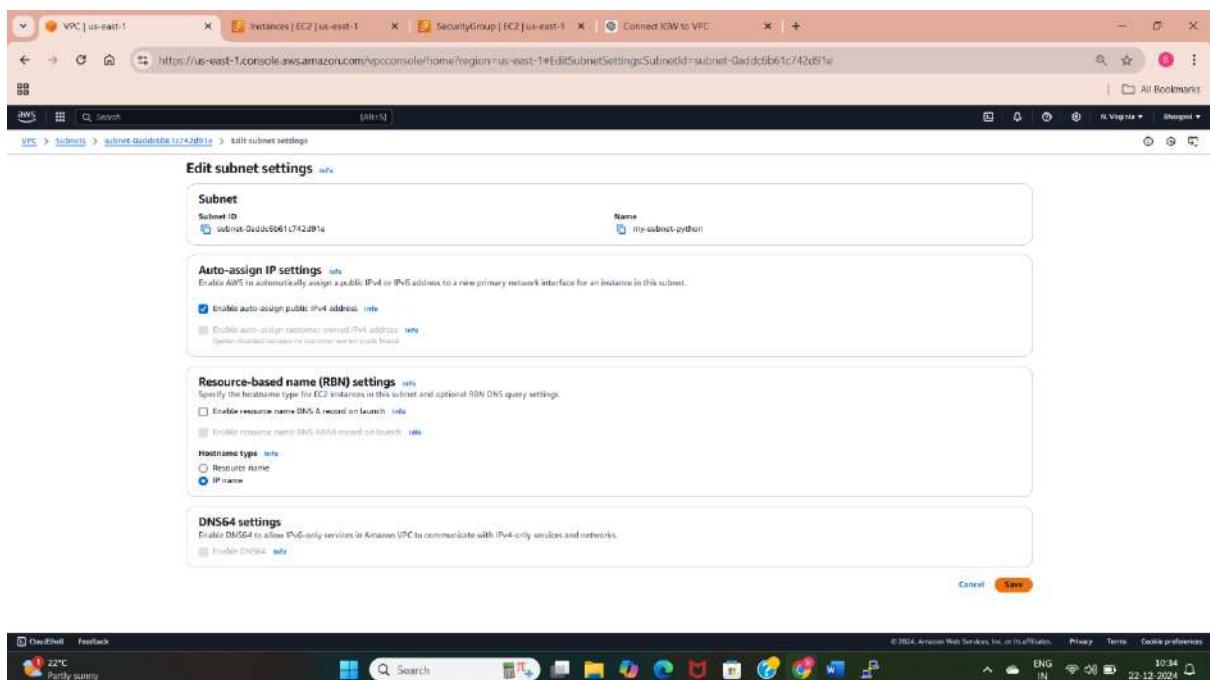
- Creating a VPC with the name of “my-vpc-python”, providing the ipv4 CIDR with 10.0.0.0/16



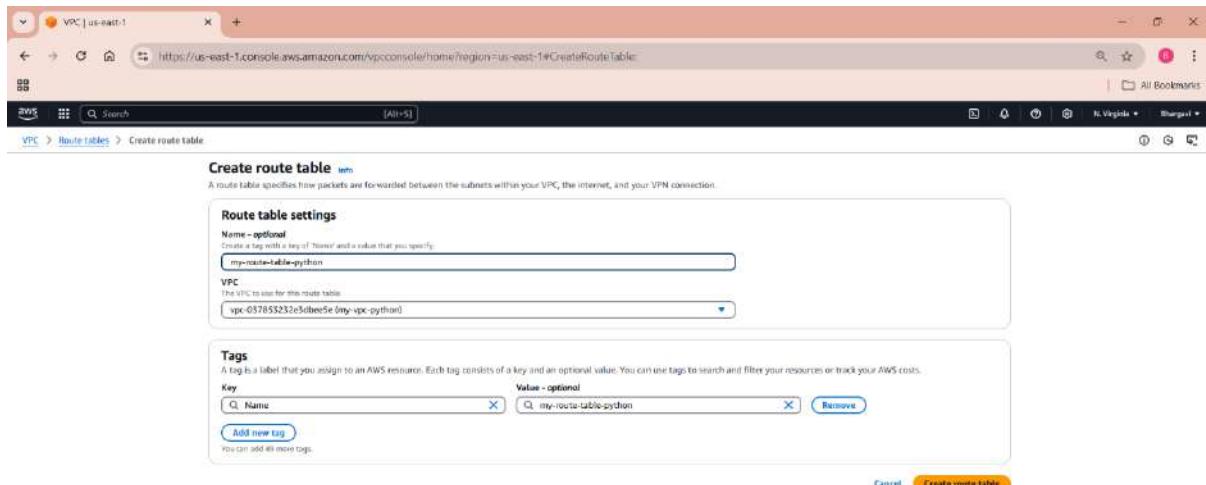
- VPC is created, we can see the vpc id, and other configurations related to the VPC.



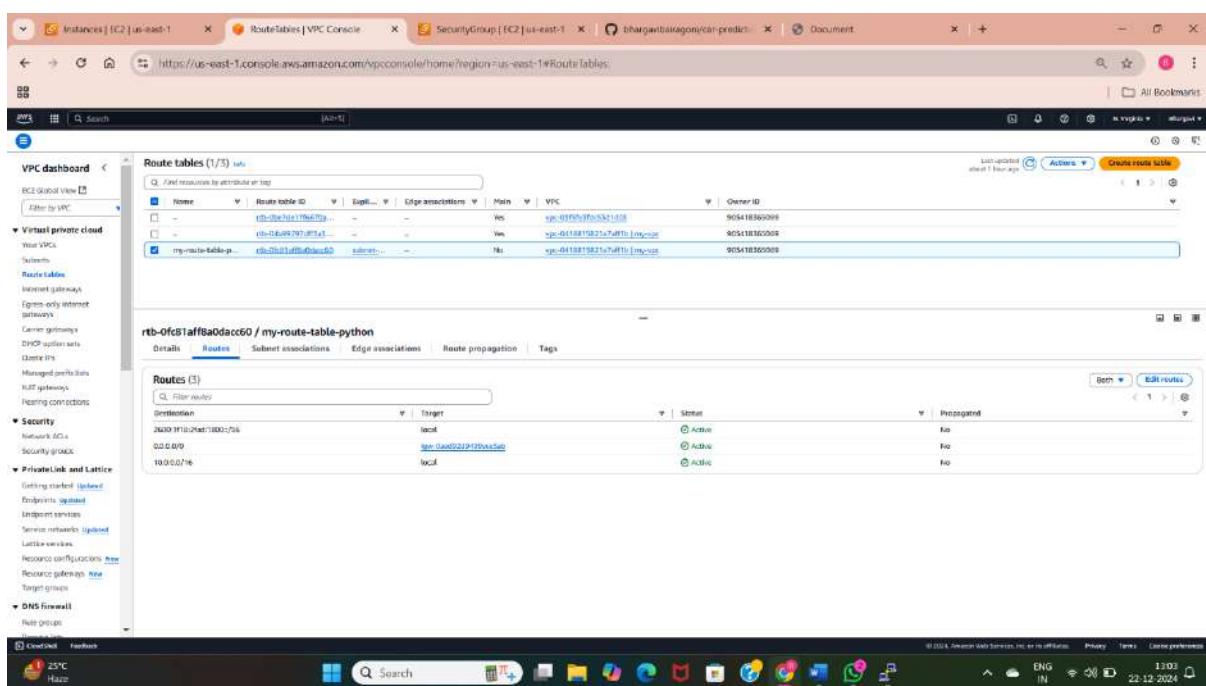
- ♣ Creating a subnet with the help of above created VPC.
- ♣ Here first we need to select the created VPC.
- ♣ Next, we need to give name for the subnet.
- ♣ Select any availability zone, if we do not select any availability zone it automatically creates in any one of the available zones.
- ♣ Provide the CIDR range for the subnet so that it assigns public IP addresses for the newly creating instance with this subnet, am giving the subnet CIDR “10.0.1.0/24” so that can get 256 IP addresses.



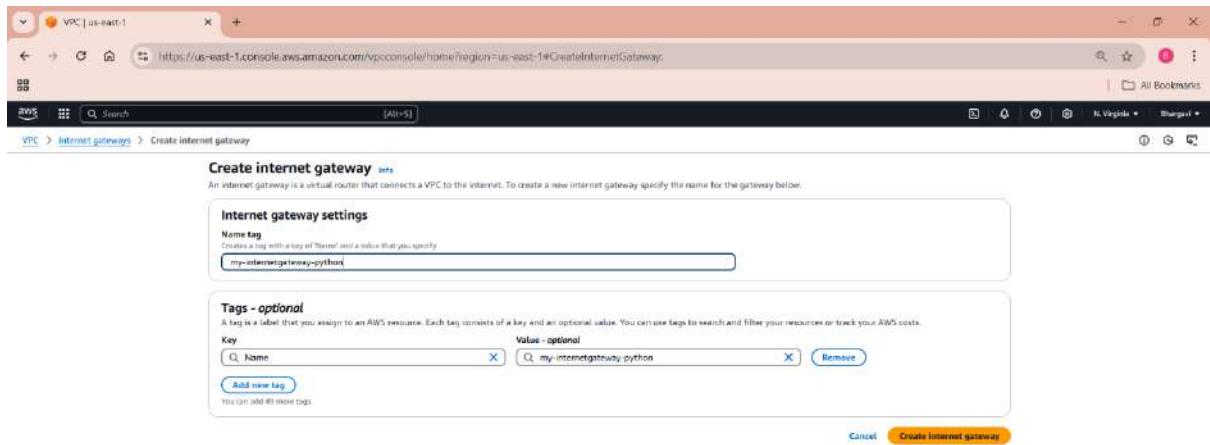
- ♣ While creating subnet we need to enable the auto-assign public IP addresses so that we can get public ip addresses for our newly creating instances.



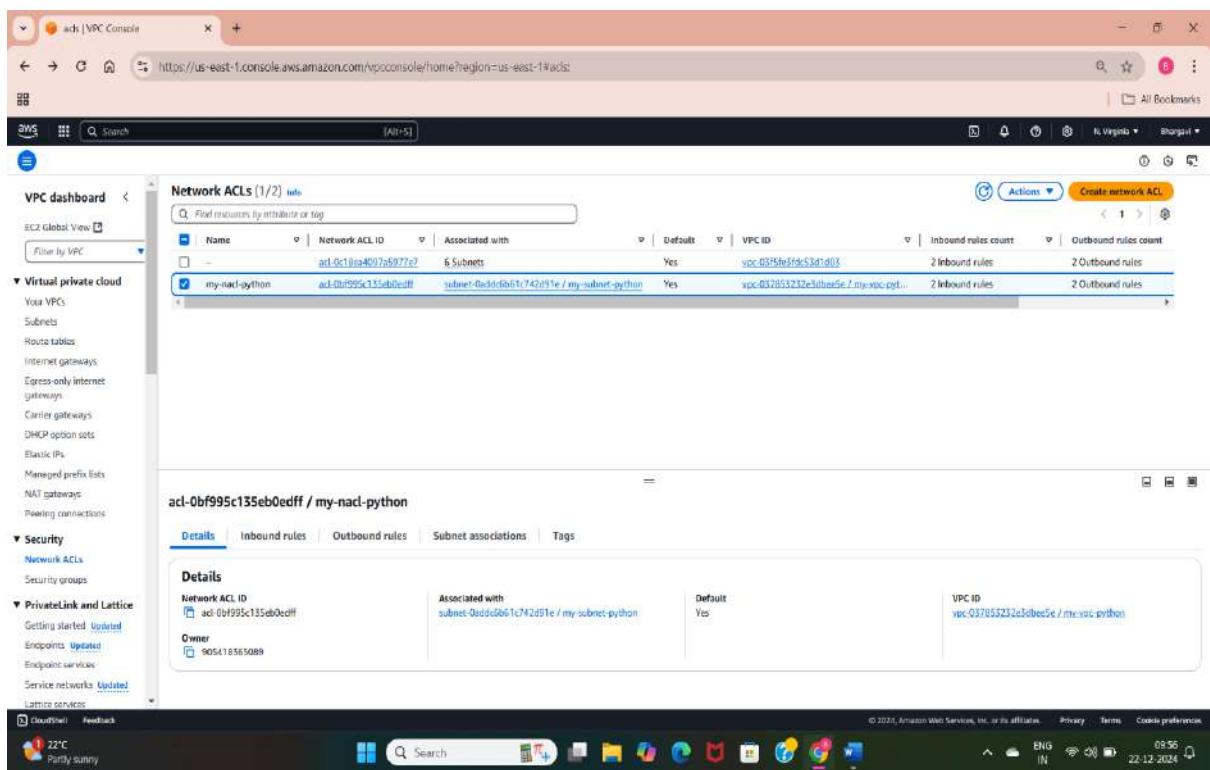
- Create a route table for above created VPC so that we can get traffic from other networks.
- Here we need to give the name for route table, select the created VPC.



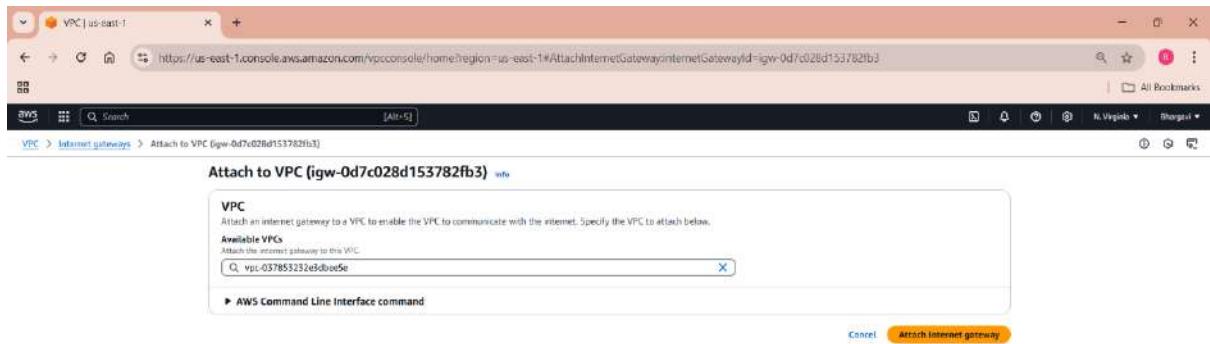
- Here we can see the configurations related to the route table which shows that CIDR which the traffic is distributed with in that IP addresses.



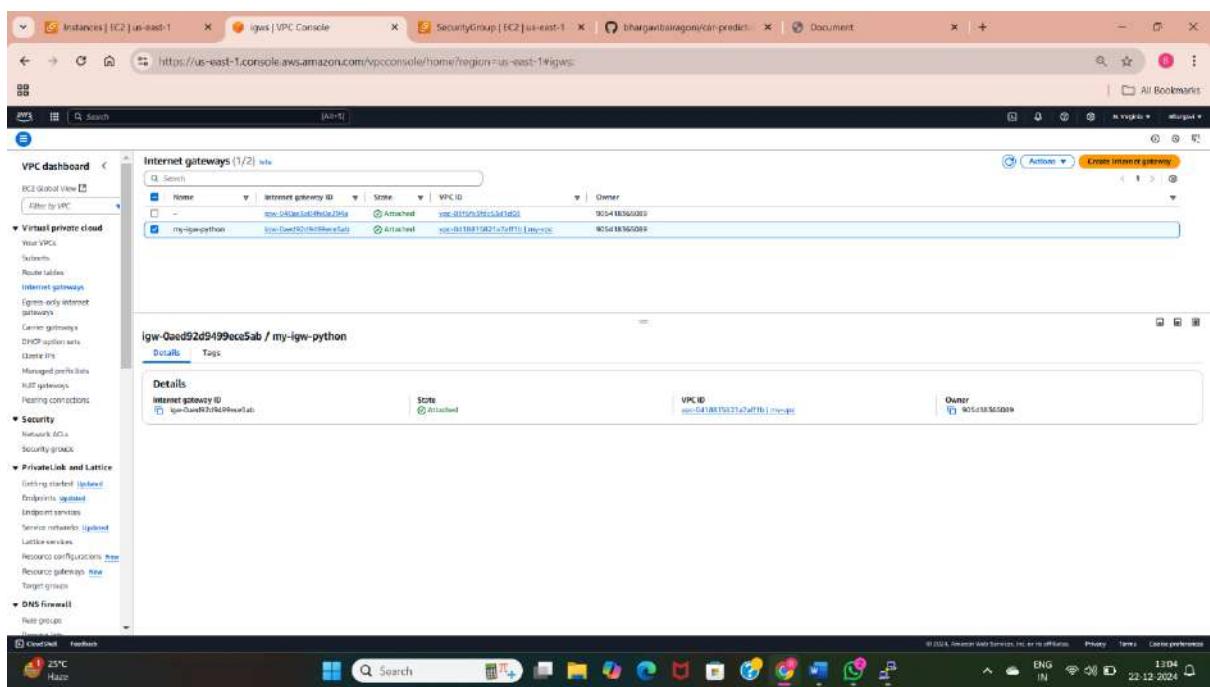
- ♣ Create a internet gateway for providing the internet for vpc and subnets.
- ♣ After creating internet gateway attach it to the vpc.



- ♣ Creating a Network Access Control List which is a security feature in AWS, and it acts as firewall for inbound and outbound traffic (for from and to traffic).



- Here am attaching the above created internet gateway to the VPC



- Here we can see that internet gateway is attached to VPC.

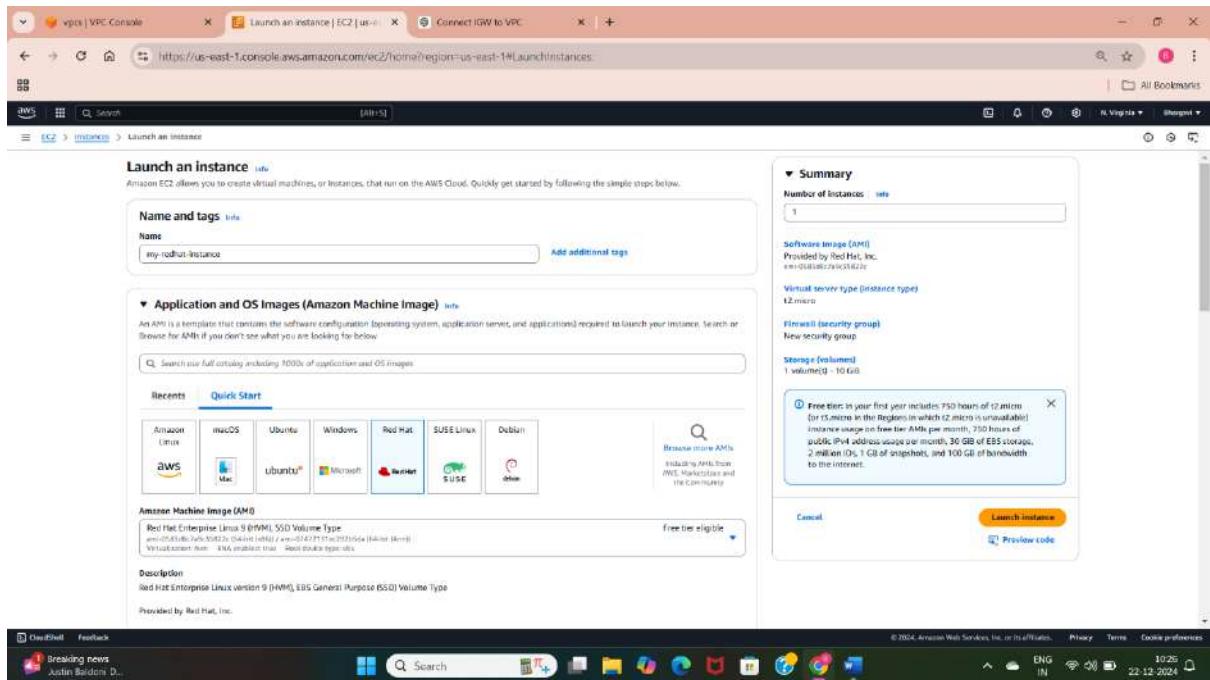
The screenshot shows the 'Edit routes' section of the AWS VPC console. A route entry for '10.0.0.0/16' is selected, pointing to 'igw-0d7c28d153782fb3'. The 'Status' is 'Active' and 'Propagated' is 'No'. Below the table, there are buttons for 'Cancel', 'Preview', and 'Save changes'.



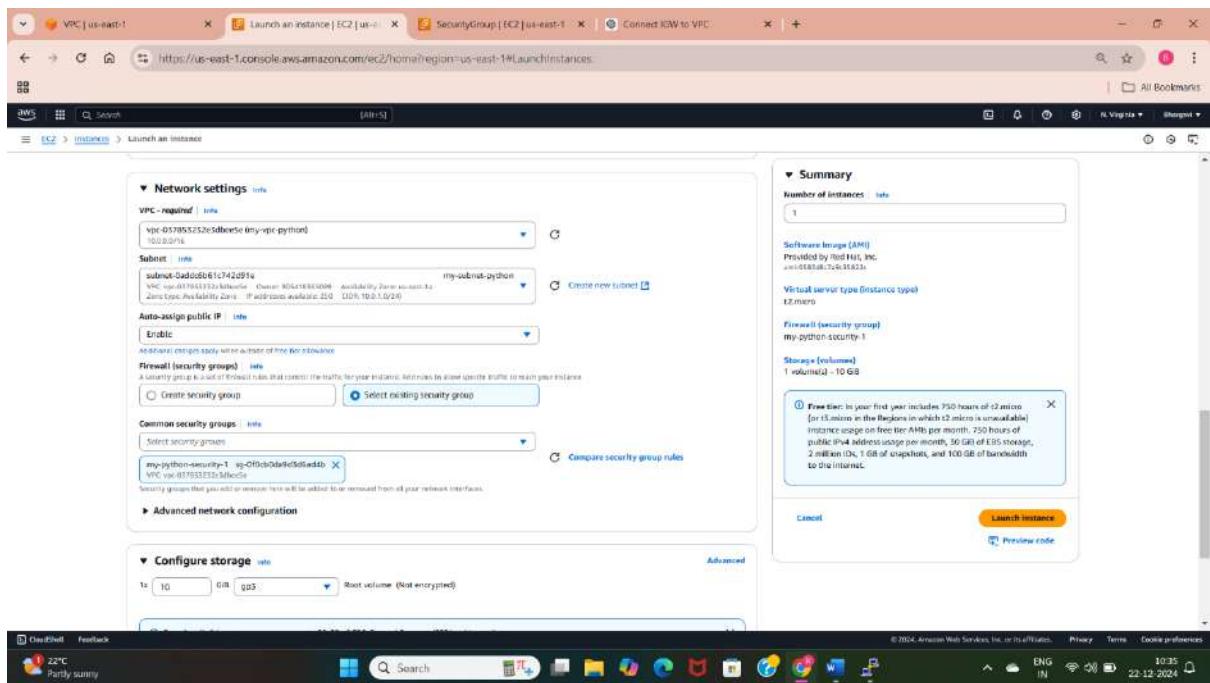
- In route table we need to set the target as above created igw to get internet from the internet gateway to the VPC, select the CIDR range for allowing the traffic.

The screenshot shows the 'Resource map' for the VPC 'my-vpc'. It displays the association between a subnet ('us-east-1a'), a route table ('rtt-048975f7ff5a54'), and an internet connection ('my-internet').

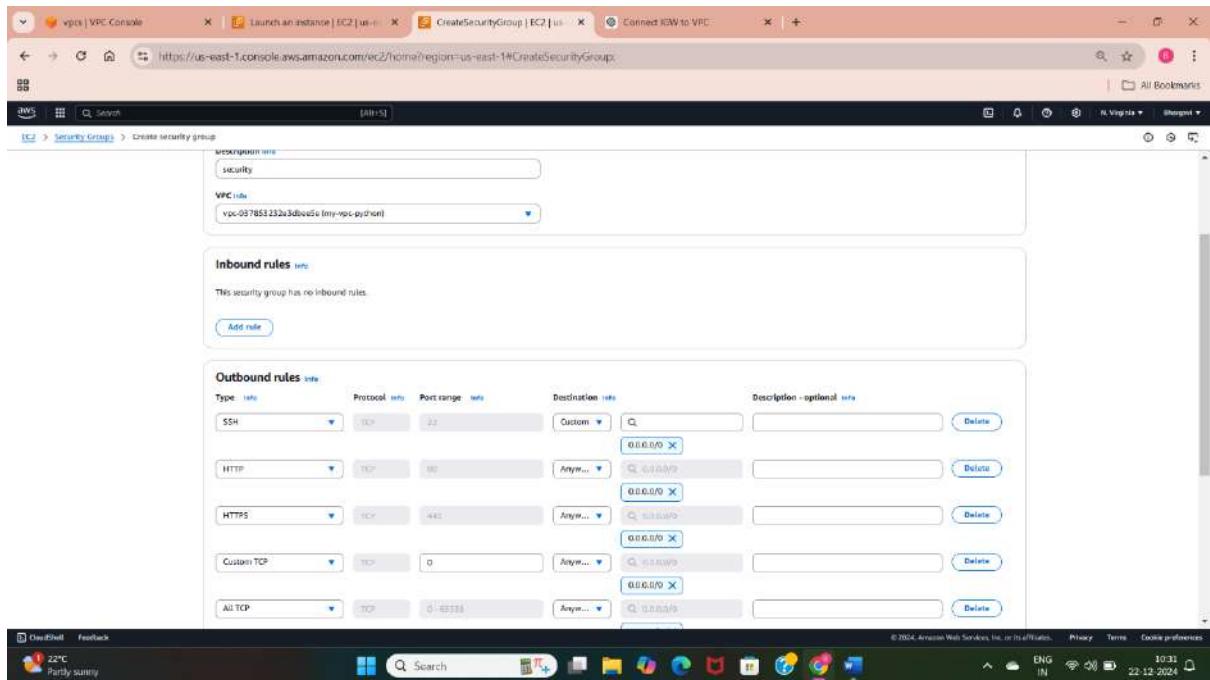
- Here we can see that the created subnet is associated with the route table and route table is associated with the internet gateway, so our VPC is connected to the internet gateway using subnet and route table.



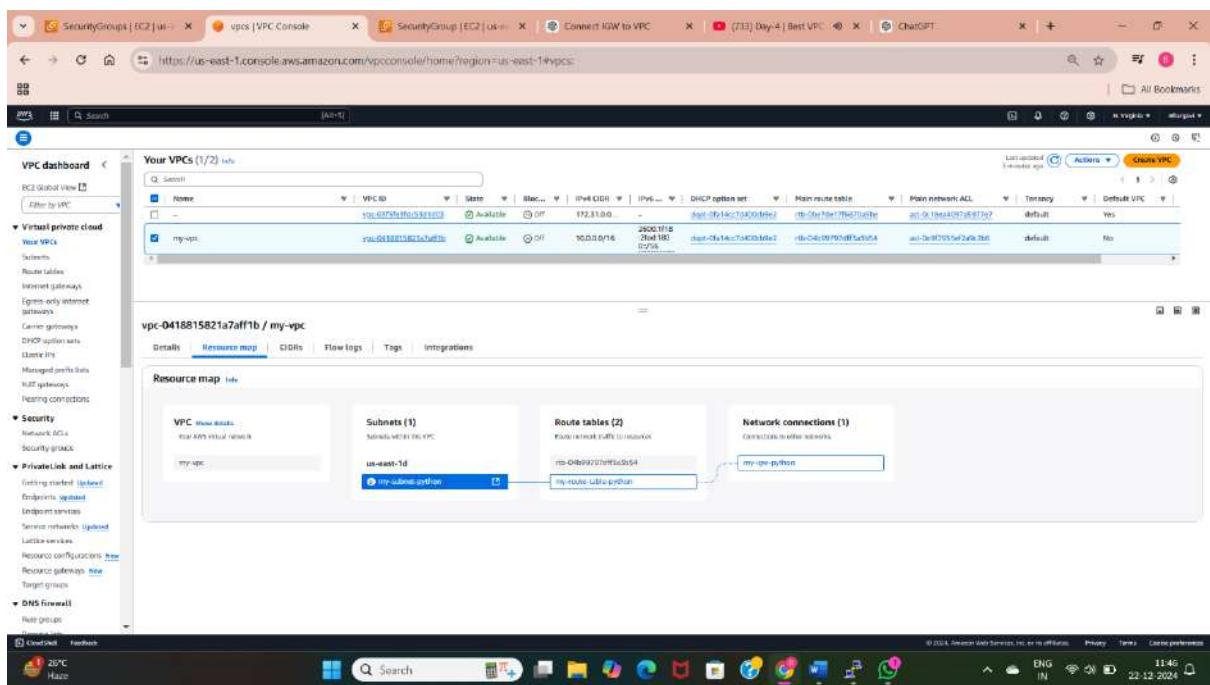
- Now am creating a EC2 instance with ubuntu.



- Now select the Network by editing the Network settings. Here am selecting the above created VPC, subnet and selecting the security group which is allowing some particular inbound ports.



- Here am creating a security group by allowing the above inbound ports.



- We can see that our subnet is associated with the route table and internet gateway is attached with the VPC.

```

root@ip-10-0-0-151:~# git init
Initialized empty Git repository in /root/.git/
[mon@ip-10-0-0-151 ~]# git remote add origin https://github.com/bhargavibairagi/car-prediction.git
[mon@ip-10-0-0-151 ~]# git pull origin master
remote: Enumerating objects: 10839, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (10/10), done.
remote: Total 10839 (delta 8), pack-reused 10837 (from 2)
Receiving objects: 100% (10830/10830), 96.95 MiB | 24.01 MiB/s, done.
Resolving deltas: 100% (563/563), done.
From https://github.com/bhargavibairagi/car-prediction
 * branch            master      -> FETCH_HEAD
 * [new branch]      master      -> origin/master
Updating files: 100% (10830/10830), done.
[mon@ip-10-0-0-151 ~]# ll
total 1256
-rw-r--r-- 1 root root    981 Dec 22 06:39 app.py
-rw-r--r-- 1 root root  16907 Dec 22 06:39 car_data.csv
-rw-r--r-- 1 root root     41 Dec 22 06:39 car_prediction deployment link.txt
drwxr-xr-x 1 root root 2275759 Dec 22 06:36 get-pip.py
-rw-r--r-- 1 root root    1694 Dec 22 06:39 gitattributes
-rw-r--r-- 1 root root 1792496 Dec 22 06:39 model.pkl
-rw-r--r-- 1 root root    156 Dec 22 06:39 OneHotEncoder.lib
-rw-r--r-- 1 root root    21 Dec 22 06:39 Profile
-rw-r--r-- 1 root root    305 Dec 22 06:39 requirements.txt
drwxr-xr-x 2 root root    33 Dec 22 06:39 static
-rw-r--r-- 1 root root 340300 Dec 22 06:39 Untitled.jupyter
[mon@ip-10-0-0-151 ~]# pip3 install -r requirements.txt
-bash: Pip3: command not found
[mon@ip-10-0-0-151 ~]# ^C

```

The terminal shows the user cloning a GitHub repository named "car-prediction" and navigating to its directory. They then run "pip3 install -r requirements.txt" but receive an error because they typed "Pip3" instead of "pip3".

- ♣ Install git and initialise it. Now take the python “car prediction” application from github.
- ♣ First I have added the remote repository in the terminal using “git remote add origin github url”
- ♣ I have pulled python application files from github.
- ♣ Install the requirements using “pip3 install -r requirements.txt”

```

root@ip-10-0-0-151:~#
root@ip-10-0-0-151:~# gunicorn --bind 0.0.0.0:8080 --workers 4 --access-logfile - --error-logfile - app:app
[mon@ip-10-0-0-151 ~]# netstat -an | grep 8080
[mon@ip-10-0-0-151 ~]# sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
[sudo] password for mon: 
[mon@ip-10-0-0-151 ~]# sudo firewall-cmd --reload
[sudo] password for mon: 
[mon@ip-10-0-0-151 ~]# ^C
[mon@ip-10-0-0-151 ~]# vim requirements.txt
[mon@ip-10-0-0-151 ~]# vim app.py
[mon@ip-10-0-0-151 ~]# python3 app.py
[mon@ip-10-0-0-151 ~]# curl http://127.0.0.1:8080
<!DOCTYPE html><html><head><title>sklearn</title></head><body><p>UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to: https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>/root@ip-10-0-0-151:~/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:<br/>https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>/root@ip-10-0-0-151:~/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:<br/>https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>/root@ip-10-0-0-151:~/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:<br/>https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>/root@ip-10-0-0-151:~/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:<br/>https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>/root@ip-10-0-0-151:~/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:<br/>https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations<br/>UserWarning,<br/>* Debugger PIN: 101-069-954
[202.254.166.133] - [22/Dec/2024 07:32:30] "GET / HTTP/1.1" 200 -
[202.254.166.133] - [22/Dec/2024 07:32:32] "GET /favicon.ico HTTP/1.1" 404 -
[mon@ip-10-0-0-151 ~]# ^C
[mon@ip-10-0-0-151 ~]# exit
[mon@ip-10-0-0-151 ~]#

```

The terminal shows the user starting a gunicorn server for their Python application. It also shows them running curl to test the application and receiving several "UserWarning" messages about incompatible estimator versions. Finally, they press Ctrl+C to stop the process.

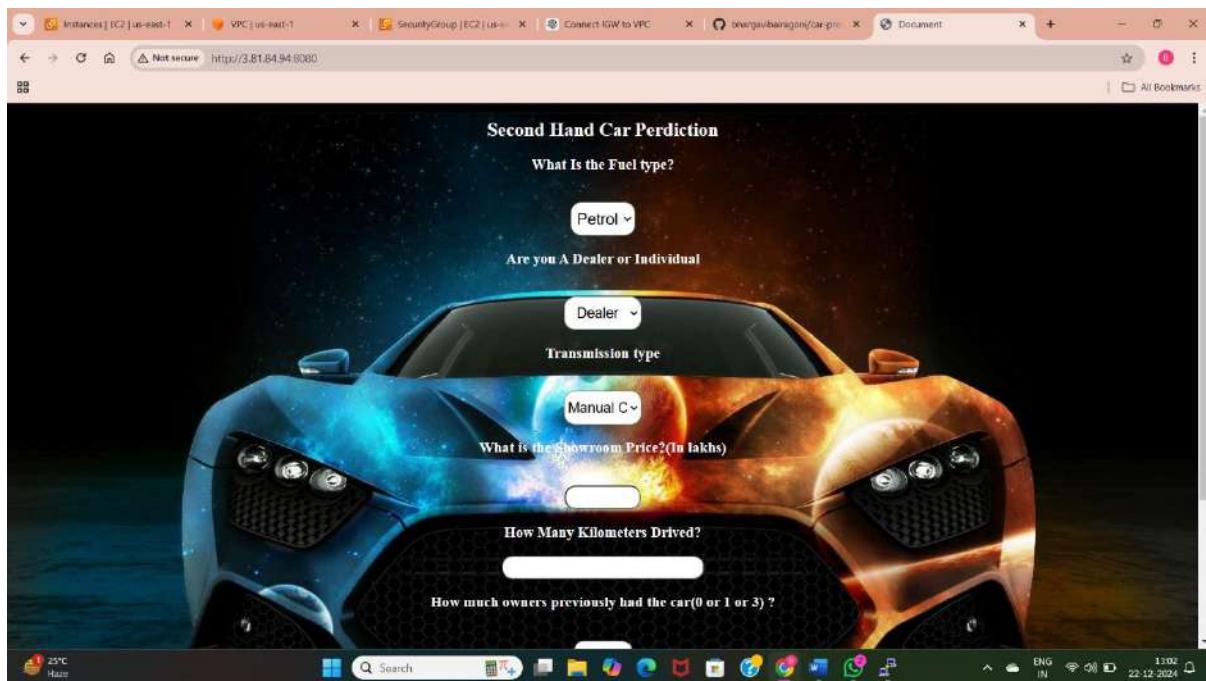
- ♣ Change the host port to 0.0.0.0 in the app.py file.
- ♣ Now deploy the python application using “python3 app.py”

```

root@ip-10-0-0-151:~#
1 ping google.com
2 sudo yum update -y
3 sudo yum install python3 -y
4 sudo yum install python3-pip -y
5 pip3 --version
6 curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
7 sudo python3 get-pip.py
8 pip3 --version
9 yum install git -y
10 git init
11 git remote add origin https://github.com/bhargavibairagchi/car-prediction.git
12 git pull origin master
13 ll
14 pip3 install -r requirements.txt
15 pip3 install -r requirements.txt
16 python3 app.py
17 pip3 install upgrade pip
18 pip3 install -r requirements.txt
19 pip3 install numpy==1.23.3
20 pip3 install numpy==1.22.0
21 python3.7 app.py
22 python3.7 -m pip install Flask
23 python3.7 -m pip install -r requirements.txt
24 python3.7 app.py
25 python3.7 -m pip install numpy==1.21.6
26 python3.7 app.py
27 python3.7 -m pip install pandas
28 python3.7 app.py
29 python3.7 -m pip install joblib
30 python3.7 app.py
31 python3.7 -m pip install scikit-learn
32 python3.7 app.py
33 python3.7 -m pip install scikit-learn==1.1.2
34 python3.7 --version
35 python3.7 -m pip install scikit-learn==1.3.2
36 python3.7 -m pip install scikit-learn==1.3.2
37 python3.7 -m pip install scikit-learn==1.0.2
38 python3.7 --version
39 python3.7 app.py
40 sudo vim app.py
41 vim app.py
42 Sudo vim app.py
43 python3.7 app.py
44 sudo vim app.py
45 python3.7 app.py
46 sudo vim app.py
47 python3.7 app.py
48 sudo vim app.py
49 python3.7 app.py

```

- These are the commands used to deploy the car-prediction application.



- Now copy the public IP address of the instance and use the car prediction application port and access this application.

```

root@ip-10-0-0-151:~/Fish
113 ps aux | grep python
113 netstat -tuln | grep 8080
114 sudo firewall-cmd --zone=public --add-port=8080/tcp --permanent
115 sudo firewall-cmd --reload
116 vi requirements.txt
117 vi app.py
118 python app.py
119 cd ..
120 history
myenv [root@ip-10-0-0-151 ~]# cd ..
-bash: cd: ..: No such file or directory
myenv [root@ip-10-0-0-151 ~]# cd Fish/
myenv [root@ip-10-0-0-151 Fish]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:    git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:    git branch -m <name>
Initialized empty Git repository in /root/Fish/.git/
myenv [root@ip-10-0-0-151 Fish]# git remote add origin https://github.com/bharatvibhavagai/fish.git
myenv [root@ip-10-0-0-151 Fish]# git pull origin master
remote: Enumerating objects: 10765, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 10765 (delta 2), reused 2 (delta 2), pack-reused 10760 (from 2)
Receiving objects: 100% (10765/10765), 81.40 MiB / 23.69 MiB/s, done.
Resolving deltas: 100% (804/804), done.
From https://github.com/bharatvibhavagai/fish
 * branch          master       -> FETCH_HEAD
 * [new branch]   master      -> origin/master
Updating 100% (10767/10767), done.
myenv [root@ip-10-0-0-151 Fish]# ls
total 1220
-rw-r--r-- 1 root root 1007 Dec 22 07:37 app.py
drwxr-xr-x 4 root root 50 Dec 22 07:37 fish
-rw-r--r-- 1 root root 5862 Dec 22 07:37 Fish.csv
-rw-r--r-- 1 root root 469729 Dec 22 07:37 Fish market.ipynb
-rw-r--r-- 1 root root 1698 Dec 22 07:37 gitattributes
-rw-r--r-- 1 root root 1059 Dec 22 07:37 .gitignore
-rw-r--r-- 1 root root 1059 Dec 22 07:37 .gitmodules
-rw-r--r-- 1 root root 828061 Dec 22 07:37 Random.pkl
-rw-r--r-- 1 root root 125 Dec 22 07:37 requirements.txt
drwxr-xr-x 2 root root 28 Dec 22 07:37 templates
myenv [root@ip-10-0-0-151 Fish]#

```

- ♣ We need to deploy other python application.
- ♣ Now taking the “fish python application” from the github using the remote repository.
- ♣ We can see the fish application files in the above picture.
- ♣ Install the requirements of fish application.

```

root@ip-10-0-0-151:~/Fish
110 flask layout Flask,render_template,url_for,request,redirect
111 import numpy as np
112 import pandas as pd
113 import joblib
114 import pickle

115 app = Flask(__name__)

model = joblib.load('Random.pkl')
encoder = joblib.load('enc.pkl')

116 @app.route('/')
117 @app.route('/main')
118 def main():
119     return render_template('main.html')

120 @app.route('/predict',methods=['POST'])
121 def predict():
122     int_features = [x for x in request.form.values()]
123     print(int_features)
124     print(encoder)
125     print(encoder.get_feature_names_out())
126     v = ['Species', 'Length', 'Length0', 'Length1', 'Height', 'Width']
127     df = pd.DataFrame(int_features,columns=v)
128     i = oshash.transform(df.iloc[0,:])
129     e = encoder.get_feature_names_out()
130     t = pd.DataFrame(i,columns=e)
131     df = t
132     final = pd.concat([t,i],axis=1)
133     result = model.predict(final)
134     print("The estimated fish weight is : ",result)

135     print(encoder.get_feature_names_out())
136
137     return render_template('main.html',prediction_text="The estimated fish weight is : {} in gms".format(result))

138 if __name__ == "__main__":
139     app.debug=True
140     app.run(host = "0.0.0.0",port=8080)
141     app.run(host = "0.0.0.0",port=80)

```

- ♣ Now change the host port to 0.0.0.0/0 in app.py file of fish application.

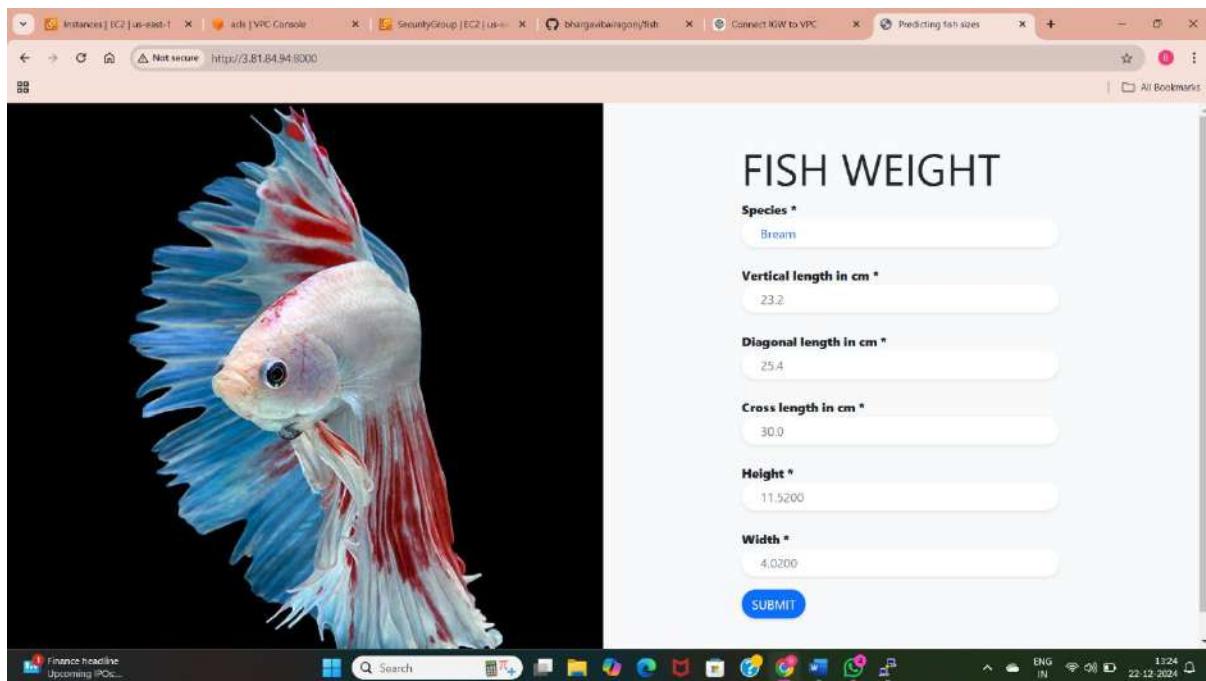
```

root@ip-10-0-0-151:~/Penguin
(mysuvu) [root@ip-10-0-0-151 Fish]# cd Fish/
(mysuvu) [root@ip-10-0-0-151 Fish]# ll
total 1320
-rw-r--r-- 1 root root 1007 Dec 22 07:37 app.py
drwxr-xr-x 4 root root 50 Dec 22 07:37 fish
-rw-r--r-- 1 root root 3863 Dec 22 07:37 FishSV
-rw-r--r-- 1 root root 4979 Dec 22 07:37 Market.ipynb
-rw-r--r-- 1 root root 1694 Dec 22 07:37 gitattributes
-rw-r--r-- 1 root root 1039 Dec 22 07:37 one.joblib
-rw-r--r-- 1 root root 21 Dec 22 07:37 profile
-rw-r--r-- 1 root root 628061 Dec 22 07:37 Random.pkl
-rw-r--r-- 1 root root 325 Dec 22 07:38 requirements.txt
drwxr-xr-x 2 root root 23 Dec 22 07:37 templates
(mysuvu) [root@ip-10-0-0-151 Fish]# vim requirements.txt
(mysuvu) [root@ip-10-0-0-151 Fish]# vim app.py
(mysuvu) [root@ip-10-0-0-151 Fish]# python3 app.py
  File "app.py", line 49
    app.run(host='0.0.0.0', port=8000, debug=True)

IndentationError: unindent does not match any outer indentation level
(mysuvu) [root@ip-10-0-0-151 Fish]# vim app.py
(mysuvu) [root@ip-10-0-0-151 Fish]# python3 app.py
/root/.myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.1 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
/root/.myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.1 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
* Serving local app 'app'
* Using mode: on
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://127.0.0.1:8000
* Running on http://10.0.0.151:8000
Press CTRL+C to quit
* Restarting with stat
/root/.myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.1 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
/root/.myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.1 when using version 1.0.2. This might lead to a breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,

```

- Now deploy the python application using “python3 app.py” or “screen -m -d python3 ./app.py &”.



- This is the fish application. We can access this with the public IP and application port number.

```

root@ip-10-0-0-151:~/Flight
(myenv) [root@ip-10-0-0-151 Flight] ~$ mkdir Flight
(myenv) [root@ip-10-0-0-151 Flight] ~$ cd Flight
(myenv) [root@ip-10-0-0-151 Flight] ~$ git remote add origin https://github.com/bhargavibairagi/flight-prediction.git
(myenv) [root@ip-10-0-0-151 Flight] ~$ git init
(myenv) [root@ip-10-0-0-151 Flight] ~$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
initialized empty Git repository in /root/Flight/.git/
(myenv) [root@ip-10-0-0-151 Flight] ~$ git pull origin master
(myenv) [root@ip-10-0-0-151 Flight] ~$ git pull origin master
remote: Enumerating objects: 10831, done.
remote: Counting objects: 1000 (6/8), done.
remote: Compressing objects: 1000 (0/6), done.
remote: Total 10831 (delta 6), reused 0 (delta 0), pack-reused 10223 (from 3)
remote: Preparing objects: 1000 (10831/10831) 4.41 MiB | 24.28 MiB/s, done.
remote: Resolving deltas: 1000 (1000/1000) 0% (0/0), done.
remote: From https://github.com/bhargavibairagi/flight-prediction
 * [branch]          master      -> origin/master
 * [new branch]       master      -> origin/master
Updating files: 100% (10351/10351), done.
(myenv) [root@ip-10-0-0-151 Flight] ~$ ls
total 13700
-rw-r--r-- 1 root root    962 Dec 22 07:56 app.py
-rw-r--r-- 1 root root  53089 Dec 22 07:56 Data_Train.xlsx
-rw-r--r-- 1 root root  45615 Dec 22 07:56 Data_Train_Price_Prediction.ipynb
-rw-r--r-- 1 root root     40 Dec 22 07:58 flight price prediction deployment link.txt
-rw-r--r-- 1 root root   1696 Dec 22 07:58 gitattributes
-rw-r--r-- 1 root root 12886240 Dec 22 07:58 model.pkl
-rw-r--r-- 1 root root   2172 Dec 22 07:58 OneHotEncoder.pkl
drwxr-xr-x 4 root root     50 Dec 22 07:58 price
-rw-r--r-- 1 root root    21 Dec 22 07:58 Proctitle
-rw-r--r-- 1 root root   181 Dec 22 07:58 requirements.txt
drwxr-xr-x 2 root root     23 Dec 22 07:58 templates
-rw-r--r-- 1 root root 10279 Dec 22 07:58 test_src.xlsx
(myenv) [root@ip-10-0-0-151 Flight] ~$ pip3 install -r requirements.txt
Requirement already satisfied: click in /root/myenv/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (8.1.6)
Collecting colorama (from -r requirements.txt (line 2))
  Downloading colorama-0.4.6-py3-none-any.whl.metadata (17 kB)


```

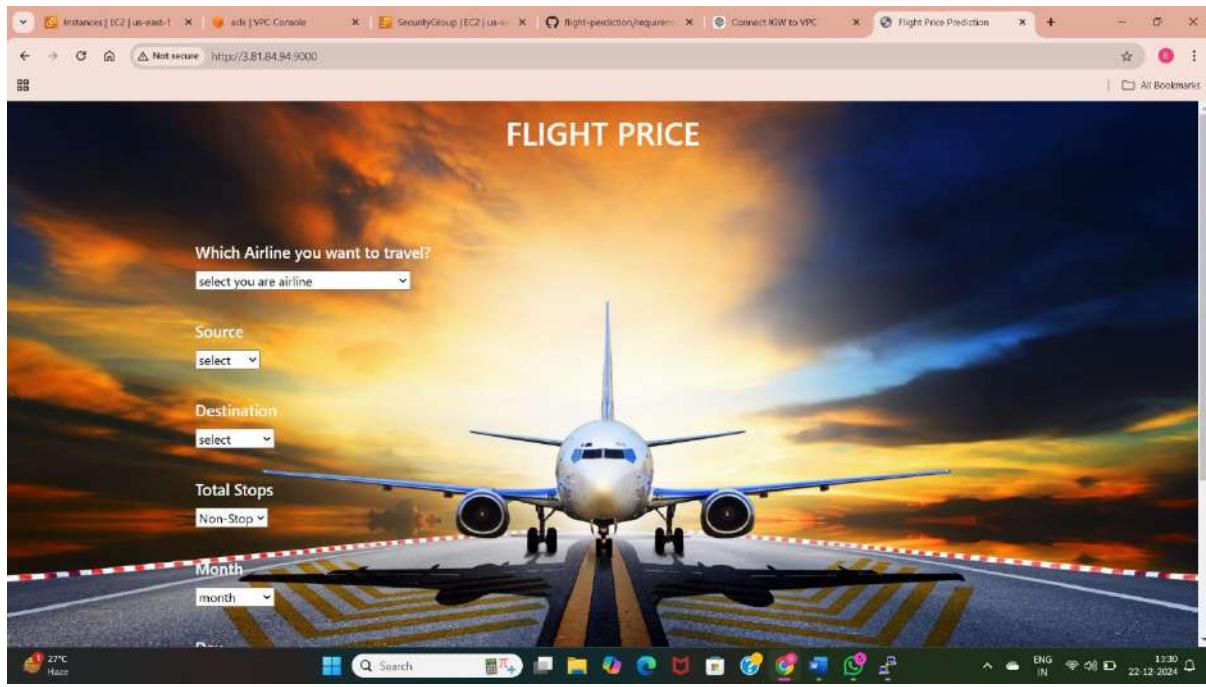
- ♣ Now am deploying the other python application i.e Flight.
- ♣ Take the code from github.
- ♣ Change the host port 0.0.0.0/0 of the application by moving into app.py.
- ♣ Install the requirements of this application.

```

root@ip-10-0-0-151:~/Pengmain
WARNING: Skipping scikit-learn as it is not installed
(myenv) [root@ip-10-0-0-151 Pengmain] ~$ pip3 install scikit-learn
Requirement already satisfied: scikit-learn in /root/myenv/lib/python3.7/site-packages (1.20.2)
Requirement already satisfied: numpy>=1.14.6 in /root/myenv/lib/python3.7/site-packages (from scikit-learn) (1.20.2)
Requirement already satisfied: scipy>=1.1.0 in /root/myenv/lib/python3.7/site-packages (from scikit-learn) (1.7.3)
Requirement already satisfied: joblib<0.11 in /root/myenv/lib/python3.7/site-packages (from scikit-learn) (1.3.2)
Requirement already satisfied: threadpoolctl<=2.0.0 in /root/myenv/lib/python3.7/site-packages (from scikit-learn) (3.1.0)
(myenv) [root@ip-10-0-0-151 Pengmain] ~$ python3 app.py
/root/myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
/root/myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
* Starting Flask app 'app'
* Debug mode: on
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://127.0.0.1:9000
* Running on http://10.0.0.151:9000
Press CTRL+C to quit
* Restarting with stat
/root/myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
/root/myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
/root/myenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
* Debugger is active!
* Debugger PIN: 101-069-954
205.254.168.133 - [22/Dec/2024 08:00:47] "GET / HTTP/1.1" 200 -
205.254.168.133 - [22/Dec/2024 08:00:47] "GET /static/css/styles.css HTTP/1.1" 404 -
205.254.168.133 - [22/Dec/2024 08:00:49] "GET /favicon.ico HTTP/1.1" 404 -
(myenv) [root@ip-10-0-0-151 Pengmain] ~$ cd Pengmain
(myenv) [root@ip-10-0-0-151 Pengmain] ~$ git init
(myenv) [root@ip-10-0-0-151 Pengmain] ~$ git init

```

- ♣ Now am deploying the application using “python3 app.py”. we can see that the python application is deployed successfully.



- This is the flight application which is deployed using the manual method.

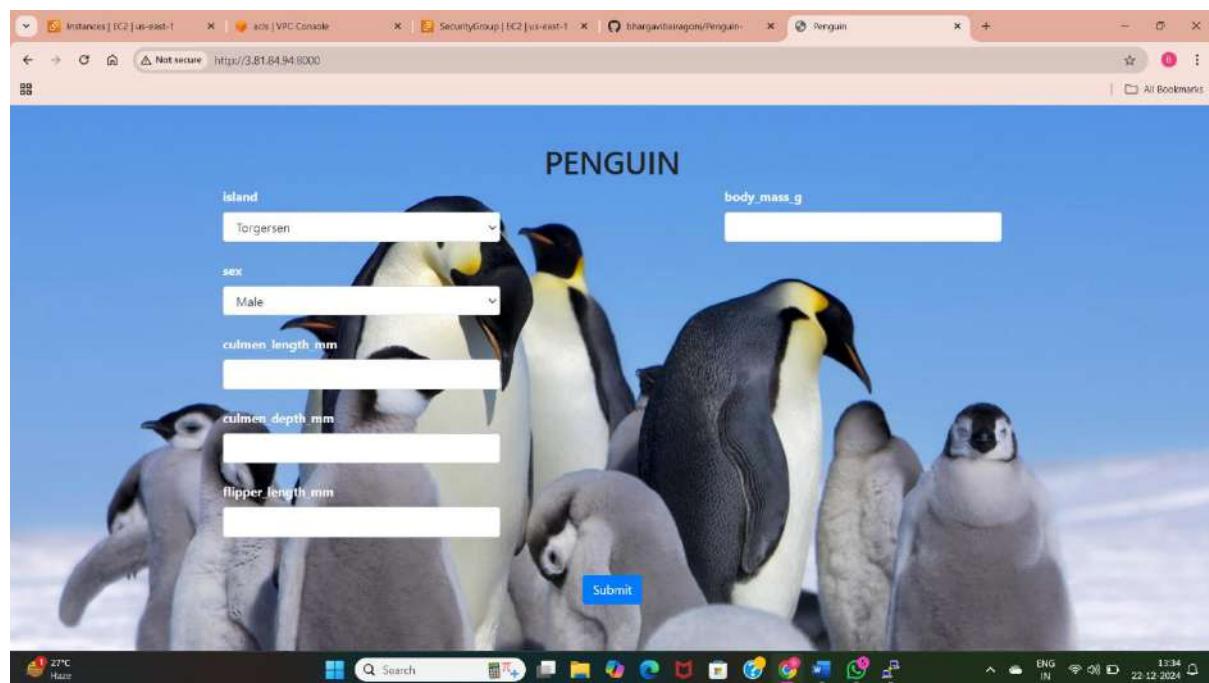
```

root@ip-10-0-0-151:/Penguin
[root@ip-10-0-0-151 python3.7/site-packages/sklearn/base.py:330]: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
http://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
* Debugger PIN: 101-069-854
203.254.1.68.133 - - [22/Dec/2024 08:00:13] "GET / HTTP/1.1" 200 -
203.254.1.68.133 - - [22/Dec/2024 08:00:13] "GET /static/css/styles.css HTTP/1.1" 404 -
203.254.1.68.133 - - [22/Dec/2024 08:00:14] "GET /favicon.ico HTTP/1.1" 404 -
[myenv] [root@ip-10-0-0-151 Penguin] cd
[myenv] [root@ip-10-0-0-151 Penguin] cd Penguin/
[myenv] [root@ip-10-0-0-151 Penguin] git init
hint: using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint: git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /root/Penguin/.git/
[myenv] [root@ip-10-0-0-151 Penguin] git remote add origin https://github.com/Sharpavibhargoni/Penguin-.git
[myenv] [root@ip-10-0-0-151 Penguin] git pull origin master
remote: Enumerating objects: 100%, done.
remote: Counting objects: 100%, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 10784 (delta 0), reused 0 (delta 0), pack-reused 10782 (from 0)
Receiving objects: 100% (10784/10784), 81.28 MiB / 23.55 MiB/s, done.
Resolving deltas: 100% (815/815), done.
From https://github.com/Sharpavibhargoni/Penguin-
 * branch            master      -> FETCH_HEAD
 * new branch        master      -> origin/master
Updating file: 10788 (10788/10788), done.
[myenv] [root@ip-10-0-0-151 Penguin] ll
total 1500
drwxr-xr-- 1 root root   1603 Dec 22 08:02 app.py
-rw-r--r-- 1 root root  3808 Dec 22 08:02 classifier.pkl
-rw-r--r-- 1 root root  97535 Dec 22 08:02 Decision Tree using penguins Data.ipynb
-rw-r--r-- 1 root root  1696 Dec 22 08:02 gitattributes
-rw-r--r-- 1 root root  1264 Dec 22 08:02 OneHotEnc.joblib
-rw-r--r-- 1 root root 35740 Dec 22 08:02 Penguin.ipynb
-rw-r--r-- 1 root root  21 Dec 22 08:02 Proctfile
-rw-r--r-- 1 root root  325 Dec 24 08:02 requirements.txt
drwxr-xr-x 4 root root  1024 Dec 22 08:02 static
drwxr-xr-x 2 root root  23 Dec 22 08:02 templates
[myenv] [root@ip-10-0-0-151 Penguin]#

```

- Now take other python application from the github. Am taking the penguin application.
- Install the requirements, change the host in the app.py of penguin application.

- After successfully installing requirements we need to deploy the penguin application using using “python3 app.py”. We can see the deployment is successful.



- Now by using the port number of application and public IP address of EC2 instance we can able to access the penguin application.

```

root@ip-10-0-0-151:~/Innews
UserWarning:
  * Debugger is active!
  * Debugger PIN: 101-069-854
20.254.166.133 - [22/Dec/2024 08:03:56] "GET / HTTP/1.1" 200 -
* [myenv] [root@ip-10-0-0-151 Innews] cd
* [myenv] [root@ip-10-0-0-151 Innews] cd
* [myenv] [root@ip-10-0-0-151 Innews] git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/Innews/.git/
[myenv] [root@ip-10-0-0-151 Innews] git remote add origin https://github.com/bhargevibairagi/InNews.git
[myenv] [root@ip-10-0-0-151 Innews] git pull origin master
remote: Enumerating objects: 57, done.
remote: Counting objects: 100% (27/27), done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 57 (delta 19), pack-reused 30 (from 1)
Unpacking objects: 100% (57/57), 331.7 KiB in 10.90 MiB, done.
From https://github.com/bhargevibairagi/InNews
 * [branch]      master    -> origin/master
[myenv] [root@ip-10-0-0-151 Innews] ll
total 340
-rw-r--r-- 1 root root 4880 Dec 22 08:05 App.py
drwxr-xr-x 2 root root 68 Dec 22 08:05 Meta
-rw-r--r-- 1 root root 1373 Dec 22 08:05 README.md
-rw-r--r-- 1 root root 53 Dec 22 08:05 requirements.txt
-rw-r--r-- 1 root root 1606 Dec 22 08:05 simple_news_scrap.py
-rw-r--r-- 1 root root 12570 Dec 22 08:05 thumb.jpg
[myenv] [root@ip-10-0-0-151 Innews] pip3 install -r requirements.txt
Collecting pillow (from -r requirements.txt (line 1))
  Downloading Pillow-9.5.0-cp37-cp37m-manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.5 kB)
Collecting newspaper3k (from -r requirements.txt (line 2))
  Downloading newspaper3k-0.2.0-py3-none-any.whl.metadata (11 kB)
Collecting urllib3 (from -r requirements.txt (line 3))
  Downloading urllib3-2.0.0-py3-none-any.whl.metadata (6.6 kB)
Collecting streamlit (from -r requirements.txt (line 4))
  Downloading streamlit-1.23.3-py3-none-any.whl.metadata (0.4 kB)
Collecting beautifulsoup4 (from -r requirements.txt (line 5))
  Downloading beautifulsoup4-4.12.3-py3-none-any.whl.metadata (3.8 kB)
Collecting protobuf==3.20.0 (from -r requirements.txt (line 6))

```

- ♣ Now I want to deploy Innews python application.
- ♣ Take the code from github.
- ♣ Install the requirements of innews.

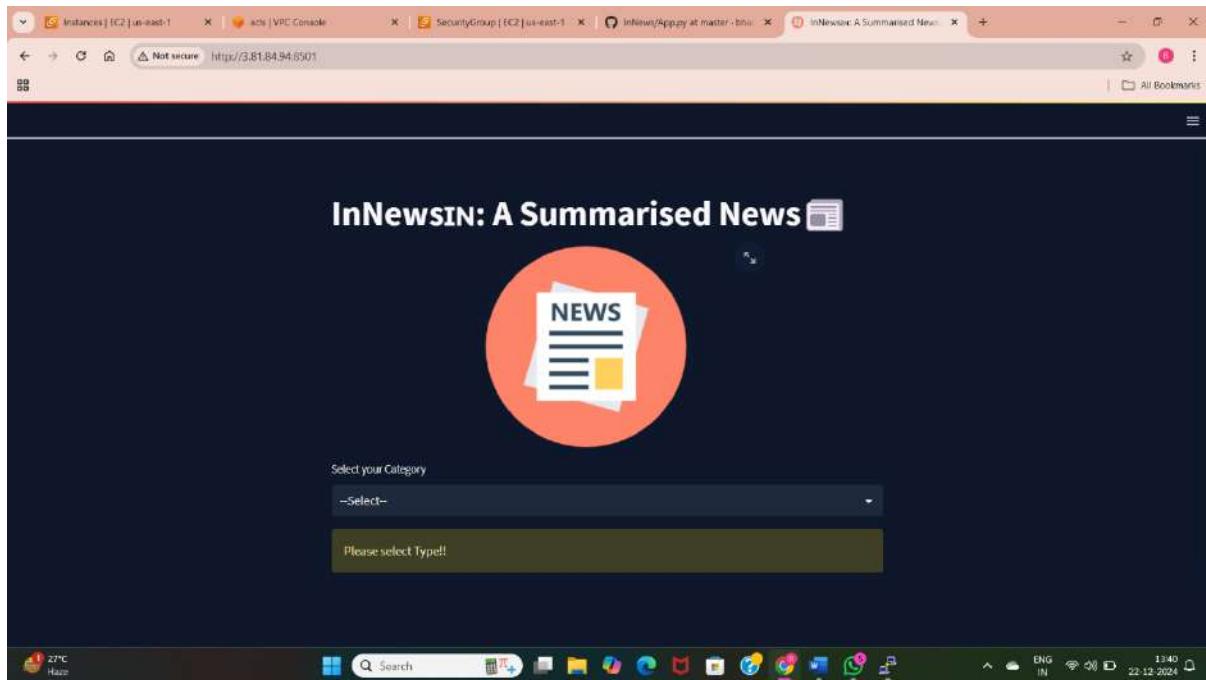
```

root@ip-10-0-0-151:~/Innews
[myenv] [root@ip-10-0-0-151 Innews] vim App.py
[myenv] [root@ip-10-0-0-151 Innews] python3 App.py
Traceback (most recent call last):
  File "App.py", line 1, in <module>
    from streamlit import App
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/_init_.py", line 10, in <module>
    from .api import build, build_article, build_text, hot, languages,
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/api.py", line 14, in <module>
    from .article import Article
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/article.py", line 15, in <module>
    from . import network
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/network.py", line 14, in <module>
    from .configuration import Configuration
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/configuration.py", line 15, in <module>
    from .logger import logger
  File "/root/myenv/lib64/python3.7/site-packages/streamlit/logger.py", line 12, in <module>
    import xml.html.clean
  File "/root/myenv/lib64/python3.7/site-packages/xml/html/clean.py", line 21, in <module>
    from .None
ImportError: xml.html.clean module is now a separate project: xml_html_clean.
Install xml_html_clean or xml_html_clean directly.
[myenv] [root@ip-10-0-0-151 Innews] pip3 install xml_html_clean
Collecting xml_html_clean
  Downloading xml_html_clean-0.4.1-py3-none-any.whl.metadata (2.4 kB)
Requirement already satisfied: xml_html_clean in /root/myenv/lib/python3.7/site-packages (from xml_html_clean) (0.4.1)
Installing collected packages: xml_html_clean
Successfully installed xml_html_clean-0.4.1
[myenv] [root@ip-10-0-0-151 Innews] python3 App.py
[myenv] [root@ip-10-0-0-151 Innews] streamlit run App.py
[myenv] [root@ip-10-0-0-151 Innews] streamlit run App.py
2024-12-22 08:05:11.543
Warning: to view this Streamlit app on a browser, run it with the following
command:
streamlit run App.py [ARGUMENTS]
[myenv] [root@ip-10-0-0-151 Innews] *C
[myenv] [root@ip-10-0-0-151 Innews] streamlit run App.py
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.
  Streamlit URL: http://10.0.0.151:8501
  Streamlit URL: http://3.81.84.94:8501

```

- ♣ Now deploy the Innews python application using “streamlit run App.py”.



- This is the innews python application. We can access this using public ip address of instance.

```
root@ip-10-0-0-151:~/Medical-Insurance
You can now view your Streamlit app in your browser.
Streamlit URL: http://10.0.0.151:8501
Streamlit URL: http://3.81.84.94:8501

[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data] Package punkt is already up-to-date!
<home>
myenv$ [root@ip-10-0-0-151 Innews]$ cd
myenv$ [root@ip-10-0-0-151 ~]$ mkdir Medical-insurance
myenv$ [root@ip-10-0-0-151 ~]$ cd Medical-insurance/
myenv$ [root@ip-10-0-0-151 Medical-insurance]$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, run:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
initialized empty Git repository in /root/Medical-insurance/.git/
myenv$ [root@ip-10-0-0-151 Medical-insurance]$ git remote add origin https://github.com/bhargavibairagi/Medical-insurance.git
remote: Enumerating objects: 1008 (k/4), done.
remote: Counting objects: 100% (494/494), done.
remote: Compressing objects: 100% (373/373), done.
remote: Total 10759 (delta 0), reused 0 (delta 0), pack-reused 10755 (from 1)
Receiving objects: 100% (10759/10759), 81.69 MiB | 34.57 MiB/s, done.
Receiving deltas: 100% (618/618), done.
From https://github.com/bhargavibairagi/Medical-Insurance
 * branch            master      -> FETCH_HEAD
 * [new branch]      master      -> origin/master
Updating files: 100% (1028/1028), done.
myenv$ [root@ip-10-0-0-151 Medical-insurance]$ ll
total 2244
-rw-r--r-- 1 root root    944 Dec 22 08:11 app.py
-rw-r--r-- 1 root root   169 Dec 22 08:11 gitattributes
drwxr-xr-x  1 root root     5 Dec 22 08:11 Medical
-rw-r--r-- 1 root root  75118 Dec 22 08:11 Medical_Insurance.ipynb
-rw-r--r-- 1 root root   1555 Dec 22 08:11 requirements.joblib
-rw-r--r-- 1 root root    21 Dec 22 08:11 profile
-rw-r--r-- 1 root root 2196012 Dec 22 08:11 regressor.pkl
-rw-r--r-- 1 root root    235 Dec 22 08:11 requirements.txt
drwxr-xr-x  2 root root    23 Dec 22 08:11 static
myenv$ [root@ip-10-0-0-151 Medical-insurance]$ pip3 install -r requirements.txt
```

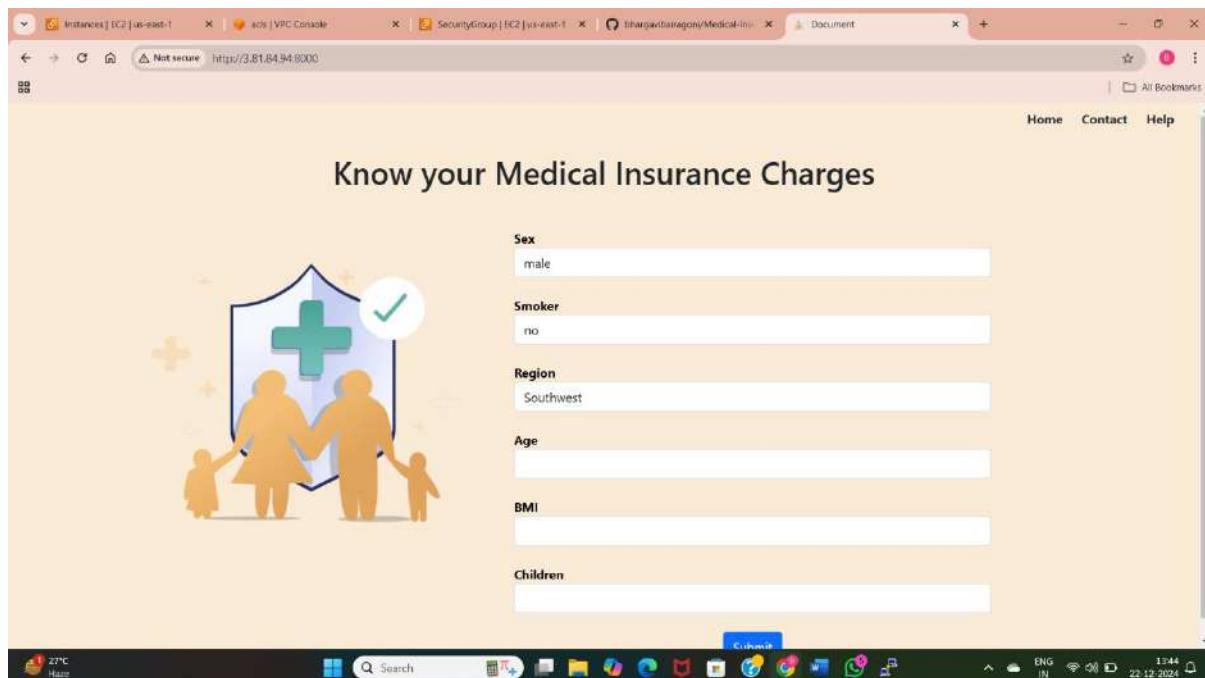
- Next I want to deploy the medical insurance python application.
- Take the medical insurance code from github.
- Install the requirements of medical insurance.
- Change the host to 0.0.0.0/0 and port number is 8000.

```

root@ip-10-0-0-151:~/Medical-insurance:
UserWarning:
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
UserWarning:
* Debugger is active!
* Debugger PIN: 101-069-954
* Copyenv: [root@ip-10-0-0-151 Medical-insurance]# vim app.py
[myenv] [root@ip-10-0-0-151 Medical-insurance]# python3 app.py
[myenv] [root@ip-10-0-0-151 Medical-insurance]# python3 app.py
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
UserWarning:
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
UserWarning:
* Debugger is active!
* Debugger PIN: 101-069-954
Press Ctrl+C to quit
Restarting with watchdog (inotify)
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
/mnt/ryenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
UserWarning:
* Debugger is active!
* Debugger PIN: 101-069-954

```

- Now deploy the medical insurance application using “python3 app.py”.



- Now we can access the medical insurance application using the public ip address of instance.

```

root@ip-10-0-0-151:~/Portfolio
UserWarning:
/cn0evn/vlib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
http://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
/cn0evn/vlib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
http://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
* debugger is active!
* Debugger PIN: 101-059-954
205.254.68.133 - - [22/Dec/2024 08:14:22] "GET / HTTP/1.1" 200 -
``(myenv) [root@ip-10-0-0-151 Medical-Insurance]# ``c
(myenv) [root@ip-10-0-0-151 Medical-Insurance]# cd
(myenv) [root@ip-10-0-0-151 -]# cd Portifolio
(myenv) [root@ip-10-0-0-151 Portifolio]#
(myenv) [root@ip-10-0-0-151 Portifolio]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The 'just-created' branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/Portifolio/.git/
(myenv) [root@ip-10-0-0-151 Portifolio]# git remote add origin https://github.com/bharavibairagomi/Portfolio.git
(myenv) [root@ip-10-0-0-151 Portifolio]# git pull origin master
remote: Enumerating objects: 1971, done.
remote: Counting objects: 100% (1971/1971), done.
remote: Compressing objects: 100% (1794/1794), done.
remote: Total 1971 (delta 147), reused 1968 (delta 147), pack-reused 0 (from 0)
Receiving objects: 100% (1971/1971) 1.74 MB | 25.63 MB/s, done.
Resolving deltas: 100% (1497/1497), done.
From https://github.com/bharavibairagomi/Portfolio
 * branch          master      -> FETCH HEAD
 * [new branch]    master      -> origin/master
(myenv) [root@ip-10-0-0-151 Portifolio]# ll
total 16
-rw-r--r-- 1 root root 364 Dec 22 08:15 app.py
-rw-r--r-- 1 root root 1656 Dec 22 08:15 gitattributes
drwxr-xr-x  4 root root  50 Dec 22 08:15 gitignore
drwxr-xr-x  2 root root  16 Dec 22 08:15 requirements
drwxr-xr-x  1 root root  202 Dec 22 08:15 requirements.txt
drwxr-xr-x  7 root root  64 Dec 22 08:15 static
drwxr-xr-x  2 root root  77 Dec 22 08:15 templates
(myenv) [root@ip-10-0-0-151 Portifolio]# 

```

- ♣ Now deploying the myportfolio python application.
- ♣ Take the code from github.
- ♣ Install the requirements.

```

root@ip-10-0-0-151:~/Portfolio
from flask import Flask, render_template, request, url_for, redirect

app = Flask(__name__)

@app.route("/")
def index():
    return render_template("index.html")

@app.route('/submit_form', methods=['GET', 'POST'])
def submit():
    return render_template("index.html")

if __name__ == "__main__":
    app.run(debug=True)
    app.run(host = '0.0.0.0', port =5001)
~ 

```

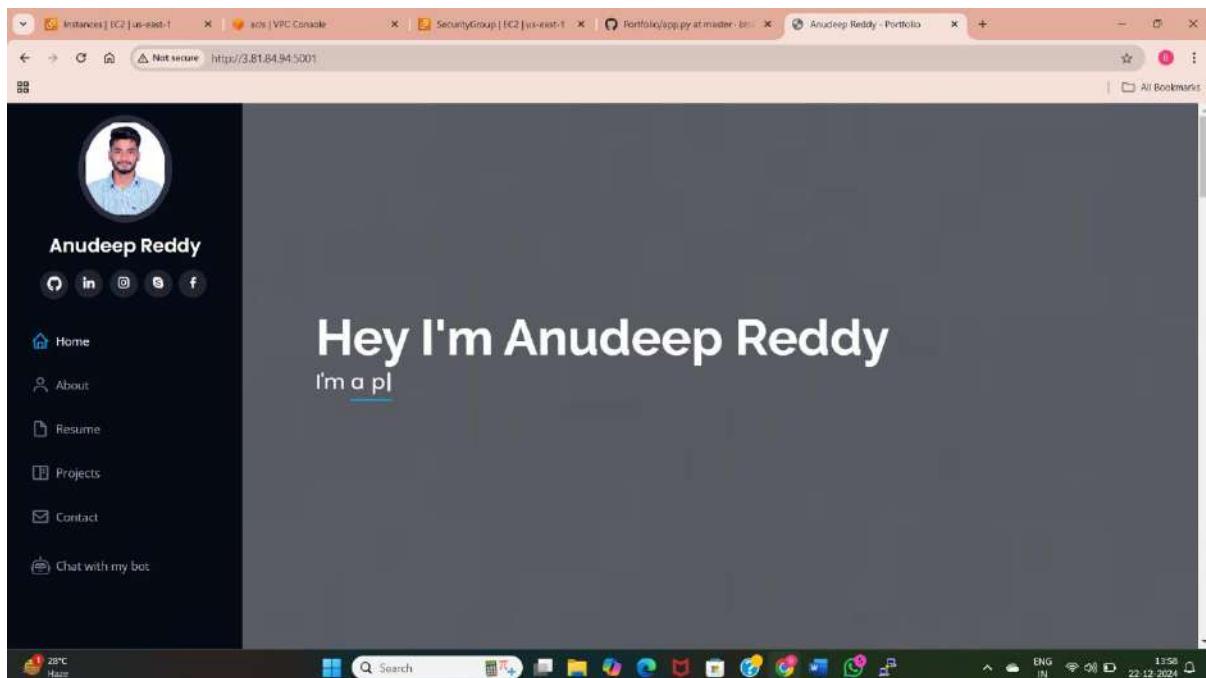
- ♣ Change the host to 0.0.0.0/0 in the app.py file.

```

root@ip-10-0-0-151:~/Portfolio
  from dataclasses import dataclass
  file "/usr/lib64/python3.7/dataclasses.py", line 155, in <module>
    class FrozenInstanceError(AttributeError): pass
KeyboardInterrupt
[myenv] root@ip-10-0-0-151:~/Portfolio# python3 app.py
  * Serving Flask app "app"
  * Debugging is active!
  * Debugger PIN: 101-059-854
  * Serving Flask app "app"
  * Debugging is active!
  * Debugger PIN: 101-059-854
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
  * Restarting with watchdog (inotify)
  * Debugger is active!
  * Debugger PIN: 101-059-854
  * Restarting with watchdog (inotify)
  * Debugger is active!
  * Debugger PIN: 101-059-854
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://10.0.0.151:5001
Press CTRL+C to quit
  * Restarting with watchdog (inotify)
  * Debugger is active!
  * Debugger PIN: 101-059-854
2024-12-22T08:27:40.000Z [22/dec/2024 08:27:40] "GET / HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:40] "GET /static/css/style.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:40] "GET /static/vendor/eas.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/bootstrap-icons/bootstrap-icons.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/bootstrap/css/bootstrap.min.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/boxicons/css/boxicons.min.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/lightbox/css/lightbox.min.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/swiper/swiper-bundle.min.css HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/purecounter/purecounter-vanilla.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/typedjs/typed.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/waypoints/waypoints.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/vendor/waypoints/waypoints-animate.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:41] "GET /static/img/herc-bug.jpg HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:42] "GET /static/img/herc-bug.jpg HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:42] "GET /static/vendor/lightbox/js/lightbox.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:42] "GET /static/vendor/iscroll/iscroll.layout/iscroll.paged.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:42] "GET /static/vendor/swiper/swiper-bundle.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:43] "GET /static/vendor/typed.js/typed.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:43] "GET /static/vendor/waypoints/waypoints.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:43] "GET /static/vendor/waypoints/waypoints-animate.min.js HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:44] "GET /static/vendor/boxicons/fonts/boxicons.woff2 HTTP/1.1" 200 -
205.254.166.133 - - [22/dec/2024 08:27:44] "GET /favicon.ico HTTP/1.1" 404 -

```

- ♣ Deploying the application manually using “python3 app.py”.



- ♣ This is the myportfolio application deployed manually. To access this we need to use public IP address of instance with application port.

```

root@ip-10-0-0-151:~/Agri
205.254.168.133 -- [22/Dec/2024 08:27:43] "GET /static/vendor/rhp-email-form/validate.js HTTP/1.1" 200 -
205.254.168.133 -- [22/Dec/2024 08:27:43] "GET /static/js/main.js HTTP/1.1" 200 -
205.254.168.133 -- [22/Dec/2024 08:27:43] "GET /static/vendor/boxicons/fonts/boxicons.woff2 HTTP/1.1" 200 -
205.254.168.133 -- [22/Dec/2024 08:27:44] "GET /favicon.ico HTTP/1.1" 404 -
*c * Debugger is active!
* debugger pid=10-0-0-151-854
(myenv) [root@ip-10-0-0-151 Portfilo]# vim app.py
(myenv) [root@ip-10-0-0-151 Portfilo]# cd
(myenv) [root@ip-10-0-0-151 ~]# mkdir Agri
(myenv) [root@ip-10-0-0-151 ~]# cd Agri/
(myenv) [root@ip-10-0-0-151 Agri]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:     git config --global init.defaultBranch <name>
hint:
hint: names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:     git branch -m <name>
initialized empty Git repository in /root/Agri/.git/
(myenv) [root@ip-10-0-0-151 Agri]# git remote add origin https://github.com/bhargavbairagi/Agri.git
(myenv) [root@ip-10-0-0-151 Agri]# git remote add origin master
error: Failed to add the remote 'origin'
(myenv) [root@ip-10-0-0-151 Agri]# git pull origin master
remote: Enumerating objects: 10765, done.
remote: Counting objects: 100% (10765/10765), done.
remote: Compressing objects: 100% (7779/7779), done.
remote: Total 10765 (delta 3), reused 6 (delta 2), pack-reused 10755 (from 1)
Receiving objects: 100% (10765/10765), 81.97 MiB / 23.02 MiB/s, done.
Resolving deltas: 100% (813/813), done.
From https://github.com/bhargavbairagi/Agri
 * branch            master       -> FETCH_HEAD
 * branch            master      -> origin/master
Updating 100% (10765/10765), done.
(myenv) [root@ip-10-0-0-151 Agri]# ls
total 1604
drwxr-xr-x 4 root root    50 Dec 22 08:30 Agri
-rw-r--r-- 1 root root 147833 Dec 22 08:30 Agriculture.csv
-rw-r--r-- 1 root root   1092 Dec 22 08:30 AgricultureDeploy - shortcut.lnk
-rw-r--r-- 1 root root 147712 Dec 22 08:30 Agriculture.ipynb
-rw-r--r-- 1 root root    70 Dec 22 08:30 app.py
-rw-r--r-- 1 root root 17988 Dec 22 08:30 decmodel.pkl
-rw-r--r-- 1 root root 16905 Dec 22 08:30 requirements.txt
-rw-r--r-- 1 root root   23 Dec 22 08:30 Frontfile
-rw-r--r-- 1 root root  325 Dec 22 08:30 requirements.txt
drwxr-xr-x 2 root root   20 Dec 22 08:30 Templates
(myenv) [root@ip-10-0-0-151 Agri]#

```

- ♣ Now deploying the Agri python application.
- ♣ Take the code from github.
- ♣ Install the requirements.

```

from flask import Flask, render_template, url_for, request, redirect
import numpy as np
import pandas as pd
import joblib
import pickle

app=Flask(__name__)

model=joblib.load('decmodel.pkl')

@app.route('/')
@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/predict',methods=['post'])
def predict():
    int_features=[[x for x in request.form.values()]]
    c=['N','P','K','temperature','humidity','ph','rainfall']

    fin=pd.DataFrame(int_features,columns=c)
    result=model.predict(fin)
    return render_template("main.html",prediction_text=" The Recommended Crop is : {}".format(result))

if __name__ == "__main__":
    app.debug=True
    app.run(host = '0.0.0.0',port= 7000)
~

```

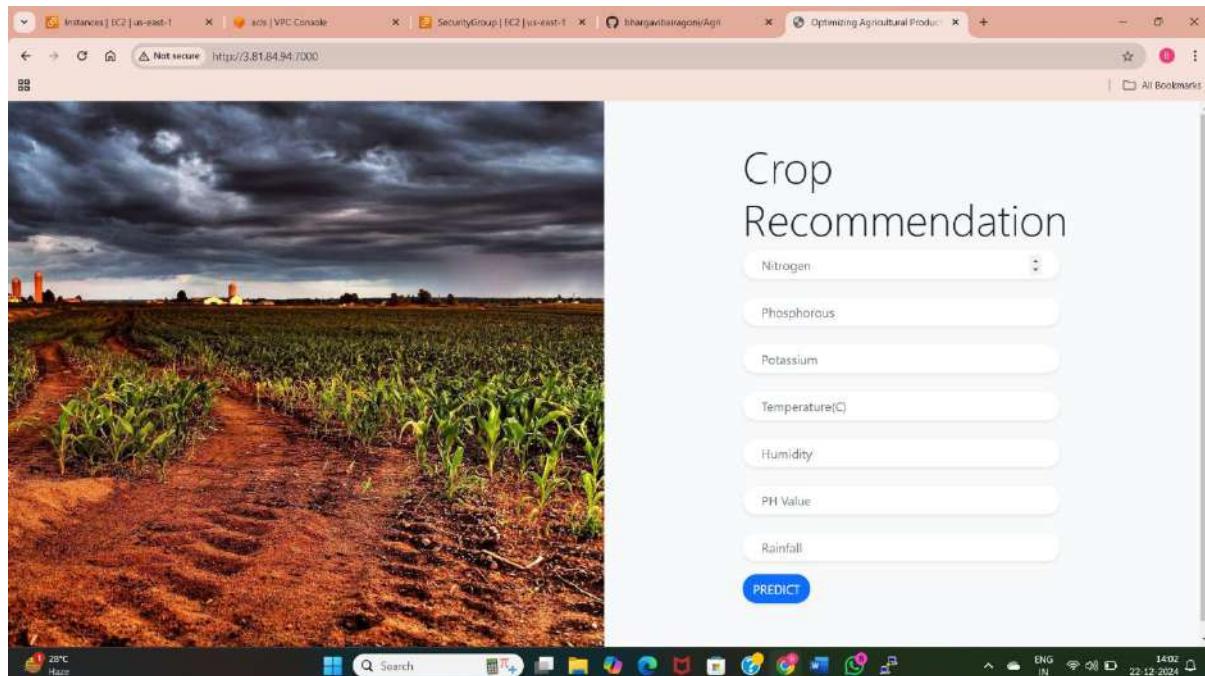
- ♣ Change the host to 0.0.0.0 in the app.py file.

```

root@ip-10-0-0-151:~/Agri
root@ip-10-0-0-151:~/Agri# pip3 install numpy==1.20.3
Requirement already satisfied: numpy==1.20.3 in /root/venv/lib/python3.7/site-packages (1.20.3)
root@ip-10-0-0-151:~/Agri# python3 app.py
/root/venv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
  * Serving Flask app 'app'
  * Running on http://127.0.0.1:7000
  Press CTRL+C to quit
  * Restarting with watchdog (inotify)
/root/venv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
  * Serving Flask app 'app'
  * Debug mode on
  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://127.0.0.1:7000
  * Running on https://10.0.0.151:7000
  Press CTRL+C to quit
  * Restarting with watchdog (inotify)
/root/venv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
  * Debug mode is active!
  * Debugger PIN: 101-049-954
205.254.166.153 - - [22/Dec/2024 08:32:06] "GET / HTTP/1.1" 200 -
205.254.166.153 - - [22/Dec/2024 08:32:06] "GET /static/styles.css HTTP/1.1" 404 -

```

- Deploying the application manually using “python3 app.py”.



- This is the Agri application deployed manually. To access this we need to use public IP address of instance with application port.

```

root@ip-10-0-0-151:~/India-liver-patients
* Running on http://10.0.0.151:7000
Press Ctrl+C to quit
* Restarting with watchdog (inotify)
/rOOT/ENV/lib64/python3.7/site-packages/scikit-learn/base.py:338: UserWarning: Trying to unpickle estimator DecisionTreeClassifier from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  warnings.warn(_UserWarning)
* Dockerizer is active!
* Dockerizer PID: 101-0-0-974
205.156.133 - - [22/Dec/2024 08:32:06] "GET / HTTP/1.1" 200 -
205.156.133 - - [22/Dec/2024 08:32:06] "GET /static/styles.css HTTP/1.1" 404 -
* Configuration [root@ip-10-0-0-151 ~]# cd India-liver-patients
[myenv] [root@ip-10-0-0-151 ~]# cd India-liver-patients
[myenv] [root@ip-10-0-0-151 India-liver-patients]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change, so to configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call
hint:
hint: git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint: git branch -m <name>
Initialized empty Git repository in /root/India-liver-patients/.git/
[myenv] [root@ip-10-0-0-151 India-liver-patients]# git remote add origin https://github.com/bhargavbairagoni/indian-liver-patients.git
[myenv] [root@ip-10-0-0-151 India-liver-patients]# git pull origin master
remote: Enumerating objects: 10757, done.
remote: Total 10757 (delta 0), reused 0 (delta 0), pack-reused 10757 (from 1)
Receiving objects: 100% (10757/10757), 83.42 MiB | 24.37 MiB/s, done.
Resolving deltas: 100% (815/815), done.
From https://github.com/bhargavbairagoni/indian-liver-patients
 * branch          master    -> FETCH_HEAD
 * branch          master    -> origin/master
Updating files: 100% (10757/10757), 83.42 MiB | 24.37 MiB/s, done.
[myenv] [root@ip-10-0-0-151 India-liver-patients]# ll
total 4200
-rw-r--r-- 1 root root 1076 Dec 22 08:33 app.py
-rw-r--r-- 1 root root 1696 Dec 22 08:33 gitattributes
-rw-r--r-- 1 root root 23230 Dec 22 08:33 Indian Liver Patient.csv
-rw-r--r-- 1 root root 629715 Dec 22 08:33 indian_liver_patient.ipynb
drwxr-xr-x  4 root root 50 Dec 22 08:33 liver
-rw-r--r-- 1 root root 1280 Dec 22 08:33 logmodel.pkl
-rw-r--r-- 1 root root 974 Dec 22 08:33 ohe.joblib
-rw-r--r-- 1 root root 23 Dec 22 08:33 Predict
-rw-r--r-- 1 root root 325 Dec 22 08:33 requirements.txt
drwxr-xr-x  2 root root 20 Dec 22 08:33 templates
[myenv] [root@ip-10-0-0-151 India-liver-patients]# pip3 install -r requirements.txt

```

- ♣ Now deploying the Indian-liver python application.
- ♣ Take the code from github.
- ♣ Install the requirements.

```

root@ip-10-0-0-151:~/India-liver-patients
from flask import Flask, render_template, url_for, request, redirect
import numpy as np
import pandas as pd
import joblib
import pickle

app = Flask(__name__)

model = joblib.load('logmodel.pkl')
onehot = joblib.load('ohe.joblib')

@app.route('/')
@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/predict', methods=['POST'])
def predict():
    int_features = [[x for x in request.form.values()]]
    print("$*30")
    print(int_features)
    c = ["Gender", "Age", "TB", "DB", "Alkphos", "Sgpt", "Sgot", "TP", "ALB", "AG"]
    df = pd.DataFrame(int_features, columns=c)
    l = onehot.transform(df.iloc[:, :1])
    c = onehot.get_feature_names_out()
    t = pd.DataFrame(l, columns=c)
    l2 = df.iloc[:, 1:]
    final = pd.concat([l2, t], axis=1)
    result = model.predict(final)
    if result == 0:
        pumba = "he/she is a liver patient"
    else:
        pumba = "he/she is not a liver patient"

    print(int_features)

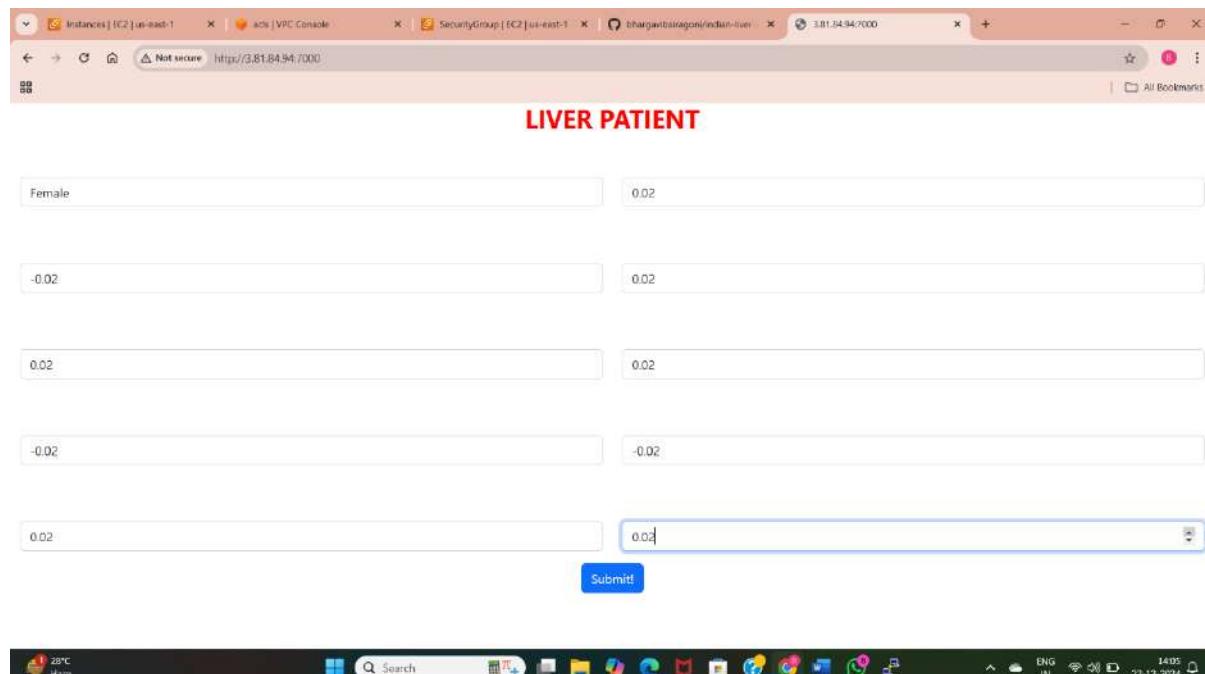
    return render_template("main.html", prediction_text="The predicted Lab-Report is : {}".format(pumba))

if __name__ == "__main__":
    app.debug=True
    app.run(host = '0.0.0.0', port=7000)

```

- ♣ Change the host to 0.0.0.0 in the app.py file.

- ♣ Deploying the application manually using “`python3 app.py`”.



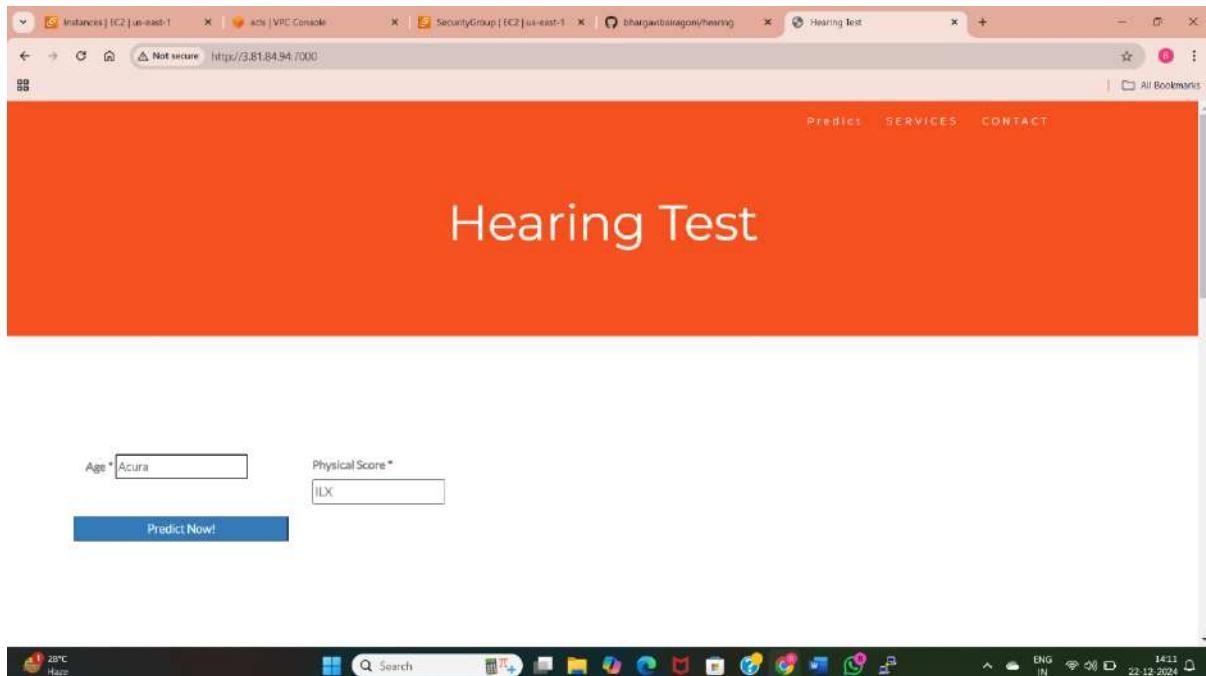
- ♣ This is the liver application deployed manually. To access this we need to use public IP address of instance with application port.

```

root@ip-10-0-0-151:~/Hearing
UserWarning:
  * Debugger is active!
  * Debugger PIN: 101-049-854
205.254.166.133 - - [22/Dec/2024 08:39:02] "GET /0_debugger?&cmd=resource&f=console.png HTTP/1.1" 204 -
205.254.166.133 - - [22/Dec/2024 08:39:02] "GET /0_debugger?&cmd=resource&f=console.png HTTP/1.1" 304 -
<ipython> [root@ip-10-0-0-151 Hearing]# ll
total 352
drwxr-xr-x 4 root root 50 Dec 22 08:37 files/
-rw-r--r-- 1 root root 1696 Dec 22 08:37 gitattributes
-rw-r--r-- 1 root root 213384 Dec 22 08:37 Logistic_Regression.ipynb
-rw-r--r-- 1 root root 797 Dec 22 08:37 model.pkl
-rw-r--r-- 1 root root 21 Dec 22 08:37 Procfile
-rw-r--r-- 1 root root 285 Dec 22 08:37 requirements.txt
-rw-r--r-- 1 root root 905 Dec 22 08:37 scaler.joblib
drwxr-xr-x 2 root root 23 Dec 22 08:37 Templates
<ipython> [root@ip-10-0-0-151 Hearing]# cd app/
<ipython> [root@ip-10-0-0-151 Hearing]# python app.py
<ipython> [root@ip-10-0-0-151 Hearing]# mv Templates templates
<ipython> [root@ip-10-0-0-151 Hearing]# python app.py
/mnt/nyenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator LogisticRegression from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
/mnt/nyenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator StandardScaler from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
  * serving flask app 'app'
  * Debug mode: on
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://127.0.0.1:7000
  * Running on https://10.0.0.1:517000
  * Press Ctrl+C to quit
  Restarting with watchdog (inotify)
/mnt/nyenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator LogisticRegression from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
/root/nyenv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: Trying to unpickle estimator StandardScaler from version 1.1.2 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
  UserWarning,
  * Debug mode: on
  * Debugger is active!
  * Debugger PIN: 101-049-854
205.254.166.133 - - [22/Dec/2024 08:40:36] "GET / HTTP/1.1" 200 -
205.254.166.133 - - [22/Dec/2024 08:40:36] "GET /images/map.jpg HTTP/1.1" 404 -

```

- Deploying the application manually using “python3 app.py”.



- This is the Hearing test application deployed manually. To access this we need to use public IP address of instance with application port.

```

root@ip-10-0-0-151:~/Fuel-consumption-rating
rekcing code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning,
/root/venv/lib64/python3.7/site-packages/sklearn/base.py:338: UserWarning: trying to unpickle estimator StandardScaler from version 1.1.2 when using version 1.0.2. This might lead to break
ing code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
StandardScaler()
  * composer is active!
  * Deployer PIN: 101-06-874
205.254.166.133 - - [22/Dec/2024 08:40:36] "GET / HTTP/1.1" 200 -
205.254.166.133 - - [22/Dec/2024 08:40:36] "GET /w3images/fmp.jpg HTTP/1.1" 404 -
"CentOS-8.7 (Core) [root@ip-10-0-0-151 ~]" cd
(myenv) [root@ip-10-0-0-151 ~] mkdir Fuel-consumption-rating
(myenv) [root@ip-10-0-0-151 ~] cd Fuel-consumption-rating/
(myenv) [root@ip-10-0-0-151 Fuel-consumption-rating]# git init
hint: Using 'master' as the name for the initial branch. You can set the default branch name
in config if you're using this command.
hint: You can change the branch name via command 'git branch --set-upstream'.
hint: Using 'main' as the name for the initial branch. You can set the default branch name
in config if you're using this command.
hint: You can change the branch name via command 'git branch --set-upstream'.
hint: git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command.
hint: 
hint: git branch -n chance
initialized empty Git repository in /root/Fuel-consumption-rating/.git/
(myenv) [root@ip-10-0-0-151 Fuel-consumption-rating]# git remote add origin https://github.com/bharatvibairagani/Fuel-Consumption-Rating.git
(myenv) [root@ip-10-0-0-151 Fuel-consumption-rating]# git pull origin master
remote: Enumerating objects: 10757 (delta 0), reused 0 (delta 0), pack-reused 10757 (from 1)
Receiving objects: 100% (10757/10757), 81.49 MiB | 23.05 MiB/s, done.
Resolving deltas: 100% (808/808), done.
From https://github.com/bharatvibairagani/Fuel-Consumption-Rating
 * branch          master    -> FETCH_HEAD
 * branch          master    -> origin/master
Updating file: 100% (10757/10757), done.
(myenv) [root@ip-10-0-0-151 Fuel-consumption-rating]# ll
total 1060
-rw-r--r-- 1 root root 1020 Dec 22 08:42 app.py
-rw-r--r-- 1 root root 2030 Dec 22 08:42 fmp.joblib
drwxr-xr-x 4 root root 56 Dec 22 08:42 joblib
-rw-r--r-- 1 root root 130931 Dec 22 08:42 fuel_consumption.ipynb
-rw-r--r-- 1 root root 3696 Dec 22 08:42 gitattributes
-rw-r--r-- 1 root root 7208 Dec 22 08:42 myenv_Fuel_consumption_Ratings.csv
-rw-r--r-- 1 root root 221 Dec 22 08:42 myenv_Fuel_consumption_Ratings.csv
-rw-r--r-- 1 root root 325 Dec 22 08:42 requirements.txt
-rw-r--r-- 1 root root 2891096 Dec 22 08:42 RFR.pkl
drwxr-xr-x 2 root root 20 Dec 22 08:42 templates
(myenv) [root@ip-10-0-0-151 Fuel-consumption-rating]# pip3 install -r requirements.txt

```

- ♣ Now deploying the duel-consumption python application.
- ♣ Take the code from github.
- ♣ Install the requirements.

```

root@ip-10-0-0-151:~/Fuel-consumption-rating
from flask import Flask, render_template, url_for, request, redirect
import numpy as np
import pandas as pd
import joblib
import pickle

app = Flask(__name__)

model = joblib.load('RFR.pkl')
onehot = joblib.load('five_joblib')

@app.route('/')
@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/predict', methods=['POST'])
def predict():
    int_features = [[x for x in request.form.values()]]
    c = ["make", "model", "vehicle_class", "transmission", "fuel", "engine", "cylinder", "co2", "co2Rating", "smokerating"]
    df = pd.DataFrame(int_features, columns=c)
    l = onehot.transform(df.iloc[:, 5:])
    c = onehot.get_feature_names_out()
    t = pd.DataFrame(l, columns=c)
    l2 = df.iloc[:, 5:]
    final = pd.concat([l2, t], axis=1)
    result = model.predict(final)
    print("The Result is : ", result)

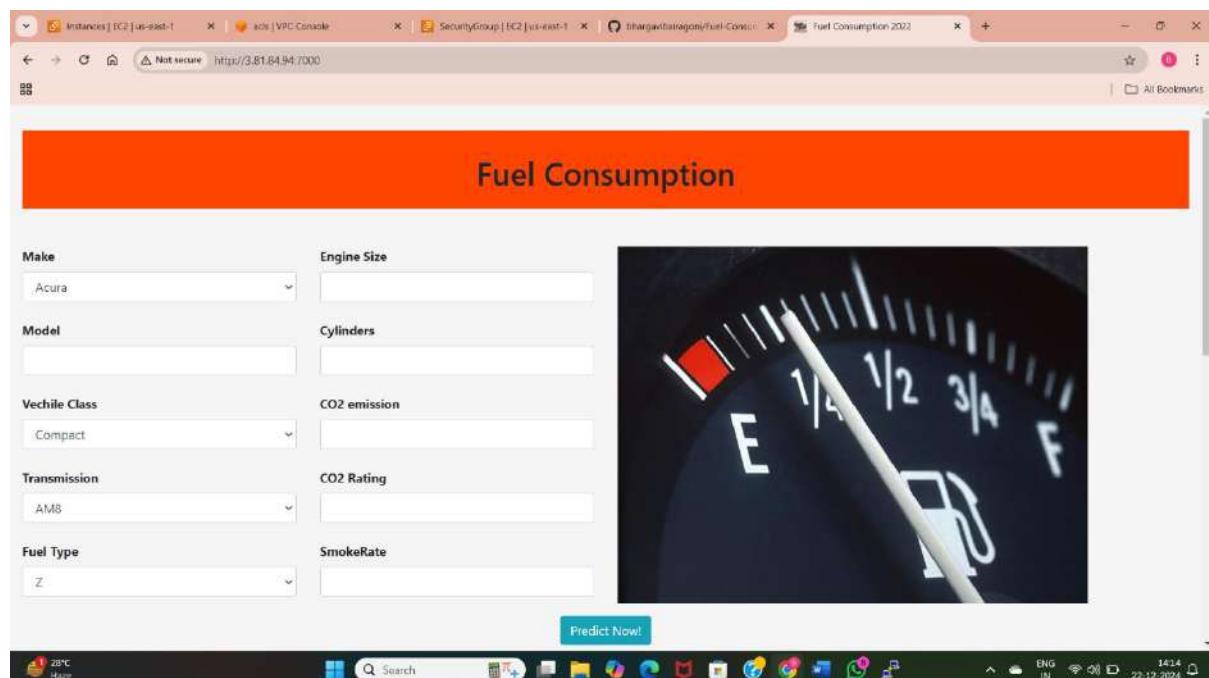
    print(int_features)

    return render_template("main.html", prediction_text=" The Estimated Fuel consumption Rating is: {}".format(result))

if __name__ == "__main__":
    app.debug=True
    app.run(host = '0.0.0.0', port = 7000)
~
```

- ♣ Change the host to 0.0.0.0 in the app.py file.

- ♣ Deploying the application manually using “`python3 app.py`”.



- ♣ This is the fuel consumption application deployed manually. To access this we need to use public IP address of instance with application port.

```

root@ip-10-0-0-151:~/USA-housing
from flask import Flask,render_template,url_for,request,redirect
import numpy as np
import pandas as pd
import joblib
import pickle

app = Flask(__name__)

model = joblib.load('regressor.pkl')

@app.route('/')
@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/predict',methods=['POST'])
def predict():
    int_features = [[x for x in request.form.values()]]
    print(int_features)
    c = ['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms', 'Avg. Area Number of Bedrooms', 'Area Population']
    final = pd.DataFrame(int_features,columns=c)
    result = model.predict(final)
    print("The result is : ",result)

    return render_template("main.html",prediction_text=" Expected Price {}".format(result))

if __name__ == "__main__":
    app.debug=True
    app.run(host = '0.0.0.0', port= 5000)

```

- Change the host to 0.0.0.0/0 in the app.py file.

```

root@ip-10-0-0-151:~/USA-housing
UserWarning:
/home/ryans/lib64/python3.7/site-packages/sklearn/base.py:336: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
http://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
/home/ryans/lib64/python3.7/site-packages/sklearn/base.py:339: UserWarning: trying to unpickle estimator OneHotEncoder from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
http://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
UserWarning:
* debugger is active!
* [DRAFT] PIDs: 101-109-154
300.25.1.66-151 ~ [23/Dec/2024 08:44:11] "GET / HTTP/1.1" 200 -
<myenv> [root@ip-10-0-0-151 ~]# mkdir USA-housing
<myenv> [root@ip-10-0-0-151 ~]# cd USA-housing
<myenv> [root@ip-10-0-0-151 USA-housing]# git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
  hint:
  hint:   git config --global init.defaultBranch <name>
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /root/USA-housing/.git/
<myenv> [root@ip-10-0-0-151 USA-housing]# git remote add origin https://github.com/bharavibairagi/USA-Housing.git
<myenv> [root@ip-10-0-0-151 USA-housing]# git pull origin master
remote: Enumerating objects: 10754 (delta 0), done.
remote: Total 10754 (delta 0), reused 0 (delta 0), pack-reused 10754 (from 3)
Receiving objects: 100% (10754/10754), 81.02 Min | 25.08 MiB/s, done.
Received deltas: 100% (10754/10754), 81.02 Min | 25.08 MiB/s, done.
From https://github.com/bharavibairagi/USA-Housing
 * branch            master      -> FETCH_HEAD
 * [new branch]        master      -> origin/master
Updating files: 100% (10754/10754), done.
<myenv> [root@ip-10-0-0-151 USA-housing]# ll
total 325
drwxr-xr-x 1 root root  325 Dec 22 08:45 app.py
-rw-r--r-- 1 root root 1698 Dec 22 08:45 gitattributes
drwxr-xr-x 4 root root  50 Dec 22 08:45 price
-rw-r--r-- 1 root root  21 Dec 22 08:45 profile
-rw-r--r-- 1 root root 1013 Dec 22 08:45 regressor.pkl
-rw-r--r-- 1 root root  325 Dec 22 08:45 requirements.txt
drwxr-xr-x 2 root root  23 Dec 22 08:45 templates
-rw-r--r-- 1 root root 28695 Dec 22 08:45 USA_House.ipynb
<myenv> [root@ip-10-0-0-151 USA-housing]# vim app.py
<myenv> [root@ip-10-0-0-151 USA-housing]# pip3 install -r requirements.txt

```

- Now deploying the USA-Housing python application.
- Take the code from github.
- Install the requirements.

```
root@ip-10-0-0-151:~/Myfuel

from flask import Flask,render_template,url_for,request,redirect
import numpy as np
import pandas as pd
import joblib
import pickle

app = Flask(__name__)

model = joblib.load('regressor.pkl')
onehot = joblib.load('OneHotee.joblib')

@app.route('/')
@app.route('/main')
def main():
    return render_template('main.html')

@app.route('/predict',methods=['POST'])
def predict():
    int_features = [[x for x in request.form.values()]]
    print(int_features)
    c = ["make","model","vehicle_class","transmission","fuel","engine","cylinder","co2","smokerating"]
    df = pd.DataFrame(int_features,columns=c)
    l = onehot.transform(df.iloc[:,5:])
    c = onehot.get_feature_names_out()
    t = pd.DataFrame(l,columns=c)
    l2 = df.iloc[:,5:]
    final =pd.concat([l2,t],axis=1)
    result = model.predict(final)
    print("The Result is :",result)

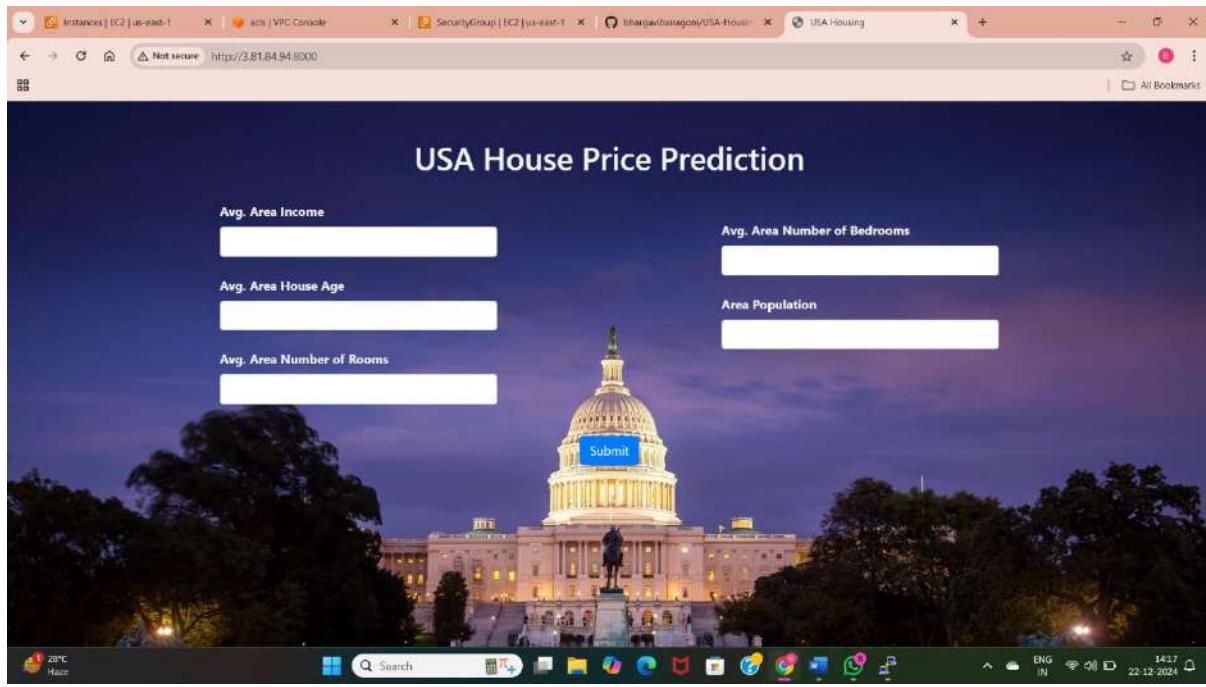
    print(int_features)

    return render_template("main.html",prediction_text="Your Vehicle Fuel Consumption is : {}".format(result))

if __name__ == "__main__":
    app.debug=True
    app.run(host = '0.0.0.0', port =8000)
~
```

- ♣ Change the host to 0.0.0.0/0 in the app.py file.

- ♣ Deploying the application manually using “`python3 app.py`”.

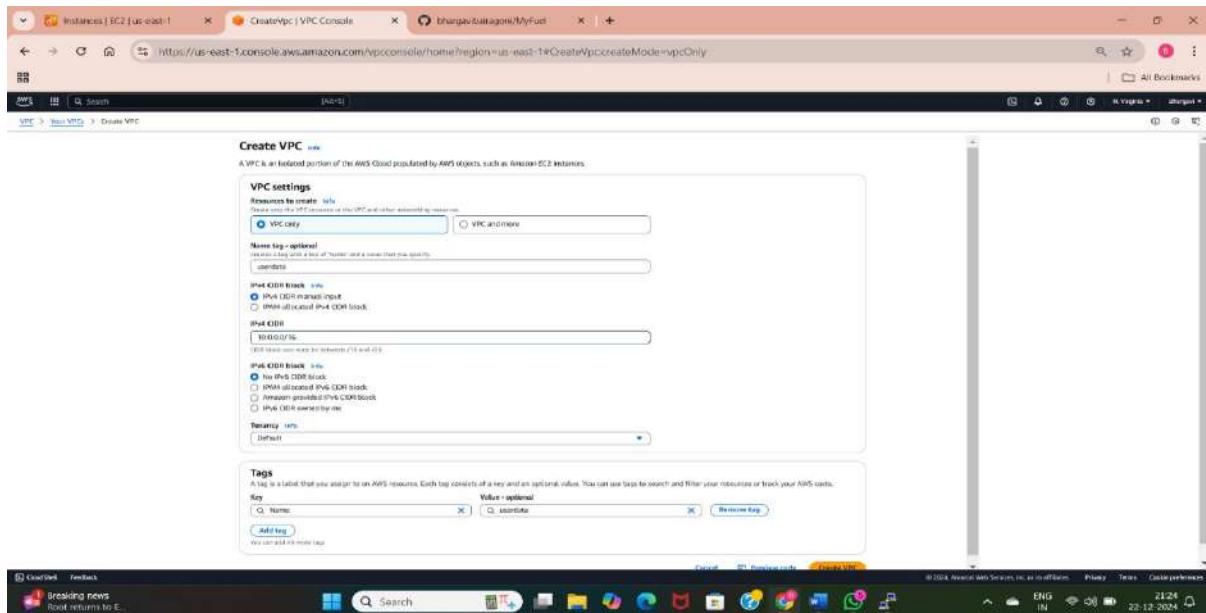


- This is the USA House price prediction application deployed manually. To access this we need to use public IP address of instance with application port.

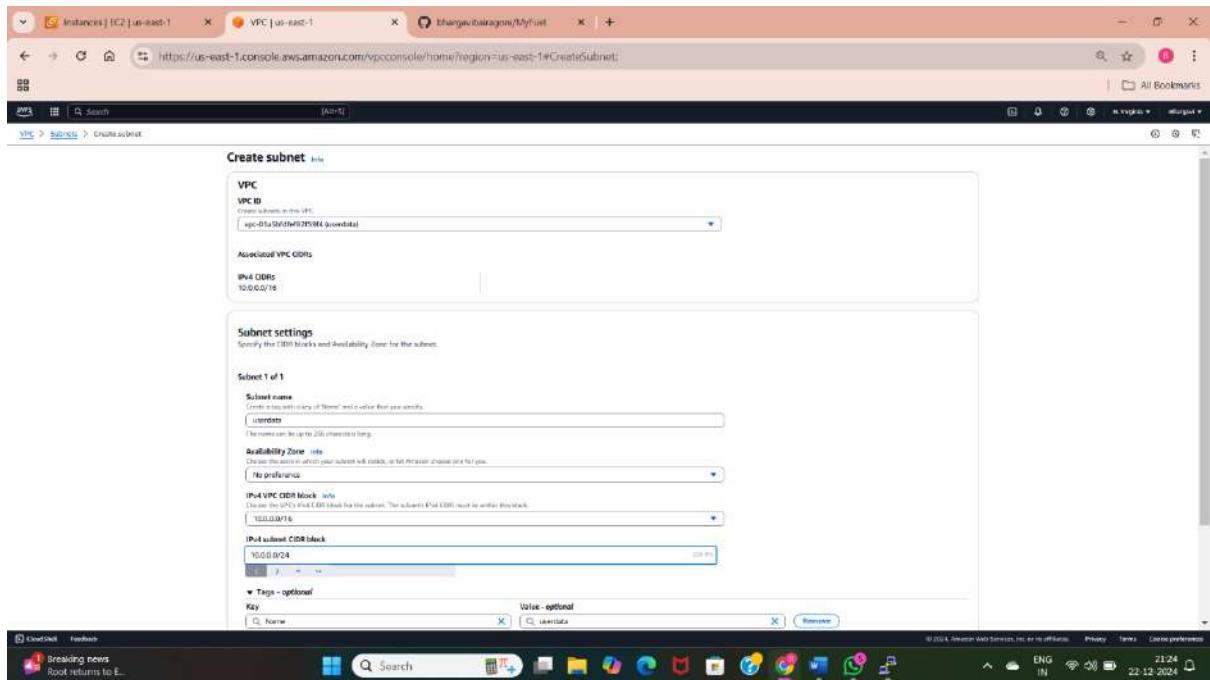
## METHOD-2: BUILD AND DEPLOY PYTHON APPLICATIONS ALONG WITH EC2 INSTANCE(USERDATA)

Userdata is a script or set of commands that you can provide when launching a virtual machine (VM) or EC2 instance. This script is executed automatically when the instance starts for the first time. It's used to automate instance configuration without manual intervention.

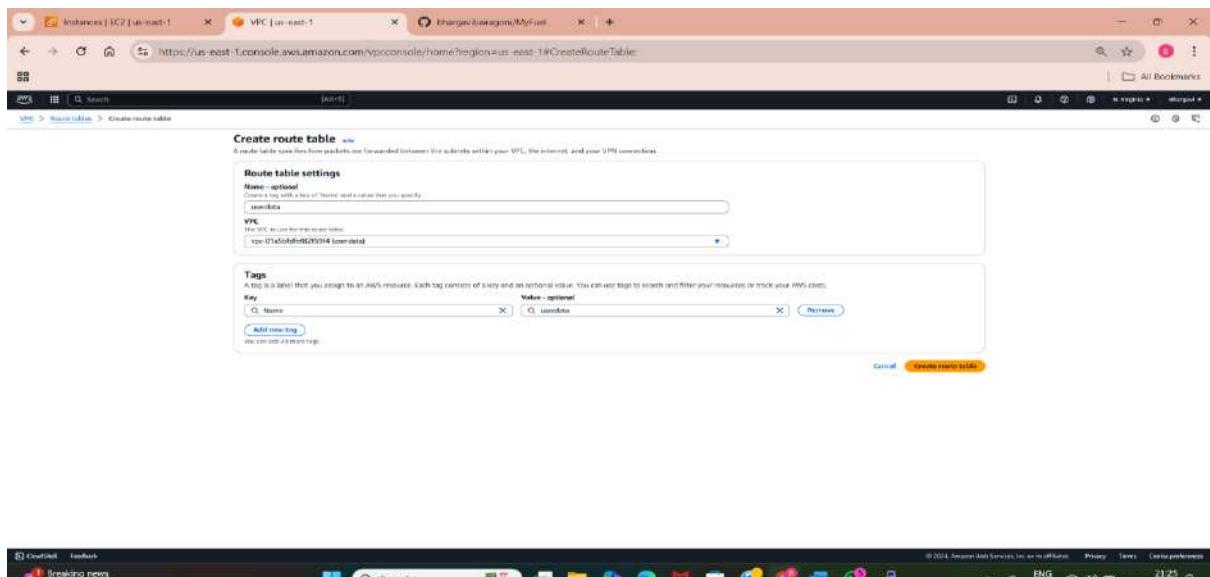
- To build and deploy python applications along with EC2 instance (userdata) first we need to create a Virtual Private Cloud along with subnets, route table, internet gateway.



- To create VPC select “vpc only”, give the name for vpc, provide VPC CIDR. The CIDR (Classless Inter-Domain Routing) block defines the range of IP addresses that the VPC can use.



- Now create a subnet by selecting the above create VPC, give the name for subnet and specify the subnet CIDR block.



- Now create a route table by selecting the VPC which is created above and give the name for route table.

The screenshot shows the AWS VPC Internet Gateways console. It lists two internet gateways:

- igw-0cd52d2d949ce5ab / my-igw-python**: Status: Available, VPC ID: vpc-01e50ffef82759f4.
- igw-0spayb0**: Status: Available, VPC ID: vpc-01e50ffef82759f4.

A context menu is open over the second gateway, with the option "Attach to VPC" highlighted.

The dialog box is titled "Attach to VPC (igw-0e0ce1a515f64d6dc)". It contains the following information:

- VPC**: Select an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.
- Available VPCs**: A dropdown menu showing "vpc-01e50ffef82759f4".
- AWS Command Line Interface command**: A text input field containing the command: `aws vpc attach-internet-gateway --internet-gateway-id igw-0e0ce1a515f64d6dc --vpc-id vpc-01e50ffef82759f4`.
- Cancel** and **Attach internet gateway** buttons.

The screenshot shows the AWS VPC dashboard. It lists one VPC named "userdata".

Name	VPC ID	Status	BLOCKED	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table	Main network ACL
userdata	vpc-01e50ffef82759f4	Available	Off	10.0.0.0/16	-	dopt-0fa14cc7d400-tn9c2	rtb-0ba70d7c7f6670a9b	acl-018ca4007a597767

- After create a internet gateway and attach it to the vpc. First select the created Internet gateway and click on actions and select attach vpc, and select the above created vpc.

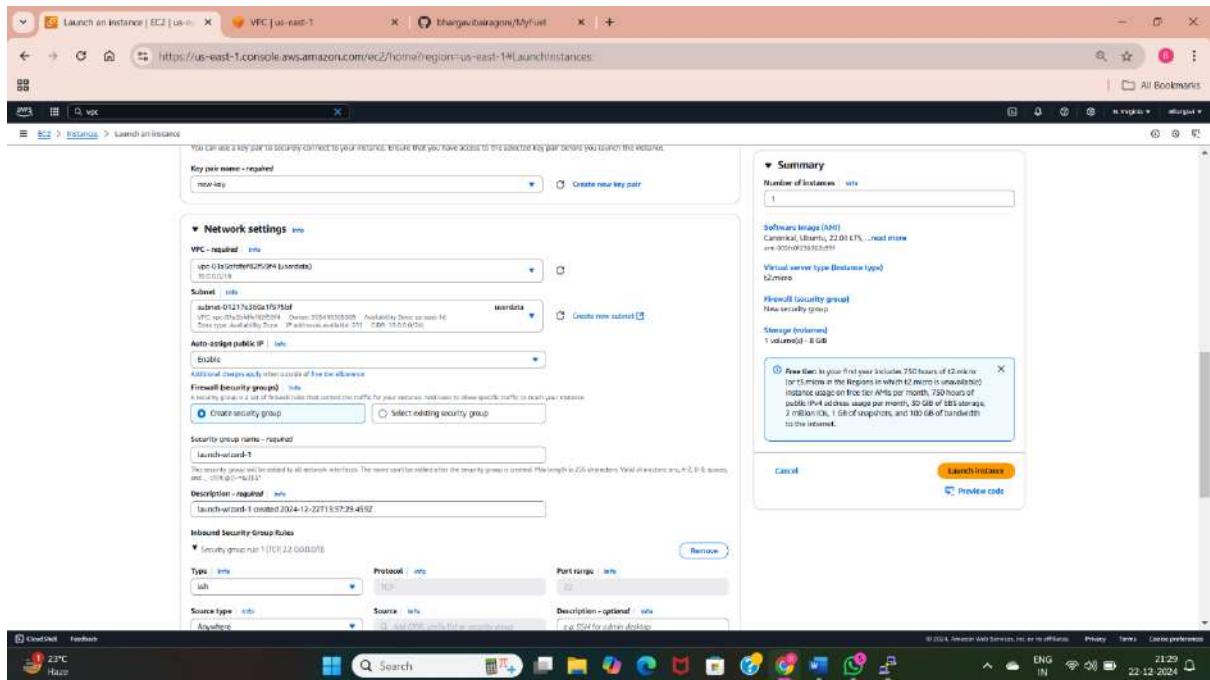
The screenshot shows the AWS VPC Resource Map for the "userdata" VPC. It displays the following components:

- VPC**: Shows details of the VPC.
- Subnets (1)**: Subnet: "userdata" (us-east-1d).
- Route tables (2)**: Route tables: "rtb-00b7a4b503c5ca09" and "rtb-00b7a4b505541109".
- Network connections (1)**: A connection labeled "userdata" is shown.

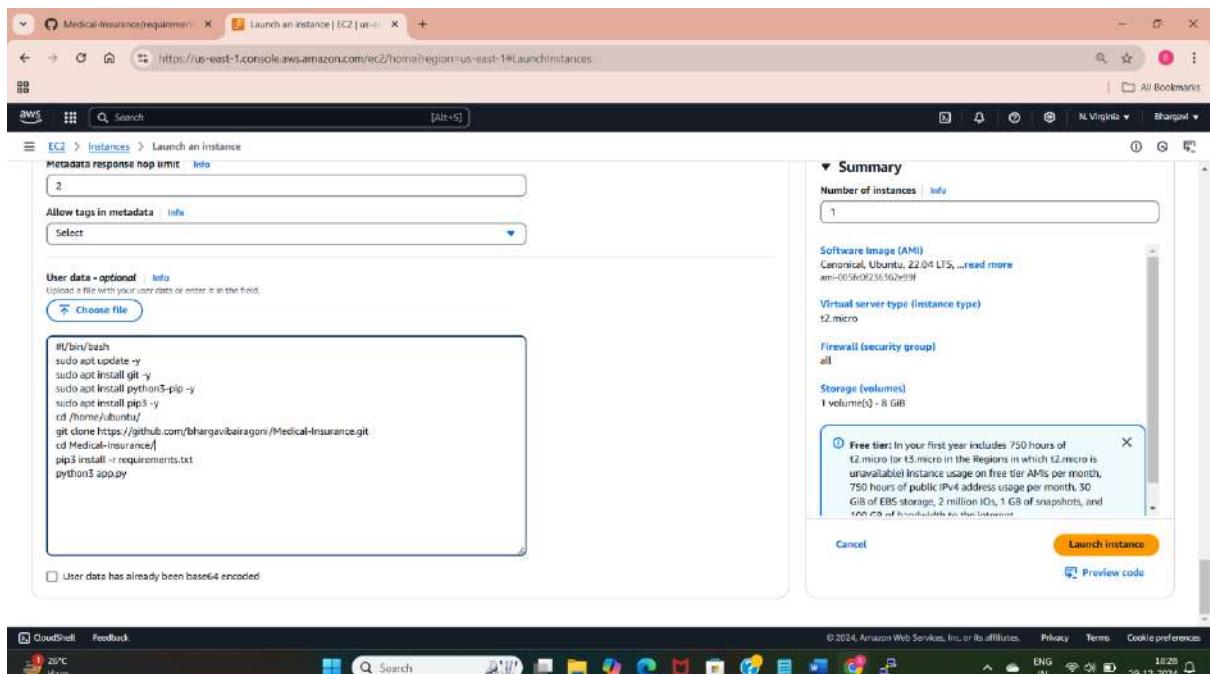
- Here I have created a vpc with the name userdata, attached internet gateway, associated route table with subnet and igw.

- Creating a Network Access Control list for providing security for subnet.

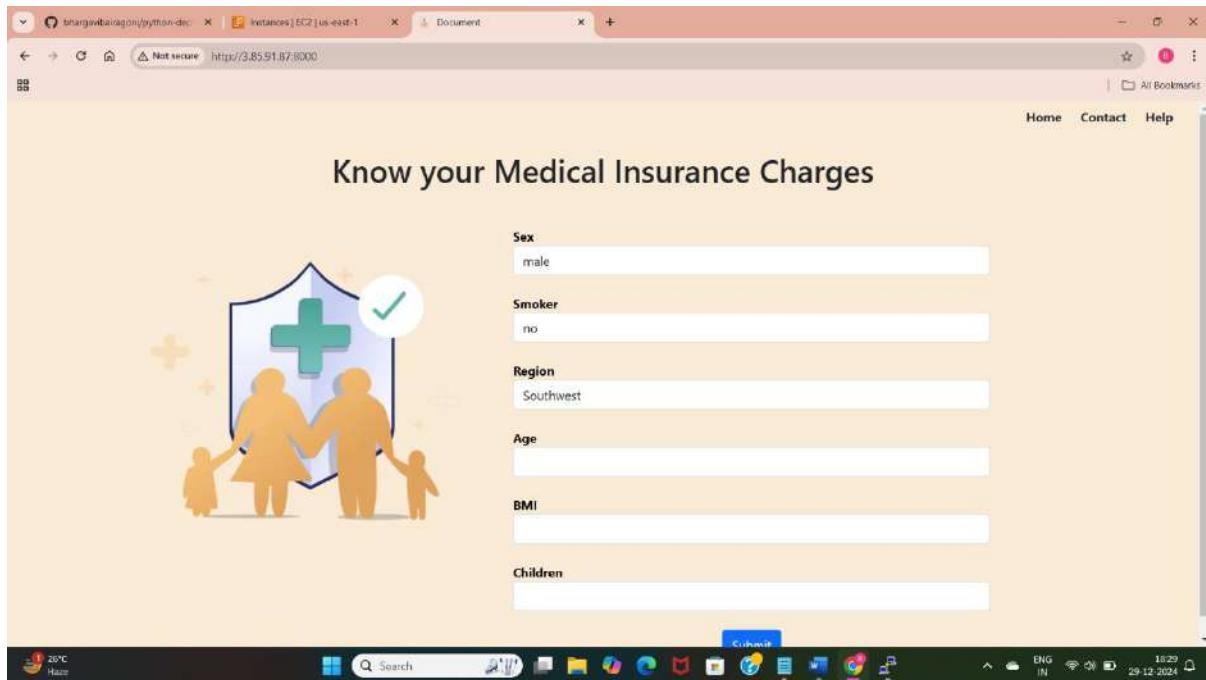
- Creating a EC2 instance with the name “userdata” and selecting ubuntu as application os, selected ubuntu server 22.04 LTS as AMI.



- Now edit the network settings by selecting the above created VPC, subnet and security group.



- Now write the bash script under the userdata by specifying the packages and requirements and moving to the path where the application is there.



- This is the python application deployed using userdata.

### METHOD-3: BUILD AND DEPLOY PYTHON APPLICATIONS IN BASH SCRIPTING

A Bash script is a simple text file that contains a list of commands written in the Bash programming language, which is used in Linux and Unix systems. It allows you to automate tasks by running these commands one after another, saving you time and effort.

- ✓ To build and deploy python applications in bash scripting first we need to launch EC2 instance and connect it with the terminal.

```
ubuntu@ip-172-31-22-150:~$ login as: ubuntu
ubuntu@ip-172-31-22-150:~$ Authenticating with public key "Imported-OpenSSH-key"
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 5.6.0-1015-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Tue Dec 24 04:42:42 UTC 2024

System load: 0.38      Processes:          109
Usage of /: 21.1% of 7.57GB   Users logged in: 0
Memory usage: 228        IPv4 address for eth0: 172.31.22.150
Swap usage: 0B

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-22-150:~$ sudo vim data.sh
ubuntu@ip-172-31-22-150:~$ sudo apt install git -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
git is already the newest version (1:2.34.1-1ubuntu1.1).
git set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
ubuntu@ip-172-31-22-150:~$ sh data.sh
[HTTP] http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
[HTTP] http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
[HTTP] http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [127 kB]
```

- ✓ Here I have connected ec2 instance with the terminal and writing a bash script file using “vim data.sh”.

```

ubuntu@ip-172-31-94-50: ~
#!/bin/bash
sudo apt update -y
sudo apt install git -y
sudo apt install python3-pip -y
sudo apt install pip3 -y
cd /home/ubuntu/
git clone https://github.com/bhargavibairagi/USA-Housing.git
cd /home/ubuntu/USA-Housing
pip3 install -r requirements.txt
python3 app.py
~

```

- ✓ This is the data.sh file written using bash script.
- ✓ The line `#!/bin/bash` is called a shebang. It tells the system to use the **Bash shell** to interpret and execute the commands in the script. Without it, the script might not run as expected, especially if a different shell is the default.
- ✓ Next updating the system packages, install git, python, pip.
- ✓ Move into the ubuntu path and clone the github url and move into that repository and install requirements of the application and deploy the application using “`python3 app.py`”.

```

ubuntu@ip-172-31-22-150: ~
Collecting scikit-learn==1.0.2
  Downloading scikit_learn-1.0.2-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (24.5 kB)
Collecting scipy==1.7.3
  Downloading scipy-1.7.3-cp310-cp310-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (39.9 kB)
Requirement already satisfied: size==1.16.0 in /usr/lib/python3/dist-packages (from -r requirements.txt (line 15)) (1.16.0)
Collecting threadpoolctl==3.1.0
  Downloading threadpoolctl-3.1.0-py3-none-any.whl (14 kB)
Collecting Werkzeug==2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
Collecting unicorn==20.1.0
  Downloading unicorn-20.1.0-py3-none-any.whl (79 kB)
Requirement already satisfied: zipp==0.5 in /usr/lib/python3/dist-packages (from importlib-metadata==4.12.0->requirements.txt (line 4)) (1.0.0)
Requirement already satisfied: setuptools==53.0 in /usr/lib/python3/dist-packages (from unicorn==20.1.0->requirements.txt (line 18)) (59.6 kB)
Installing collected packages: pytz, threadpoolctl, python-dateutil, numpy, MarkupSafe, joblib, itsdangerous, importlib-metadata, unicorn, colorama, click, Werkzeug, scipy, pandas, Jinja2, Werkzeug, Click
Attempting uninstall: pytz
  Found existing installation: pytz 2022.1
  Not uninstalling pytz at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'pytz'. No files were found to uninstall.
Attempting uninstall: MarkupSafe
  Found existing installation: MarkupSafe 2.0.1
  Not uninstalling MarkupSafe at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'MarkupSafe'. No files were found to uninstall.
Attempting uninstall: importlib-metadata
  Found existing installation: importlib-metadata 4.6.4
  Not uninstalling importlib-metadata at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'importlib-metadata'. No files were found to uninstall.
Attempting uninstall: colorama
  Found existing installation: colorama 0.4.4
  Not uninstalling colorama at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'colorama'. No files were found to uninstall.
Attempting uninstall: click
  Found existing installation: click 8.0.3
  Not uninstalling click at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'click'. No files were found to uninstall.
Attempting uninstall: Jinja2
  Found existing installation: Jinja2 3.0.3
  Not uninstalling jinja2 at /usr/lib/python3/dist-packages, outside environment /usr
  Can't uninstall 'Jinja2'. No files were found to uninstall.
Successfully installed Flask-2.0.3 Jinja2-3.1.3 MarkupSafe-3.1.1 Werkzeug-2.2.2 click-8.1.3 colorama-0.4.5 unicorn-20.1.0 importlib-metadata-4.12.0 itsdangerous-2.1.2 joblib-1.1.0 numpy-1.21.6 pandas-1.3.5 python-dateutil-2.0.2 pytz-2022.2.1 scikit-learn-1.0.2 scipy-1.7.3 threadpoolctl-3.1.0
WARNING: Running pip as the root user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv
Ubuntu@ip-172-31-22-150: ~

```

- ✓ Now run the above bashscript file using “`sh data.sh`”. Now by running this file our python application automatically deployed.

```

ubuntu@ip-172-31-22-150:~ 
Running kernel seems to be up-to-date.

Restarting services...
Service restarts being deferred:
/etc/needrestart/restart ./xibus.service
systemctl restart networkd-dispatcher.service
systemctl restart unattended-upgrades.service

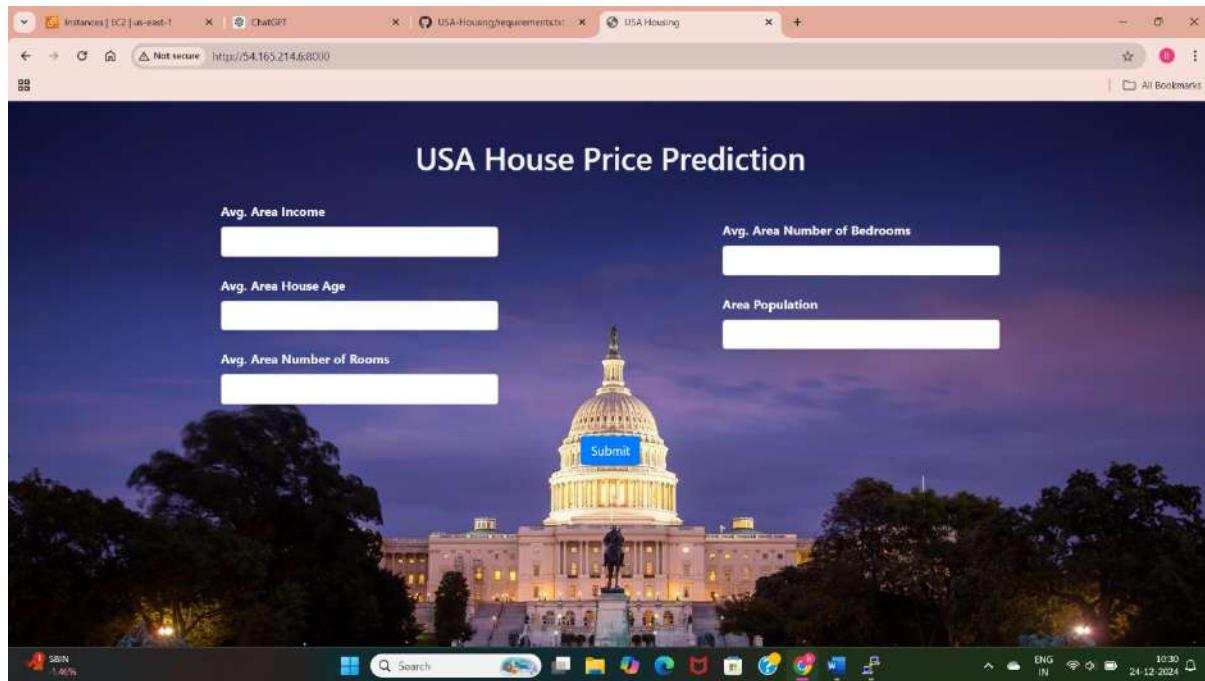
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (Qemu) binaries on this host.
ubuntu@ip-172-31-22-150:~$ sudo history
sudo: history: command not found
ubuntu@ip-172-31-22-150:~$ history
  1 sudo vim data.sh
  2 sudo apt install git -y
  3 sh data.sh
  4 sudo apt update -y
  5 sudo vim data.sh
  6 sh data.sh
  7 sudo vim data.sh
  8 ll
  9 sudo vim data.sh
 10 sudo sh data.sh
 11 pip uninstall -y scikLearn
 12 pip install scikit-learn
 13 sudo sh data.sh
 14 ls
 15 cd USA-Housing/
 16 pip3 install -r requirements.txt
 17 sudo vim requirements.txt
 18 pip3 install -r requirements.txt
 19 cd ..
 20 cd
 21 cd USA-Housing/
 22 cd
 23 sudo sh data.sh
 24 sudo vim data.sh
 25 java
 26 sudo apt install default-jre
 27 java
 28 sudo apt install openjdk-17-jre-headless
 29 sudo history
 30 history
ubuntu@ip-172-31-22-150:~$ 

```

- ✓ These are the commands used in the to deploy the python application using bash script.



- ✓ This is the python application deployed using the bashscript. We can access this by using the Public IP address of instance.

## METHOD-4: BUILD AND DEPLOY PYTHON APPLICATIONS WITH GIT, GITHUB AND JENKINS (EXECUTE SHELL)

Jenkins is an open-source automation server that helps automate parts of the software development process related to building, testing, and deploying applications. It is widely used for continuous integration (CI) and continuous delivery (CD) of projects.

- ➔ To build and deploy python application with git, github and Jenkins execute shell first we need to create a EC2 instance and setup the Jenkins by installing the dependencies and start the Jenkins. After we need to take the public IP address of the instance and use the Jenkins port 8080, and access the Jenkins.

The screenshot shows a terminal window on an Ubuntu 22.04 LTS desktop environment. The terminal output includes:

```
ubuntu@ip-172-31-27-52: ~
login as: ubuntu
Authenticating with public key "imported-openssh-key"
Welcome to Ubuntu 22.04.5 LTS (GNU/Linux 6.0.0-1015-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/pro

System information as of Mon Dec 23 16:12:16 UTC 2024

System load: 0.29 Processes: 108
Usage of /: 21.1% of 7.57GB Users logged in: 0
Memory usage: 228 IPv4 address for eth0: 172.31.27.52
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old,
to check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc//copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-172-31-27-52:~$ java
Command 'java' not found, but can be installed with:
sudo apt install openjdk-11-jre-headless  # version 11.0.24+8-1ubuntu3~22.04, or
sudo apt install default-jre            # version 2:1.11-12build12
sudo apt install openjdk-17-jre-headless # version 17.0.12+7-1ubuntu3~22.04
sudo apt install openjdk-18-jre-headless # version 18.0.2+9-2-22.04
sudo apt install openjdk-19-jre-headless # version 19.0.2+1-0ubuntu3~22.04
sudo apt install openjdk-21-jre-headless # version 21.0.4+1-1ubuntu2~22.04
sudo apt install openjdk-9-jre-headless  # version 9u422-b05-1~22.04
ubuntu@ip-172-31-27-52:~$ sudo apt install openjdk-17-jre-headless
Reading package lists... Done
```

The desktop taskbar at the bottom shows icons for various applications like Dash, Home, File, Terminal, and a clock indicating 21:44 on 23-12-2024.

- ➔ I have connected the ubuntu instance with the terminal and am installing java 17 using “sudo apt install openjdk-17-jre-headless” which is the dependency for Jenkins.
- ➔ Download the Jenkins repository, token and install using “sudo apt install Jenkins -y” & start Jenkins using “sudo systemctl start Jenkins”, Install git using “sudo apt install git -y” for taking the application code from the github.
- ➔ Next use the public IP address of instance with 8080 and paste in browse the we can able to access the Jenkins.

```

root@ip-172-31-34-234:~
perl-Error
perl-Ext
perl-TermReadKey

Transaction Summary
Install 1 Package (+5 Dependent packages)

Total download size: 19 K
Installed size: 44 K
Downloading packages:
(1/6): git-2.40.1-1.amzn2.0.3.x86_64.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64
(2/6): git-core-doc-2.40.1-1.amzn2.0.3.noarch.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64
(3/6): perl-Error-0.17020-2.amzn2.noarch.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64
(4/6): perl-core-2.40.1-1.amzn2.0.3.x86_64.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64
(5/6): perl-Git-2.40.1-1.amzn2.0.3.noarch.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64
(6/6): perl-TermReadKey-2.30-20.amzn2.0.2.x86_64.rpm 1:0.17020-2.amzn2.noarch 2.40.1-1.amzn2.0.3.x86_64

Total
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : git-core-2.40.1-1.amzn2.0.3.x86_64
  Installing : git-core-doc-2.40.1-1.amzn2.0.3.noarch
  Installing : perl-Error-0.17020-2.amzn2.noarch
  Installing : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64
  Installing : perl-Git-2.40.1-1.amzn2.0.3.noarch
  Installing : perl-Git-2.40.1-1.amzn2.0.3.x86_64
  Verifying : git-2.40.1-1.amzn2.0.3.x86_64
  Verifying : git-core-2.40.1-1.amzn2.0.3.x86_64
  Verifying : git-core-doc-2.40.1-1.amzn2.0.3.noarch
  Verifying : perl-Error-0.17020-2.amzn2.noarch
  Verifying : perl-Git-2.40.1-1.amzn2.0.3.x86_64
  Verifying : perl-Git-2.40.1-1.amzn2.0.3.noarch
  Verifying : perl-TermReadKey-2.30-20.amzn2.0.2.x86_64

Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.3

Dependency Installed:
  git-core.x86_64 0:2.40.1-1.amzn2.0.3          git-core-doc.noarch 0:2.40.1-1.amzn2.0.3      perl-Error.noarch 1:0.17020-2.amzn2           perl-Git.noarch 0:2.40.1-1.amzn2.0.3

Dependencies Resolved

root@ip-172-31-24-234:~# systemctl start jenkins
[root@ip-172-31-24-234 ~]# cat /var/lib/jenkins/secrets/initialAdminPassword
ccae7a42ff647e8a93f4c70777db49
[root@ip-172-31-24-234 ~]# 
[root@ip-172-31-24-234 ~]# 

Breaking news
J Khalistan Int... 11:45 23-12-2024
```

- After that we need to give the password for Jenkins to create account in Jenkins, in Jenkins gui we get the path which is having the password for Jenkins use “cat <path provided in Jenkins>”. Now we will get the password for Jenkins paste that password in Jenkins gui.

New Item

Enter an item name  
bashscript

Select an item type

**Freestyle project**  
Classic, general purpose job type that checks out from up to one SCM, executes build steps serially, followed by post-build steps like archiving artifacts and sending email notifications.

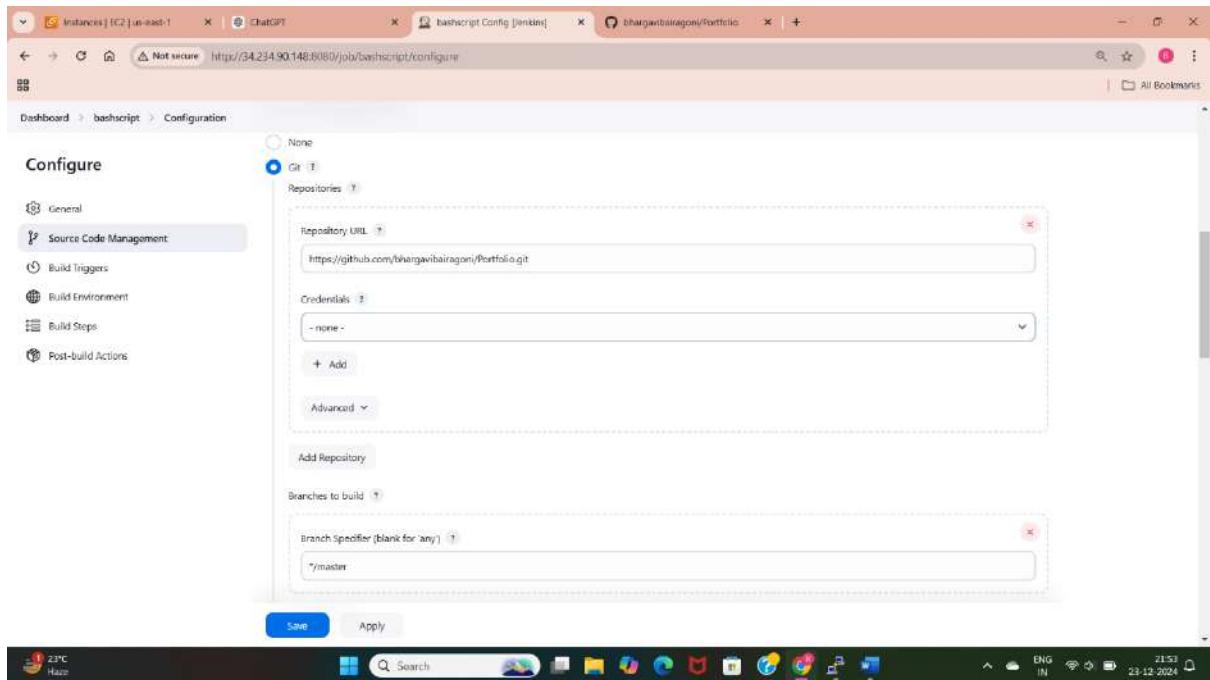
**Pipeline**  
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

**Multi-configuration project**  
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

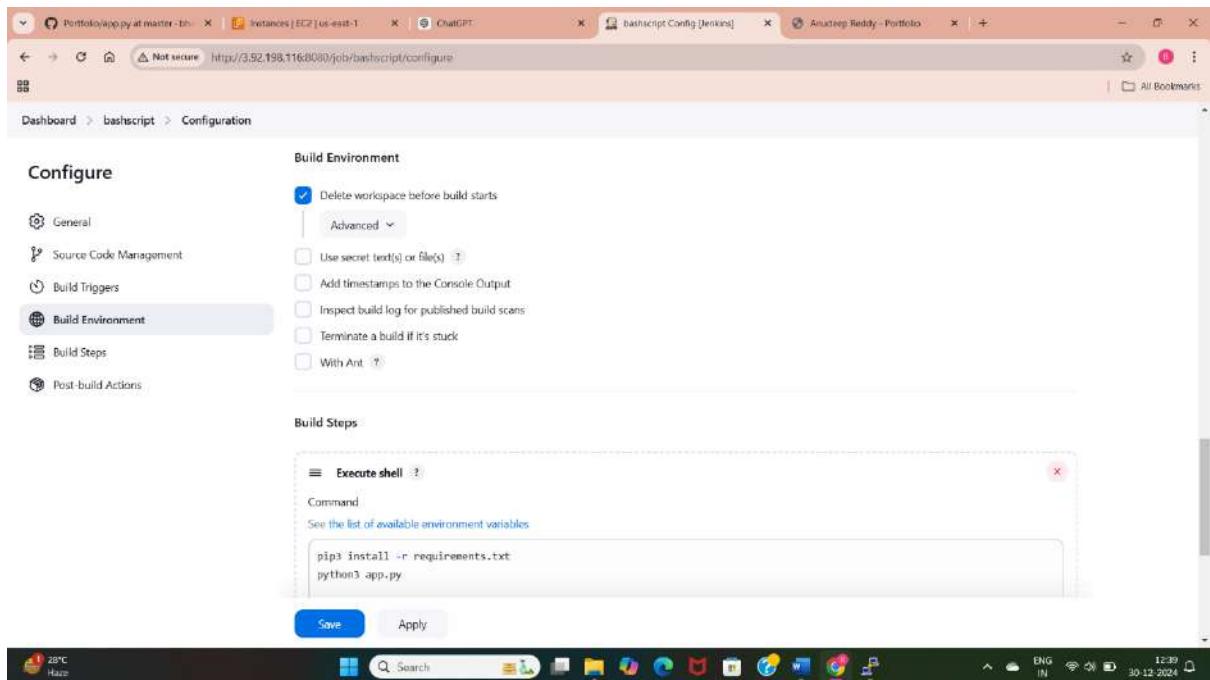
**Folder**  
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a

OK

- Now we will get the Jenkins dashboard, create a freestyle job for deploying python application.



- I have created a Jenkins freestyle job with the name “bashscript”, taking the python application code from the git.
- In Jenkins job select the “Git” and give the application url, select the branch where you are having the code, am having code in “master” branch.



- Now am using the “execute shell”
- Go to build steps, select execute shell and give the script to move into the job where we are having the Portfolio application.
- Give the command to install the requirements and deployment command and save the changes.

```

root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript
drwxr-xr-x 4 jenkins jenkins 4096 Dec 23 16:16 .cache/
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:20 .gitaway/
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:16 .java/
-rw-r--r-- 1 jenkins jenkins 0 Dec 23 16:16 .lastStarted
-rw-r--r-- 1 jenkins jenkins 1660 Dec 23 16:16 config.xml
-rw-r--r-- 1 jenkins jenkins 156 Dec 23 16:16 jenkins-model-updatercenter.xml
-rw-r--r-- 1 jenkins jenkins 300 Dec 23 16:16 jenkins-model-updatercenter.git.utillist.xml
-rw-r--r-- 1 jenkins jenkins 1600 Dec 23 16:20 identity.key.enc
-rw-r--r-- 1 jenkins jenkins 7 Dec 23 16:20 jenkins.install.InstallUtil.lastExecVersion
-rw-r--r-- 1 jenkins jenkins 7 Dec 23 16:20 jenkins.install.UpgradeWizard.state
-rw-r--r-- 1 jenkins jenkins 193 Dec 23 16:20 jenkins.model.JenkinsLocationConfiguration.xml
-rw-r--r-- 1 jenkins jenkins 171 Dec 23 16:16 jenkins.telemetry.Correlator.xml
drwxr-xr-x 2 jenkins jenkins 4096 Dec 23 16:30 jems/
drwxr-xr-x 2 jenkins jenkins 4096 Dec 23 16:30 logs/
drwxr-xr-x 1 jenkins jenkins 1037 Dec 23 16:16 nodeMonitors.xml
drwxr-xr-x 50 jenkins jenkins 12200 Dec 23 16:16 pipelines/
drwxr-xr-x 1 jenkins jenkins 258 Dec 23 16:31 secrets.xml
-rw-r--r-- 1 jenkins jenkins 64 Dec 23 16:16 secret.kay
-rw-r--r-- 1 jenkins jenkins 0 Dec 23 16:16 secret.kay.not-so-secret
drwxr-xr-x 2 jenkins jenkins 4096 Dec 23 16:24 secrets/
drwxr-xr-x 2 jenkins jenkins 4096 Dec 23 16:20 updates/
drwxr-xr-x 2 jenkins jenkins 4096 Dec 23 16:16 userContent/
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:20 users/
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:24 workspace/
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# cd workspace
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# ls
total 12
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:24 ./
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:31 ../
drwxr-xr-x 6 jenkins jenkins 4096 Dec 23 16:24 bashscript/
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# cd bashscript/
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# ll
total 40
drwxr-xr-x 6 jenkins jenkins 4096 Dec 23 16:24 ./
drwxr-xr-x 8 jenkins jenkins 4096 Dec 23 16:34 ../
drwxr-xr-x 0 jenkins jenkins 4096 Dec 23 16:24 .git/
drwxr-xr-x 4 jenkins jenkins 4096 Dec 23 16:24 MyPortfolio/
-rw-r--r-- 1 jenkins jenkins 21 Dec 23 16:24 .profile
-rw-r--r-- 1 jenkins jenkins 363 Dec 23 16:24 app.py
-rw-r--r-- 1 jenkins jenkins 1696 Dec 23 16:24 gitattributes
-rw-r--r-- 1 jenkins jenkins 202 Dec 23 16:24 requirements.txt
drwxr-xr-x 7 jenkins jenkins 4096 Dec 23 16:24 static/
drwxr-xr-x 3 jenkins jenkins 4096 Dec 23 16:24 static/
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# pwd
/wsl/lib/jenkins/workspace/bashscript
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# ~
root@ip-172-31-27-52:/var/lib/jenkins/workspace/bashscript# 

```

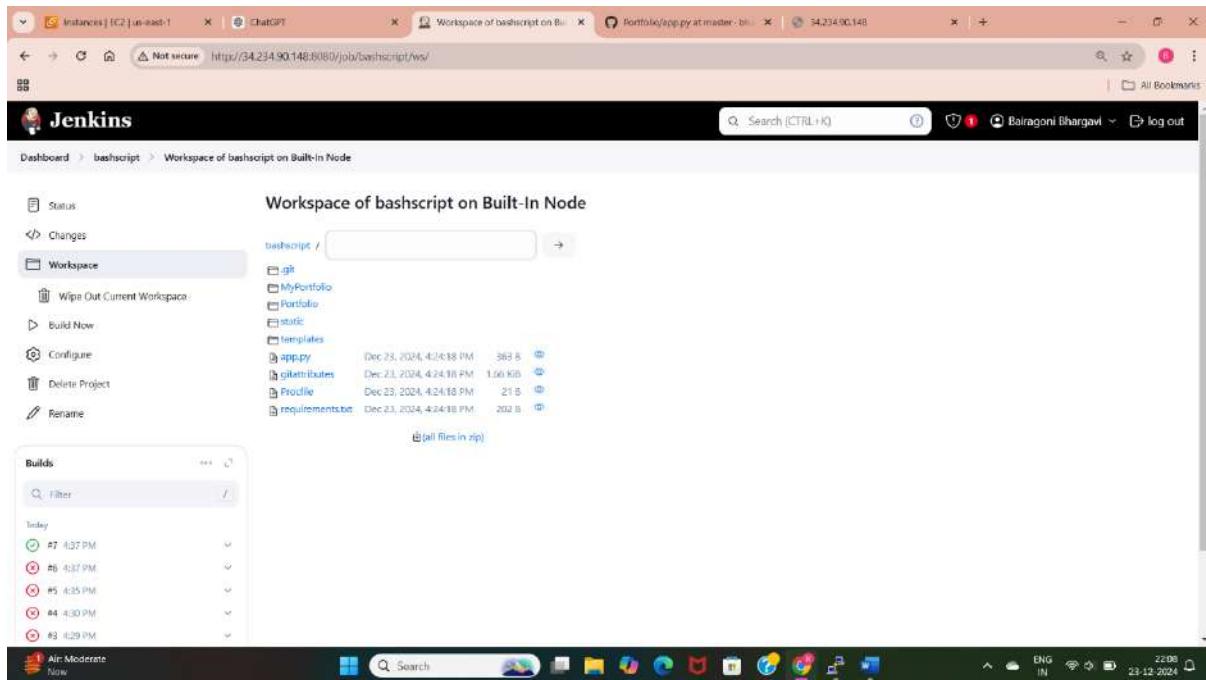
- Here am checking the files by moving into “/var/lib/Jenkins/workspace/jobname”, we can see the application with the name “MyPortfolio”.

The screenshot shows the Jenkins interface for the 'bashscript' job. At the top, there's a navigation bar with tabs for 'Instances | EC2 | u6-had-1', 'ChatGPT', and 'bashscript [Jenkins]'. Below the navigation bar, the main content area has a header 'Jenkins' with a sub-header 'Dashboard > bashscript >'. The main content includes:

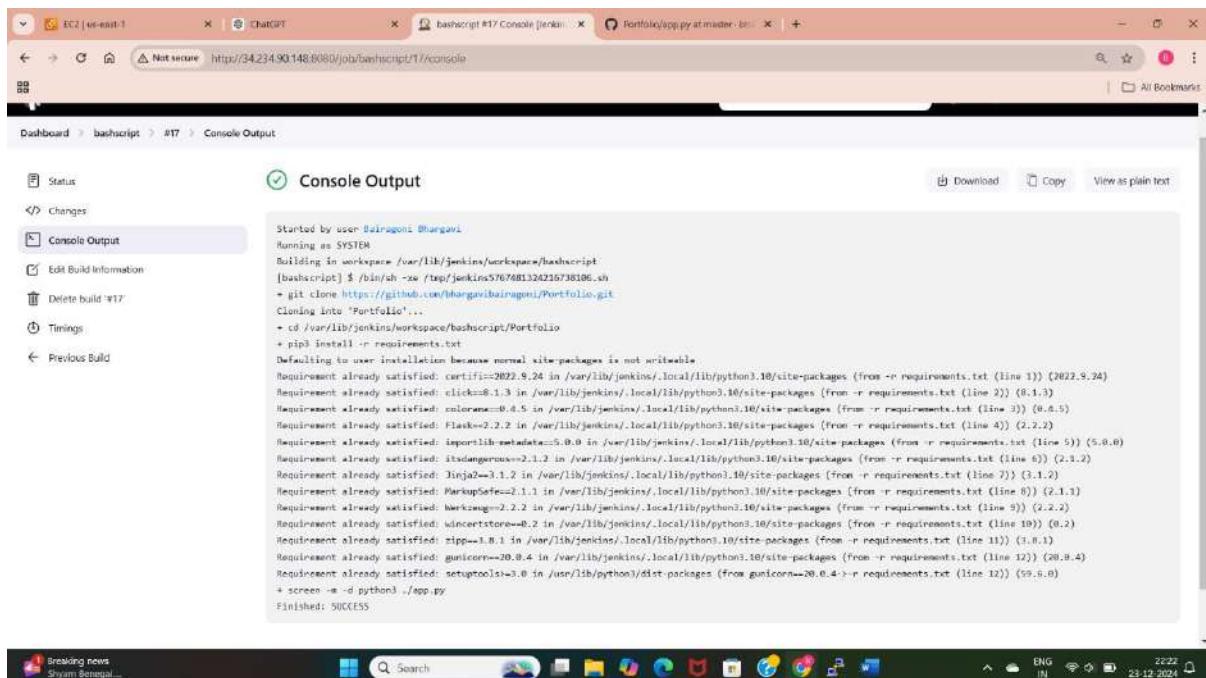
- Status:** Shows the job is 'Up-to-date' with a green checkmark icon.
- Permalinks:** A list of build links:
  - Last build (#7), 1 min 7 sec ago
  - Last stable build (#7), 1 min 7 sec ago
  - Last successful build (#7), 1 min 7 sec ago
  - Last failed build (#6), 1 min 33 sec ago
  - Last unsuccessful build (#6), 1 min 33 sec ago
  - Last completed build (#7), 1 min 7 sec ago
- Builds:** A table showing build history from today:
 

Build	Time
#7	4:37 PM
#6	4:37 PM
#5	4:35 PM
#4	4:30 PM
#3	4:29 PM
#2	4:25 PM

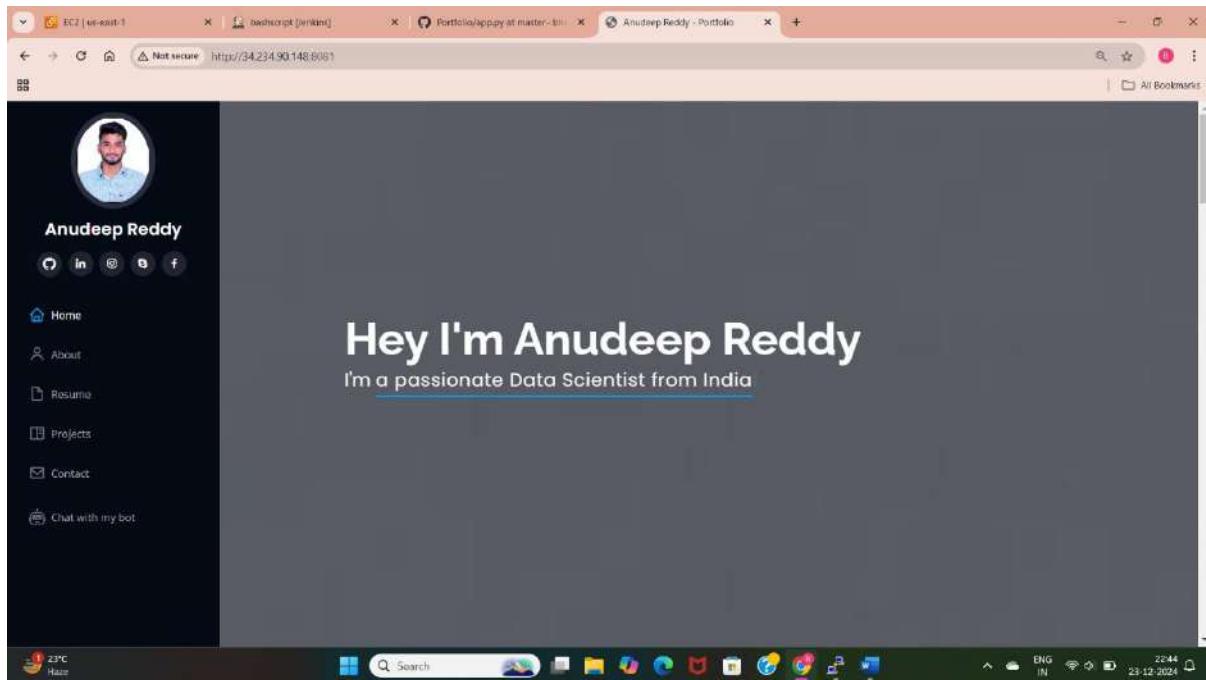
- Now go to the bashscript job in jenkins and click “build now” option, if there is no error in the configurations, script then only our build will successful.



→ Now we can see the files by using workspace option in the Jenkins GUI.



→ We can also check with the “console output” so we can able to see what we have executed in the Jenkins.



- Now by using the public IP address and application port number we can able to access the application.

## METHOD-5: BUILD AND DEPLOY PYTHON APPLICATIONS WITH THE TERRAFORM(DATA.SH) AND PUSH THE TERRAFORM SCRIPTED FILES IN GITHUB:

To deploy the python application using the terraform first we need to launch a instance and install the terraform and give the aws credentials to access the aws services.

```
ubuntu@ip-172-31-22-150: ~
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
run this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
ubuntu@ip-172-31-22-150: ~$ sudo aws configure
aws: command not found
ubuntu@ip-172-31-22-150: ~$ aws --version
Command 'aws' not found, but can be installed with:
sudo snap install aws-core=1.15.59* or
sudo apt-get install awscli=1.22.344*
See 'snap info aws-clif' for additional versions.
ubuntu@ip-172-31-22-150: ~$ 
ubuntu@ip-172-31-22-150: ~$ sudo snap install aws-clii
error: This revision of snap "aws-clii" was published using classic confinement and thus may perform
arbitrary system changes outside of the security sandbox that snaps are usually confined to,
which may put your system at risk.

If you understand and want to proceed repeat the command including --classic.
ubuntu@ip-172-31-22-150: ~$ 
ubuntu@ip-172-31-22-150: ~$ sudo apt install awscli
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  awscli-common fontconfig fonts-otf-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsfonts hicolor-icon-theme imagemagick imagemagick-6-common imagemagick-6.q16 libbam3
  libbaario2 libdav1d5 libde265-0 libdyvulibre-text libfftw3-double3 libgs3-common libheif1 libimlib2 libis3-0.35 liblmbase25 libmagickquant0 libtbig2dec
  libmagick2-0.1-0 libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libopenexr2 libopenjp2-7 libpango-1.0-0 libpango-1.0-0
  libpango-1.0-0 libpaper-utils libpaper1 libpixman-1 libx11 libxext libxpm libxrender0 mailcap mime-support netpbm
  poppler-data python3-botcore python3-dateutil python3-docutils python3-imsgpath python3-pil python3-pygments python3-roman python3-s3transfer
  xml base xml-core
Suggested packages:
  font-noto fonts-freefont-ttf fonts-texgyre ghostscript-x imagemagick-doc autorotate cups-hd | lpr | lprng exscript ffsmpg gimp gnuplot grads graphviz hp2xx html2ps
  libavformat libavutil libdc1394-0.1 libdav1d5 libdyvulibre-text libfftw3-bin libfftw3-dev inkscape poppler-utils fonts-japanese-mincho
  fonts-ipafont-mincho fonts-japanese-gothic | fonts-ipafont-gothic fonts-aptic-ukai fonts-aptic-uming fonts-nanum docutils-doc fonts-latinlibertine | ttf-latin-libertine
  texlive-latex-french texlive-latex-base texlive-latex-recommended python-pil-doc python-pymments-doc ttf-bitstream-vera sgml-base-doc debhelper
The following NEWER packages will be installed:
  awscli-common fontconfig fonts-otf-droid-fallback fonts-noto-mono fonts-urw-base35 ghostscript groff gsfonts hicolor-icon-theme imagemagick-6-common imagemagick-6.q16
  libbam3 libbaario2 libdav1d5 libde265-0 libdyvulibre-text libfftw3-double3 libgs3-common libheif1 libimlib2 libis3-0.35 liblmbase25 libmagickquant0
  libmagick2-0.1-0 libmagickcore-6.q16-6 libmagickcore-6.q16-6-extra libmagickwand-6.q16-6 libopenexr2 libopenjp2-7 libpango-1.0-0
  libpango-1.0-0 libpaper-utils libpaper1 libpixman-1 libx11 libxext libxpm libxrender0 mailcap mime-support netpbm poppler-data python3-botcore python3-dateutil python3-docutils python3-imsgpath python3-pil python3-pygments python3-roman python3-s3transfer
  xml base xml-core
EBC ENG 24/12/2024 12:40
```

- ⇒ Now connect the ec2 instance with the terminal and use “aws configure” provide access keys, secret access keys and region.
- ⇒ Here am using the ubuntu so we need to install the awscli then provide aws configure.
- ⇒ Install the terraform.

```

ubuntu@ip-172-31-22-150: ~
-rw-r--r-- 1 ubuntu ubuntu 1263 Dec 24 06:12 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Jan 6 2022 .hash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Jan 6 2022 .bashrc
drwxr-xr-x 3 ubuntu ubuntu 4096 Dec 24 04:49 .cache/
-rw-r--r-- 1 root root 12208 Dec 24 05:29 ec2.tf.swp
drwxrwxr-x 9 ubuntu ubuntu 4096 Dec 24 05:29 igis/
drwxrwxr-x 8 ubuntu ubuntu 4096 Dec 24 05:29 igis/
drwxrwxr-x 8 ubuntu ubuntu 4096 Dec 24 05:29 igis/
drwxrwxr-x 8 ubuntu ubuntu 4096 Dec 24 05:29 igis/
-rw-r--r-- 1 ubuntu ubuntu 897 Jan 6 2022 installfile
drwxr-xr-x 2 ubuntu ubuntu 4096 Dec 24 04:42 .ssh/
-rw-r--r-- 1 ubuntu ubuntu 9 Dec 24 04:42 sudo as admin successful
-rw-rw-r-- 1 ubuntu ubuntu 181 Dec 24 06:59 .wget-hists
drwxrwxr-x 6 ubuntu ubuntu 4096 Dec 24 04:57 DSA_Housing/
-rw-r--r-- 1 root root 299 Dec 24 05:01 data.sh
-rw-r--r-- 1 root root 618 Dec 24 05:37 ec2.tftf
-rw-r--r-- 1 root root 301 Dec 24 06:57 vpc.tftf
ubuntu@ip-172-31-22-150: ~$ sudo vim igis.tftf
ubuntu@ip-172-31-22-150: ~$ sudo vim igis.tftf
ubuntu@ip-172-31-22-150: ~$ git add route.tftf
ubuntu@ip-172-31-22-150: ~$ git commit -m "committing"
[master 43c9eaa] committing
  Committer: Ubuntu <ubuntu@ip-172-31-22-150.ec2.internal>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:
git config --global --edit
After doing this, you may fix the identity used for this commit with:
  git commit --amend --reset-author

3 files changed, 21 insertions(+)
create mode 100644 igis.tftf
create mode 100644 route.tftf
ubuntu@ip-172-31-22-150: ~$ git push -u origin master
Username for 'https://github.com': ph@80encoddd41LXQY2Tfzs20838XnL7ZGsaTa
Password for 'https://ph@80encoddd41LXQY2Tfzs20838XnL7ZGsaTa@github.com':
Enumerating objects: 5, done.
Counting objects: 100% (5/5), done.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (5/5) | 699.00 KiB/s, done.
Total 5 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/charansivaloopath/python-deployment.git
 ! [ref create] master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
ubuntu@ip-172-31-22-150: ~$ terraform init

```

- Write the terraform file creating ec2 instance, VPC, subnet, IGW, route table and data.sh file which is having the python application.

```

ubuntu@ip-172-31-29-81: ~
resource "aws_vpc" "demovpc" {
  cidr_block = "10.0.0.0/16"
  instance_tenancy = "default"
  tags = {
    Name = "DEMO VPC"
  }
}
resource "aws_subnet" "public_subnet-1" {
  vpc_id = "${aws_vpc.demovpc.id}"
  cidr_block = "10.0.1.0/24"
  map_public_ip_on_launch = true
  availability_zone = "us-east-1a"
  tags = {
    Name = "Web Subnet 1"
  }
}

```

- This is the vpc.tf file which is used to create a vpc with a subnet and provided the configurations related to them.

```

ubuntu@ip-172-31-29-81: ~
#!/bin/bash
sudo apt update -y
sudo apt install git -y
sudo apt install python3-pip -y
sudo apt install pip3 -y
mkdir /bhargavi
cd /home/ubuntu/
git clone https://github.com/bhargavibairagi/USA-Housing.git
cd /home/ubuntu/USA-Housing
pip3 install -r requirements.txt
screen -m -d python3 ./app.py
~
```

- This is the data.sh file which is updating the system, installing git, python, and pip. Cloned the application from github in the /home/ubuntu path, installed the requirements(dependencies) of application and finally running/deploying the application using terraform.

```

ubuntu@ip-172-31-19-232: ~
resource "aws_instance" "public_subnet-1" {
ami="ami-005fc0f236362e99f"
instance_type="t2.micro"
count=1
key_name="new-key"
vpc_security_group_ids=["${aws_security_group.demosg.id}"]
subnet_id="${aws_subnet.public_subnet-1.id}"
associate_public_ip_address=true
user_data="${file("data.sh")}"
tags={}
Name="My polscom 2"
}
resource "aws_security_group" "demosg" {
vpc_id="${aws_vpc.demovpc.id}"
ingress{
from_port=80
to_port=80
protocol="tcp"
cidr_blocks=["0.0.0.0/0"]
}
ingress{
from_port=8000
to_port=9000
protocol="tcp"
cidr_blocks=["0.0.0.0/0"]
}
ingress{
from_port=22
to_port=22
protocol="tcp"
cidr_blocks=["0.0.0.0/0"]
}
ingress{
from_port=0
to_port=0
protocol="-1"
cidr_blocks=["0.0.0.0/0"]
}
egress{
from_port=0
to_port=0
protocol="-1"
cidr_blocks=["0.0.0.0/0"]
-- INSERT --
}
```

- Terraform configurations for creating ec2 instance and security group.

```
ubuntu@ip-172-31-19-232: ~

resource "aws_route_table" "route" {
  vpc_id = "${aws_vpc демовр.id}"
  route {
    cidr_block = "0.0.0.0/0"
    gateway_id = "${aws_internet_gateway демогейвей.id}"
  }
  tags = {
    Name = "Route to Internet"
  }
}

resource "aws_route_table_association" "rt1" {
  subnet_id = "${aws_subnet.public_subnet-1.id}"
  route_table_id = "${aws_route_table.route.id}"
}

~
```

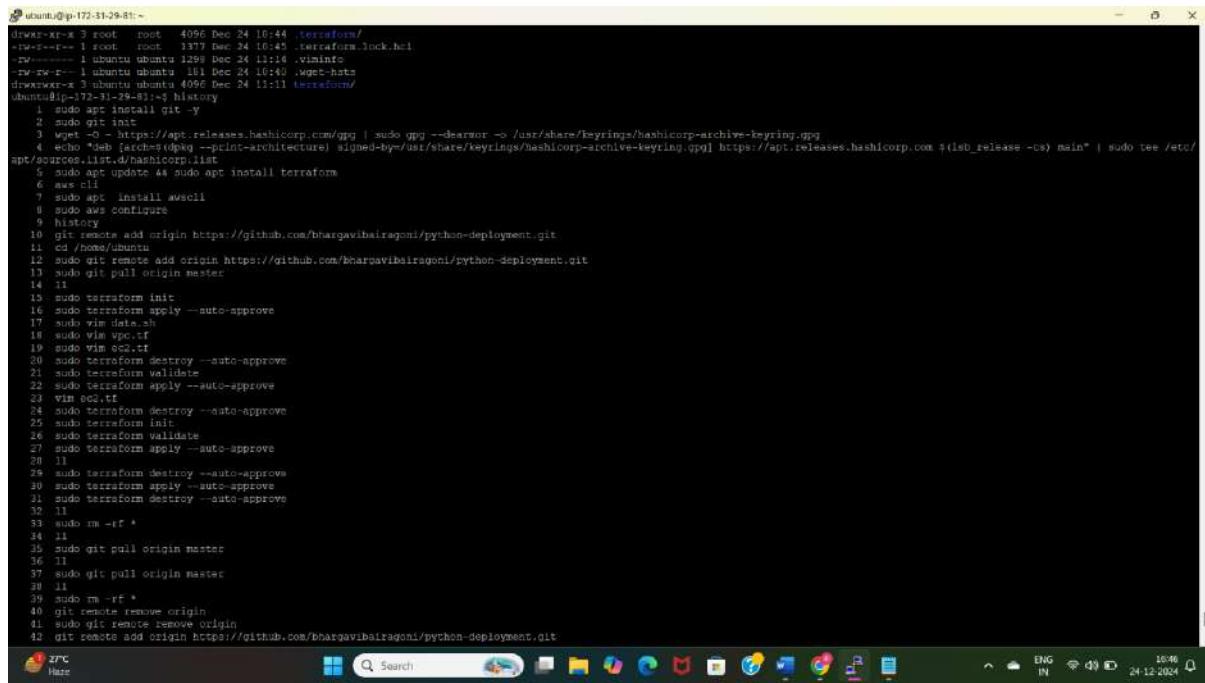
- Terraform code for creating route table and associating it with subnet.

```
ubuntu@ip-172-31-19-232: ~

resource "aws_internet_gateway" "demogateway" {
  vpc_id = "${aws_vpc демовр.id}"
  tags = {
    Name = "Internet Gateway"
  }
}

~
```

- Terraform code for creating internet gateway.

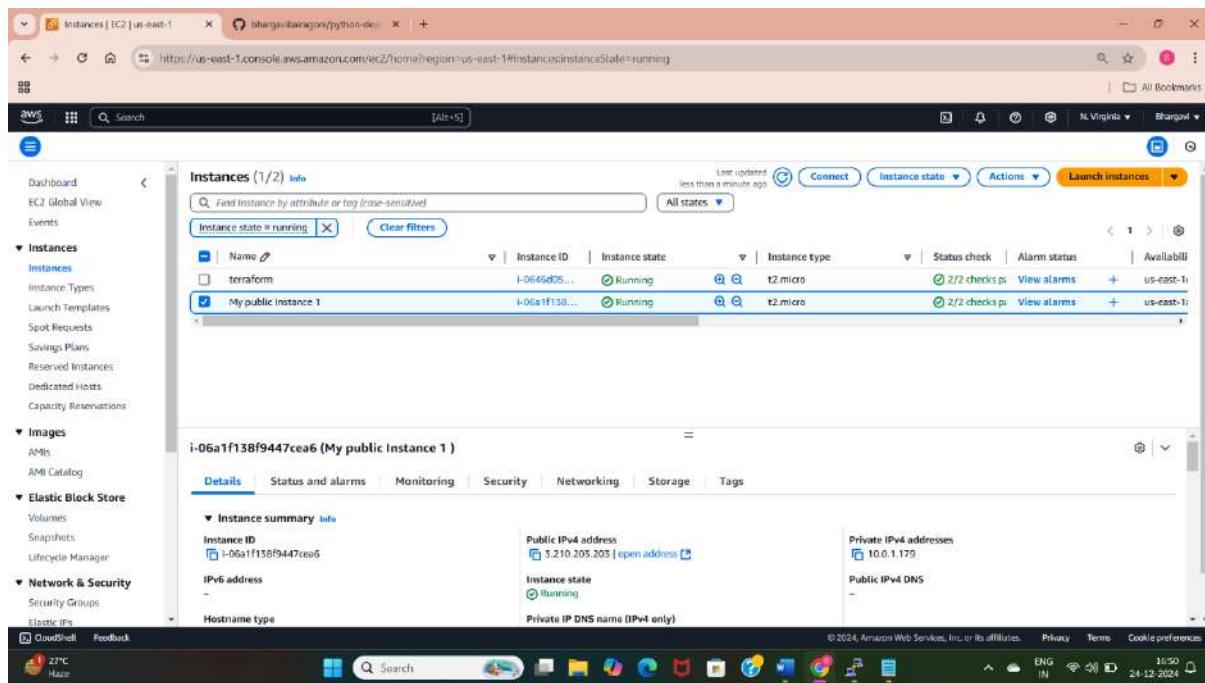


```

ubuntu@ip-172-31-29-81:~ 
dowm-xr-x 3 root root 4096 Dec 24 10:44 .terraform/
-rw-r--r-- 1 root root 1377 Dec 24 10:45 .terraform.lock.hcl
-rw-r--r-- 1 ubuntu ubuntu 1299 Dec 24 11:14 .viminfo
-rw-r--r-- 1 ubuntu ubuntu 181 Dec 24 10:40 .wget-hsts
dowm-xr-x 3 ubuntu ubuntu 4096 Dec 24 11:11 .terraform/
ubuntu@ip-172-31-29-81:~$ history
2 sudo apt install git -y
3 sudo apt int
3 wget -O https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg
4 echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/
apt/sources.list.d/hashicorp.list
5 sudo apt update
6 aws cli
7 sudo apt install awscli
8 sudo aws configure
9 history
10 sudo git remote add origin https://github.com/bhargavibairagomi/python-deployment.git
11 cd /home/ubuntu
12 sudo git remote add origin https://github.com/bhargavibairagomi/python-deployment.git
13 sudo git pull origin master
14 ll
15 sudo terraform init
16 sudo terraform apply --auto-approve
17 sudo vim data.sh
18 sudo vim vpc.tf
19 sudo terraform init
20 sudo terraform destroy --auto-approve
21 sudo terraform validate
22 sudo terraform apply --auto-approve
23 vim ec2.tf
24 sudo terraform destroy --auto-approve
25 sudo terraform init
26 sudo terraform validate
27 sudo terraform apply --auto-approve
28 ll
29 sudo terraform destroy --auto-approve
30 sudo terraform apply --auto-approve
31 sudo terraform destroy --auto-approve
32 ll
33 sudo rm -rf *
34 ll
35 sudo git pull origin master
36 ll
37 sudo git pull origin master
38 ll
39 sudo rm -rf *
40 git remote remove origin
41 sudo git remote remove origin
42 git remote add origin https://github.com/bhargavibairagomi/python-deployment.git

```

- These are the commands used to install terraform, create terraform files, automate the process of deploying using terraform.



- By running the above files using “terraform apply” we can able to check with the aws console that a new instance is created.

The screenshot shows the AWS VPC console. On the left, there's a sidebar with navigation links like 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), 'Security' (Network ACLs, Security groups), and 'PrivateLink and Lattice' (Getting started, Updated). The main area displays 'Your VPCs (1/2) Info' with a table showing one VPC named 'DEMO VPC' with VPC ID 'vpc-03e329e8292792cf6'. Below this, under 'vpc-03e329e8292792cf6 / DEMO VPC', there are tabs for Details, Resource map, CIDRs, Flow logs, Tags, and Integrations. The Resource map tab is selected, showing a diagram with four components: VPC (Your AWS virtual network), Subnets (1), Route tables (2), and Network connections (1). A blue box highlights 'Web Subnet 1' under Subnets. A blue line connects it to 'Route to Internet' under Route tables, which is further connected to 'Internet Gateway' under Network connections.

- ⌚ New VPC with the specified configurations is creating, attached to the internet gateway.
- ⌚ Subnet is associated to the route table and route table is associated with the internet gateway to allow the traffic from and to the VPC.

The screenshot shows the AWS VPC console. The sidebar includes 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), 'Security' (Network ACLs, Security groups), and 'PrivateLink and Lattice' (Getting started, Updated). The main area shows 'Internet gateways (1/2) Info' with a table listing one Internet gateway named 'Internet Gateway' with Internet gateway ID 'igw-06b100f58fdaf5c3da'. Below this, under 'igw-06b100f58fdaf5c3da / Internet Gateway', there are tabs for Details and Tags. The Details tab shows the Internet gateway ID 'igw-06b100f58fdaf5c3da', State 'Attached', VPC ID 'vpc-03e329e8292792cf6 | DEMO VPC', and Owner '905418365089'.

- ⌚ With the terraform script I have created a internet gateway for providing internet to the VPC.
- ⌚ After creating IGW attach it to the VPC.

**Subnets (7) Info**

Name	Subnet ID	State	VPC	IPv4 CIDR	IPv6 ...	IPv6 CIDR association ID	Available IPv4 addr.
subnet-01657fa4557...	vpc-03f5f...	Available	172.51.48.0/20	-	-	-	4091
subnet-0b407fb5e517...	vpc-03f5f...	Available	172.51.0.0/20	-	-	-	4091
subnet-0bf503e1b007...	vpc-03f5f...	Available	172.51.64.0/20	-	-	-	4091
subnet-09c3ec58856f...	vpc-03f5f...	Available	172.51.80.0/20	-	-	-	4091
subnet-0ff75dfe9ba...	vpc-03f5f...	Available	172.51.16.0/20	-	-	-	4090
subnet-0bb0d5cfefc...	vpc-03f5f...	Available	172.51.32.0/20	-	-	-	4091
Web Subnet 1	subnet-0bb0d5cfefc...	Available	vpc-03e5e...	10.0.1.0/24	-	-	250

Select a subnet:

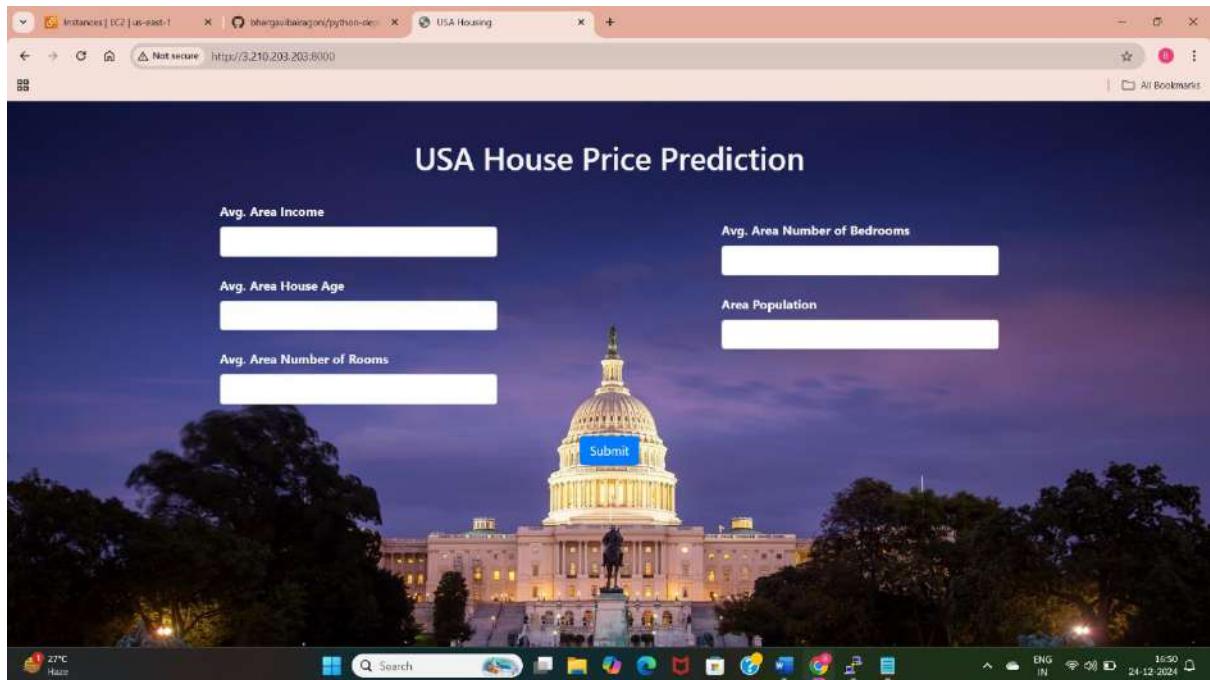
⌚ This is the subnet created using the terraform script.

**Route tables (3) Info**

Name	Route table ID	Expli...	Edge associations	Main	VPC	Owner ID
rtb-0be2de17f6670a...	-	-	-	Yes	vpc-05f5fe3fd53d1f03	905418365089
Route to internet	rtb-013eb50f750b6...	subnet-...	-	No	vpc-05e329e8292792e6   DEMO VPC	905418365089
rtb-0ceef1aeb05729...	-	-	-	Yes	vpc-03e329e8292792e6   DEMO VPC	905418365089

Select a route table:

⌚ This is the route table created using the terraform script to allow the routes.



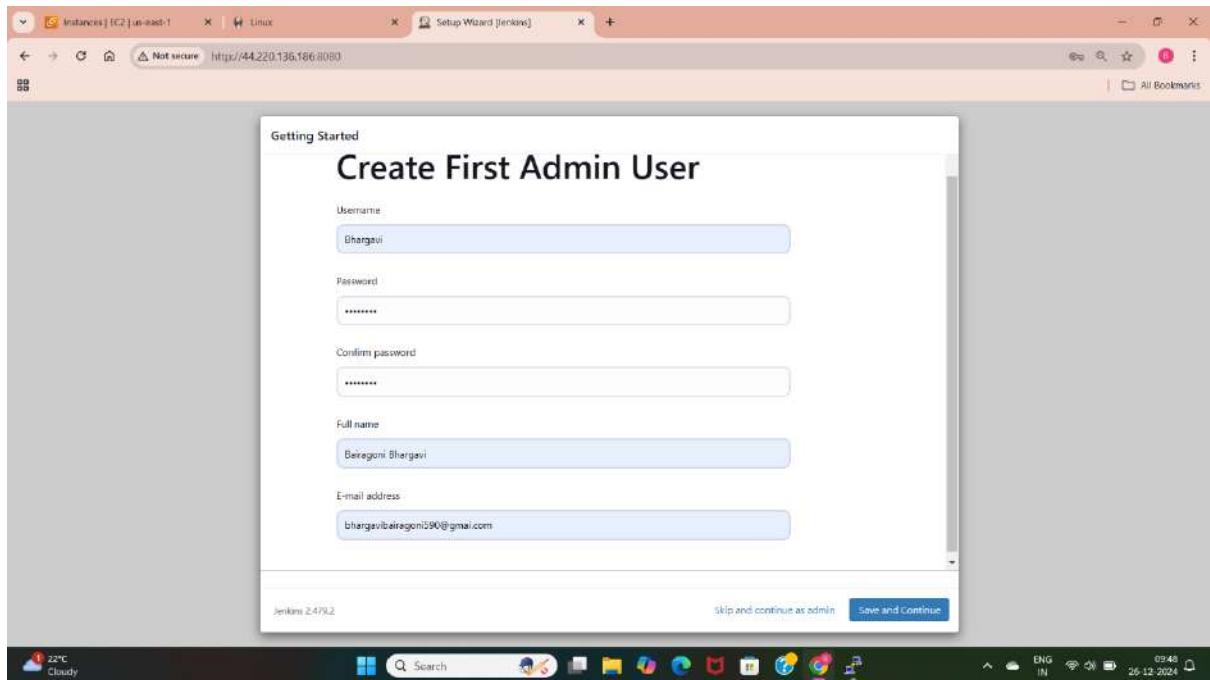
- ⦿ After performing the “terraform apply” we will get the new instance, copy the public IP address of that instance and paste in the browser with the application port number, then we can able to access the application.

#### **METHOD-6: BUILD AND DEPLOY PYTHON APPLICATIONS WITH GIT, GITHUB, JENKINS AND TERRAFORM**

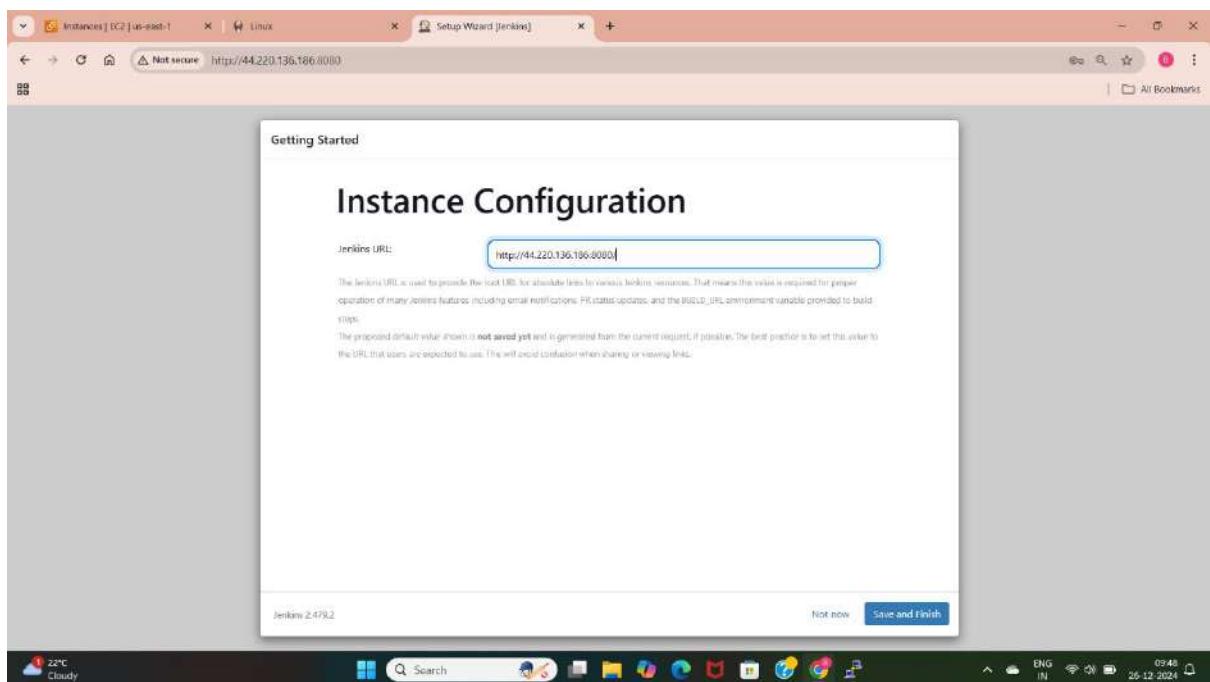
To deploy python application using git, github, Jenkins and terraform first we need to create a EC2 instance and connect it with terminal and install terraform, and setup Jenkins.

Terraform is an open-source Infrastructure as Code (IAC) tool developed by HashiCorp that allows you to define, provision, and manage infrastructure using a declarative configuration language.

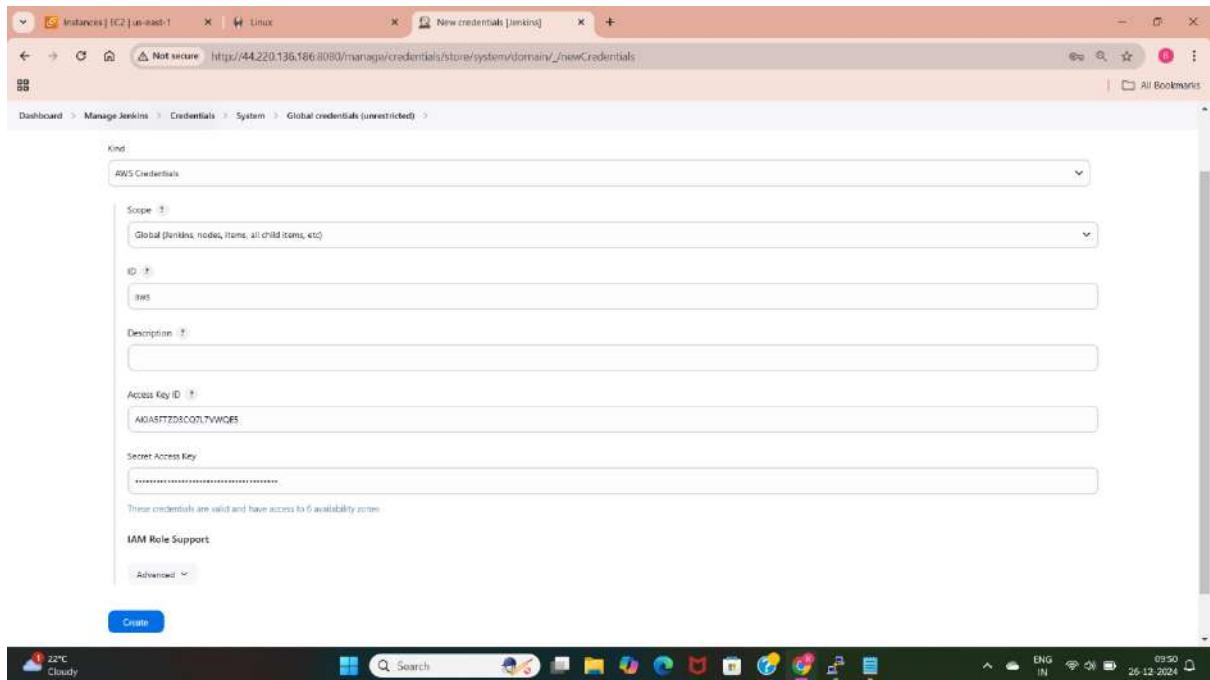
- ❖ I have created a EC2 instance with the ubuntu os and t2.medium instance type, installed dependency of Jenkins java17 and Jenkins, terraform, awscli (yum install awscli -y) for providing the AWS credentials.
- ❖ Copy the public IP address of instance and paste it in browser with Jenkins port, then we can able to access the Jenkins.



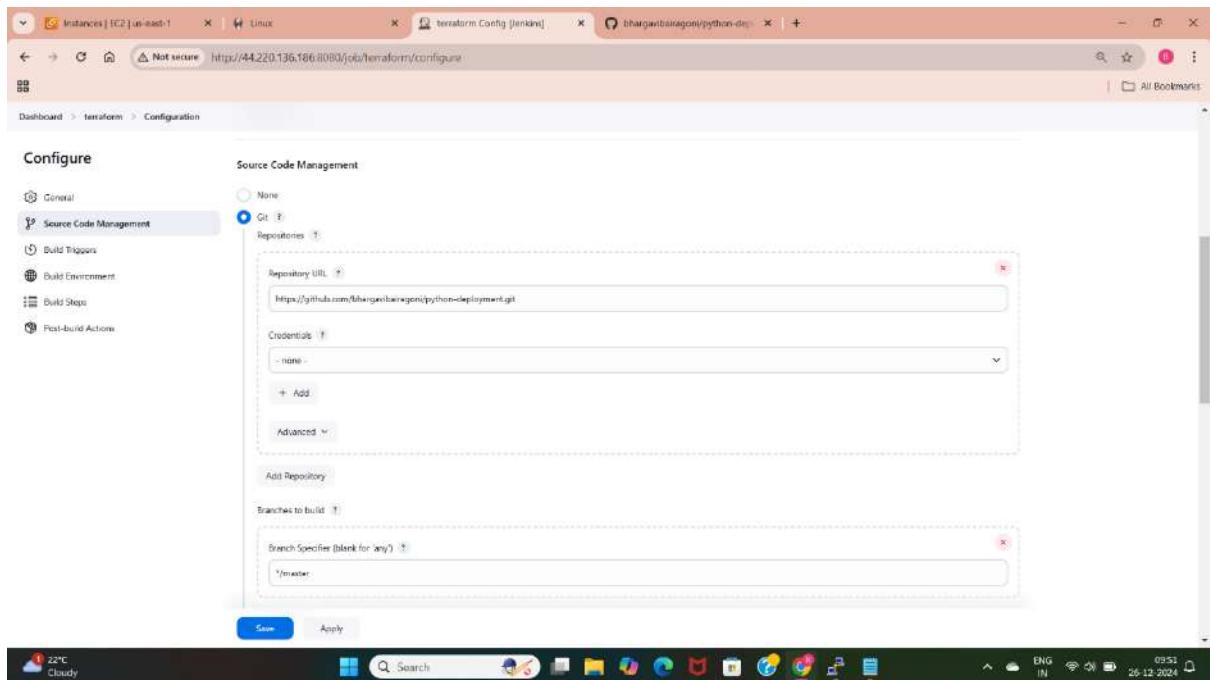
- ❖ Here am creating a user in Jenkins, by providing username, password, full name and email address.



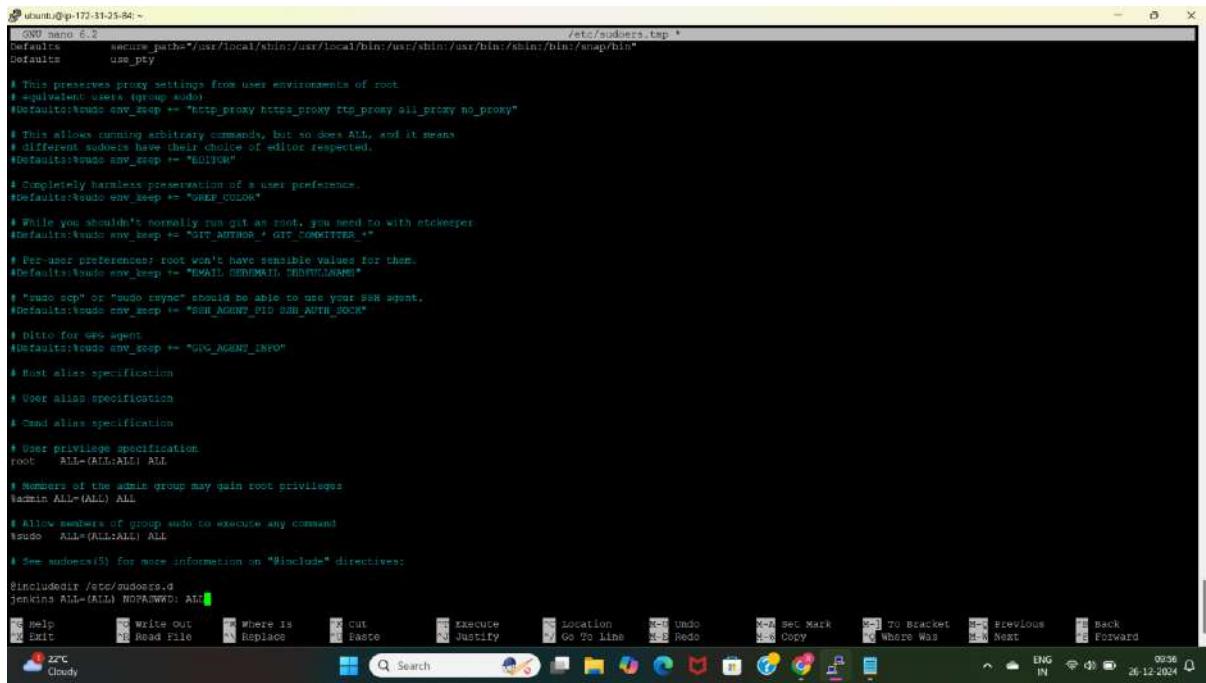
- ❖ This is the url of Jenkins which is the public IP address of instance with Jenkins port number.



- ❖ I have installed the “aws credentials” plugin by navigating to Manage Jenkins>plugins > available plugins.
- ❖ By navigating to credentials>global credentials > we can give access keys and secret access keys.



- ❖ Now go to Jenkins dashboard select new item and create job. Go inside the created job select ‘git’ and provide git url which is having the terraform files, and select the branch where we are having terraform files.

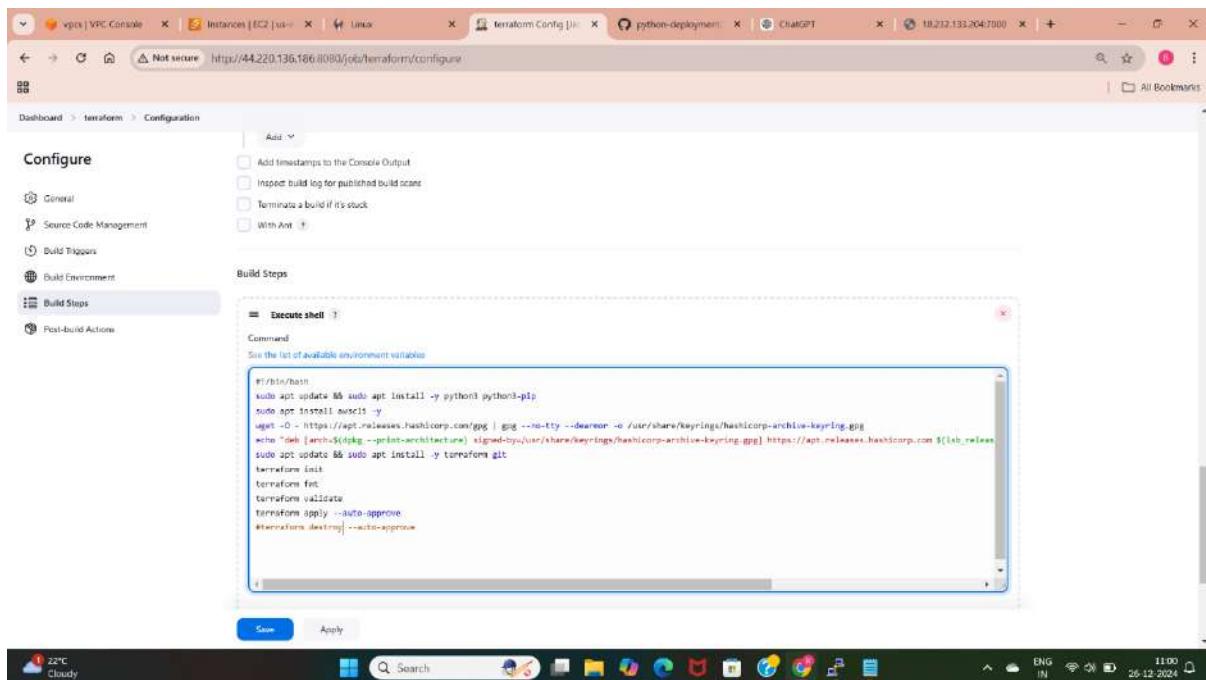


```

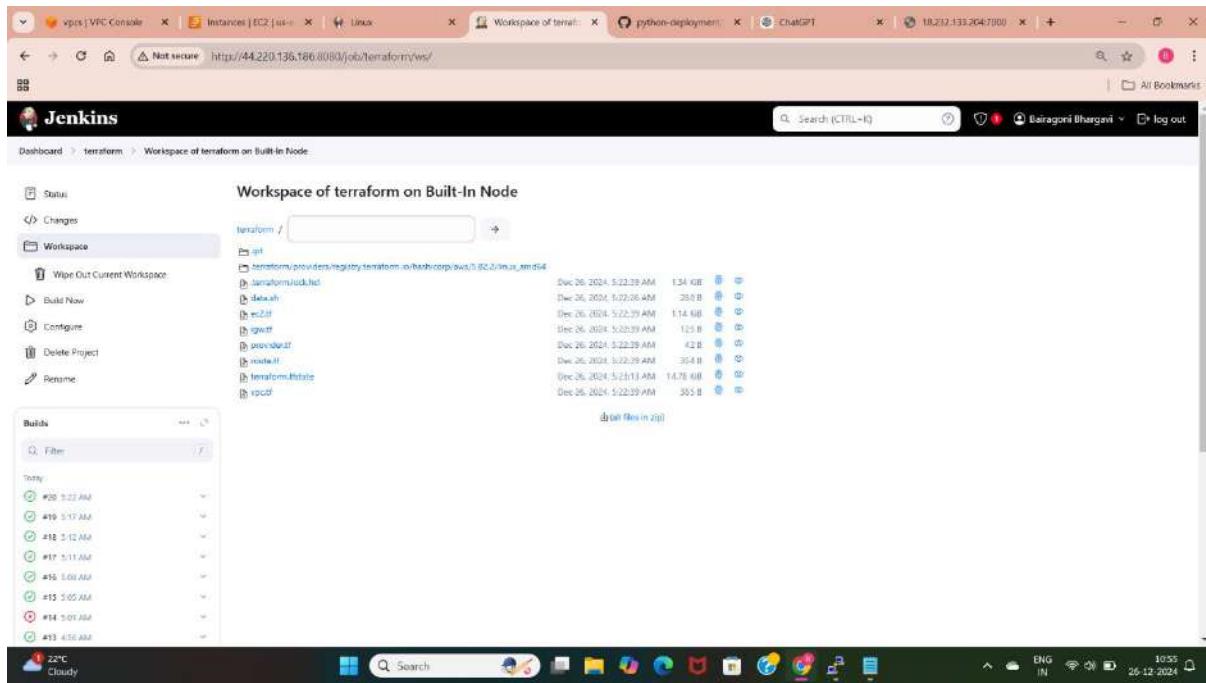
ubuntu@ip-172-31-25-84: ~
$ nano /etc/sudoers.d/jenkins
Defaults    secure_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults    use_pty
# This preserves proxy settings from user environments of root
# equivalent users (group sudo)
Defaults:root env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"
# This allows running arbitrary commands, but so does ALL, and it means
# different sudoers have their choice of editor respected.
Defaults:root env_keep += "EDITOR"
# Completely harmless preservation of a user preference.
Defaults:root env_keep += "GREP_COLOR"
# While you shouldn't normally run git as root, you need no with etckeeper
Defaults:root env_keep += "GIT_AUTHOR GIT_COMMITTER"
# Per-user preferences: root won't have sensible values for them.
Defaults:root env_keep += "EMAIL BENCHMARK BENCHMARKS"
# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
Defaults:root env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"
# Distro for GPG agent
Defaults:root env_keep += "GPG_AGENT_INFO"
# Host alias specification
# User alias specification
# Cmd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
# Members of the admin group may gain root privileges
%admin  ALL=(ALL) ALL
# Allow members of group sudo to execute any command
sudo   ALL=(ALL:ALL) ALL
# See sudoers(5) for more information on "#include" directives:
#includedir /etc/sudoers.d
jenkins ALL=(ALL) NOPASSWD: ALL

```

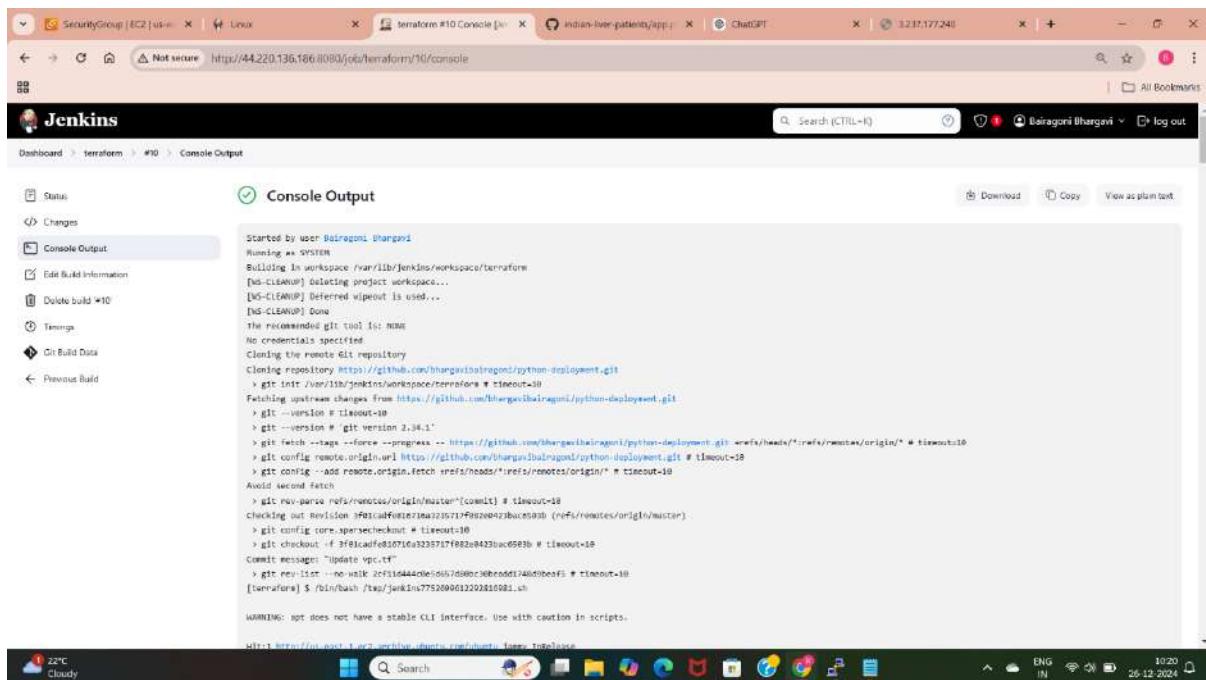
- ❖ Now go to terminal and give sudo permissions for Jenkins using “sudo visudo” ,and in that file provide the permissions using <Jenkins ALL=(ALL) NOPASSWD: ALL.



- ❖ Now in execute shell am providing the commands which installs terraform, python, pip, awscli in newly creating instance.
- ❖ Initialize the terraform in new instance, validate terraform files, apply the configurations in the new instance.



- ❖ Now go to the job and build the job using “build now”. Here we can see the build is successful and the terraform files stored in jenkins workspace.



- ❖ We can also check the process of build in the console output. Here we can see every step mentioned in the Jenkins.

```

[...]
  },
  "tags_all": [
    {
      "name": "Web Subnet 1"
    }
  ],
  "vpc_id": "vpc-04bb311361e8ff54e"
}
]
}

# terraform apply -auto-approve
[...]

```

- ❖ Here we can see the steps that after performing the terraform apply command we can see that a VPC is creating here.

Name	VPC ID	State	BLOCKED	IPv4 CIDR	IPv6 CIDR	DHCP option set	Main route table
DEMO VPC	vpc-04bb311361e8ff54e	Available	Off	172.31.0.0/16	-	dopt-0fa14cc7e4d0cb9e2	rtb-0be7de17f6620a5be

**vpc-04bb311361e8ff54e / DEMO VPC**

**Resource map**

- VPC Show details**: Your AWS virtual network
- Subnets (1)**: Subnets within this VPC
  - us-east-1a: Web Subnet 1
- Route tables (2)**: Router network traffic to resources
  - Route to Internet: rtb-02e62afcb8d8259d
- Network connections (0)**: Connections to other networks.

- ❖ We can check with the AWS console that Demo VPC is created.

The screenshot shows a browser window with multiple tabs open. The active tab is 'http://44.220.136.186:8080/jobs/terraform/20/console'. The content of the tab displays Terraform console output for a VPC configuration. The output shows the creation of various resources:

- A VPC with a specific CIDR block (10.0.0.0/16).
- A public subnet with a specific CIDR block (10.0.0.0/24).
- A private subnet with a specific CIDR block (10.0.1.0/24).
- A route table associated with the public subnet.
- A security group named 'DemoVPC'.
- A network interface card (NIC) for an EC2 instance.

The output concludes with a summary: "Apply complete! Resources: 7 added, 0 changed, 0 destroyed." and "0 errors".

❖ Here ec2 instance is creating with the security group.

The screenshot shows a browser window with multiple tabs open. The active tab is 'http://44.220.136.186:8080/jobs/terraform/10/console'. The content of the tab displays Terraform console output for a security group configuration. The output shows the creation of a security group named 'DEMO\_VPC' with the following rules:

- An ingress rule allowing traffic from 0.0.0.0/0 on port 80.
- An egress rule allowing traffic to 0.0.0.0/0 on port 80.

The output concludes with a summary: "Apply complete! Resources: 7 added, 0 changed, 0 destroyed." and "0 errors".

❖ Creating all the specified configurations is successful in the build.

The screenshot shows the AWS EC2 Instances page. A green banner at the top indicates "Successfully initiated termination (deletion) of i-05c81957a9833134d". Below this, a table lists two instances:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability zone
ter	i-097a8bf1...	Running	t2.medium	2/2 checks p	View alarms	us-east-1
<b>My public Instance</b>	i-0462fb062676c126f	Running	t2.micro	Initializing	View alarms	us-east-1

The instance **i-0462fb062676c126f (My public Instance)** is selected. The "Details" tab is active, showing the following information:

- Instance summary**:
  - Instance ID: i-0462fb062676c126f
  - Public IPv4 address: 18.232.133.204 | [open address](#)
  - IPv6 address: -
  - Instance state: Running
  - Private IP DNS name (IPv4 only): ip-10-0-1-192.ec2.internal
  - Answer private resource DNS name: -
  - Instance type: t2.micro
- Elastic IP addresses**:
  - Private IPv4 addresses: 10.0.1.192
  - Public IPv4 DNS: -

- ❖ Here we can see that the EC2 instance is created with the specified configurations in the terraform file.

The screenshot shows a web browser window displaying a form. The title of the page is "LIVER PATIENT". The form contains several input fields and a submit button. The input fields include:

- Female
- 
- 
- 
- 
- 
- 
- 

A blue "Submit!" button is located at the bottom of the form. The browser's address bar shows the URL <http://18.232.133.204:7000>.

- ❖ By copying the above created ec2 instance IP address we can access the python application.

## METHOD-7: BUILD AND DEPLOY PYTHON APPLICATIONS WITH DOCKER (WRITE DOCKER FILE AND CREATE IMAGE FROM DOCKER FILE AND RUN CONTAINERS) AND PUSH CREATED DOCKER IMAGE IN DOCKER HUB

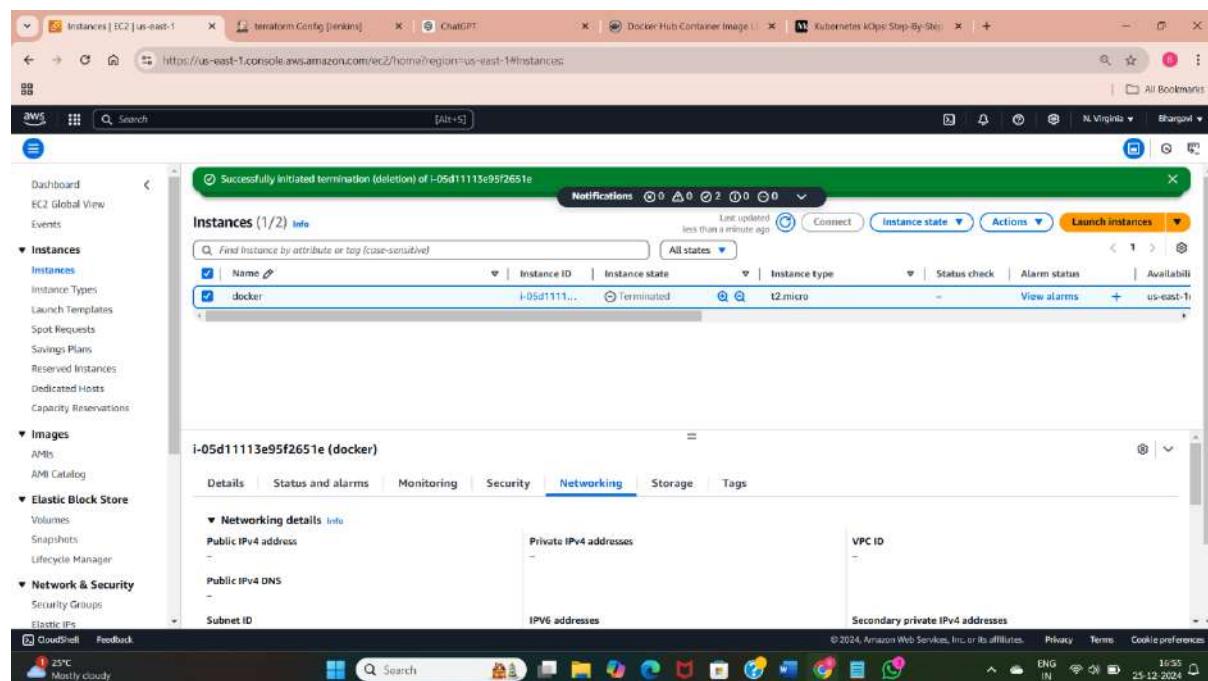
Build and deploy python applications with Docker (Write docker file and create image from docker file and run containers) and push created docker image in docker hub.

Docker is a tool that allows you to package applications and all their dependencies into lightweight, portable containers. These containers ensure that your application works consistently, no matter where it's running—on your computer, a server, or in the cloud.

A container is like a lightweight, standalone package that contains everything your app needs to run: code, libraries, tools, and settings.

A Dockerfile tells Docker what to do step by step, such as which base image to use, what files to add, and which commands to run.

When you use a Dockerfile, Docker creates an image, which can then be run as a container.



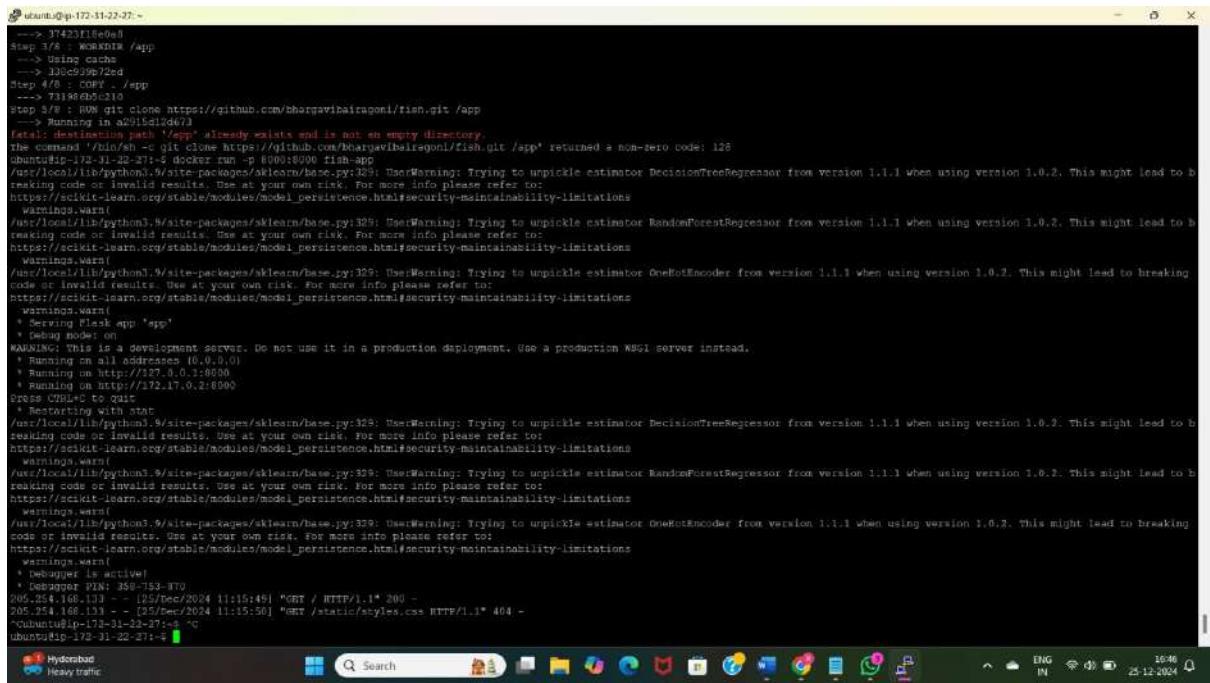
- ⦿ Creating a EC2 instance for deploying the python application with ubuntu os application and t2.micro instance type.
- ⦿ Write a docker file using “vim Dockerfile” and deploy the python application.

```

ubuntu@ip-172-31-94-50: ~
FROM python:3.9-slim
RUN apt-get update && apt-get install -y git && apt-get clean
WORKDIR /app
RUN git clone https://github.com/bhargavibairagoni/fish.git /app
RUN pip install -r requirements.txt
EXPOSE 8000
CMD ["python", "app.py"]
~

```

- ⦿ I have written the docker file to deploy fish python application. Here I have taken a python base image and updated the system, installed git using run command, moved working directory to /app.
- ⦿ I have downloaded the python application fish repository from the github, installed requirements.
- ⦿ I have used the port number 8000 for exposing the deployed application.
- ⦿ I have used the command to deploy the application that is “python app.py”.



```

ubuntu@ip-172-31-22-27: ~
--> 37423f1febed
Step 3/8 : WORKDIR /app
--> Using cache
--> 330c939e72ed
Step 4/8 : RUN git clone https://github.com/bhargavibairagoni/fish.git /app
--> T3198050e10
Step 5/8 : RUN git clone https://github.com/bhargavibairagoni/fish.git /app
--> Running in a2915d1ddc73
fatal: destination path '/app' already exists and is not an empty directory.
the command '/bin/sh -c git clone https://github.com/bhargavibairagoni/fish.git /app' returned a non-zero code: 128
ubuntu@ip-172-31-22-27:~$ docker run -p 8000:8000 fish-app
/usr/local/lib/python3.9/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
/usr/local/lib/python3.9/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
/usr/local/lib/python3.9/site-packages/sklearn/Base.py:329: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
* Serving flask app 'app'
* For stop flask app 'ctrl+c'
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://127.0.0.1:8000
* Running on http://172.17.0.2:8000
Press Ctrl+C to quit
* Restarting with stat
/usr/local/lib/python3.9/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator DecisionTreeRegressor from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
/usr/local/lib/python3.9/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator RandomForestRegressor from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
/usr/local/lib/python3.9/site-packages/sklearn/base.py:329: UserWarning: Trying to unpickle estimator OneHotEncoder from version 1.1.1 when using version 1.0.2. This might lead to breaking code or invalid results. Use at your own risk. For more info please refer to:
https://scikit-learn.org/stable/modules/model_persistence.html#security-maintainability-limitations
warnings.warn(
* No browser is active!
* Dockerport PORT=593-870
205.254.168.113 - [25/Dec/2024 11:15:46] "GET / HTTP/1.1" 200 -
205.254.168.113 - [25/Dec/2024 11:15:50] "GET /static/styles.css HTTP/1.1" 404 -
ubuntu@ip-172-31-22-27:~$ 

```

- ⦿ After writing docker file we need to build using “docker build -t fish .” and run the docker file using “docker run 8000:8000 fish” to deploy using the docker.
- ⦿ Here we can see the deployment is successful.

```

ubuntu@ip-172-31-22-27:~ 
34 docker build -t fish-app .
35 docker run -p 8000:8000 fish-app
36 vim Dockerfile
37 docker build -t fish-app .
38 docker run -p 8000:8000 fish-app
39 history
ubuntu@ip-172-31-22-27:~$ vim Dockerfile
ubuntu@ip-172-31-22-27:~$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: bharavvibaicagni
Password:
Error response from daemon: Get "https://registry-1.docker.io/v2/": unauthorized: incorrect username or password
ubuntu@ip-172-31-22-27:~$ docker login
Log in with your Docker ID or email address to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com/ to create one.
You can log in with your password or a Personal Access Token (PAT). Using a limited-scope PAT grants better security and is required for organizations using SSO. Learn more at https://docs.docker.com/go/access-tokens/
Username: bharavvibaicagni
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
ubuntu@ip-172-31-22-27:~$ docker tag bharavvibaicagni/fish-app:python-deployment
"docker tag" requires exactly 2 arguments.
See 'docker tag --help'.

Usage: docker tag SOURCE_IMAGE[:TAG] TARGET_IMAGE[:TAG]

Create a tag TARGET_IMAGE that refers to SOURCE_IMAGE
ubuntu@ip-172-31-22-27:~$ cd
ubuntu@ip-172-31-22-27:~$ docker tag fish-app bharavvibaicagni/fish-app:python-deployment
ubuntu@ip-172-31-22-27:~$ docker push bharavvibaicagni/fish-app:python-deployment
The push refers to repository [docker.io/bharavvibaicagni/fish-app]
3b1501993a40: Pushed
e2894403e5bd: Pushed
0f0d0d050c11: Pushed
f980000f6e15: Pushed
c9393019f641: Merged from library/python
313239303553: Merged from library/python
4cc45f469460: Merged from library/python
python-deployment: digest: sha256:7c92267e8c80d289f044699b13bcfe101707993b7iac33b6223c96810fc9f=7d size: 2003
ubuntu@ip-172-31-22-27:~$ 

```

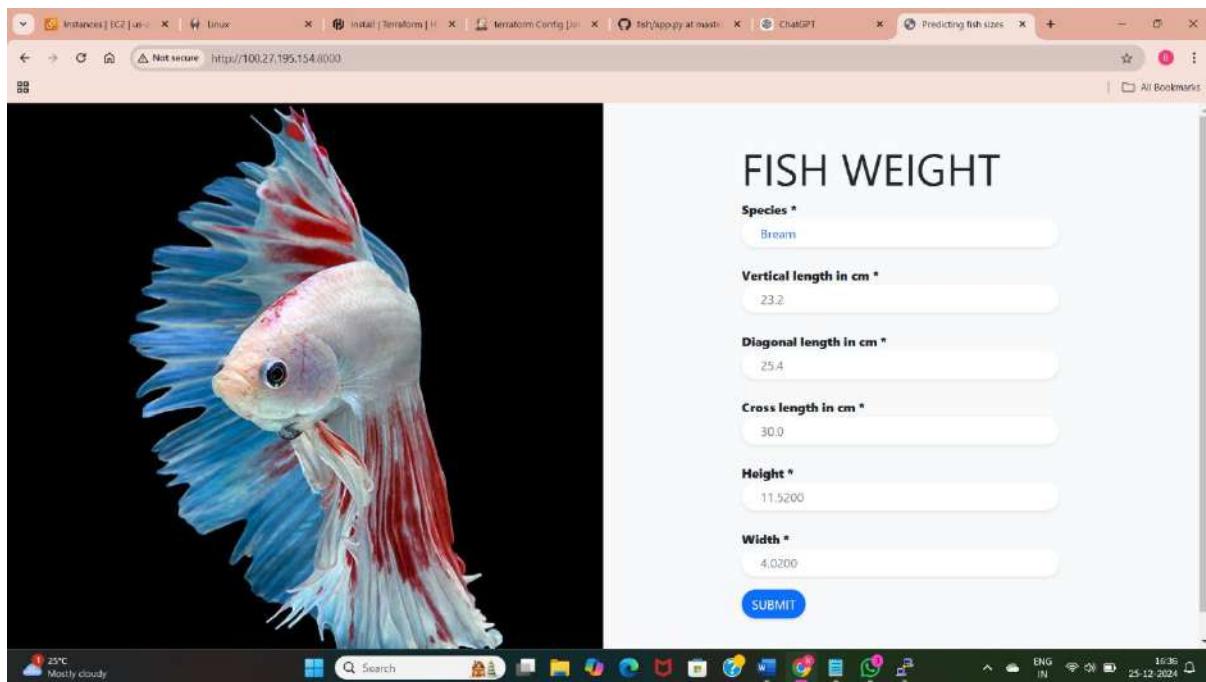
- After running the image we need to push that image to docker hub using the docker tag and docker push.

```

ubuntu@ip-172-31-22-27:~ 
code or invalid results. Use at your own risk. For more info please refer to:
https://xkcd.com/2294/modules/model_persistence.html#security-maintainability-limitations
warning.wasm
1 docker ps -a
2 docker ps -a
3 Dockerfile active!
4 docker build -t fish-app .
5 docker run -p 8000:8000 fish-app
6 docker run -p 8000:8000 fish-app
7 docker run -p 8000:8000 fish-app
8 vim Dockerfile
9 docker build -t fish-app .
10 docker run -p 8000:8000 fish-app
11 docker ps -a
12 docker ps -a
13 docker start 8208872397c7
14 docker logs 8208872397c7
15 python manage.py runserver 0.0.0.0:8000
16 python3 manage.py runserver 0.0.0.0:8000
17 docker run -it fish-app /bin/bash
18 vim Dockerfile
19 docker build -t fish-app .
20 docker run -p 8000:8000 fish-app
21 docker ps -a
22 docker ps -a
23 docker rm 31dockers ps -a -q
24 docker run -it fish-app .
25 docker run -p 8000:8000 fish-app
26 docker ps -a
27 docker rm 31dockers ps -a -q
28 docker stop vigorous_burnell
29 docker rm vigorous_burnell
30 docker ps -a
31 docker build -t fish-app .
32 docker run -p 8000:8000 fish-app
33 vim Dockerfile
34 docker build -t fish-app .
35 docker run -p 8000:8000 fish-app
36 vim Dockerfile
37 docker build -t fish-app .
38 docker run -p 8000:8000 fish-app
39 history
ubuntu@ip-172-31-22-27:~$ 

```

- These are the commands I have used in the docker to create a python application image, running the container.

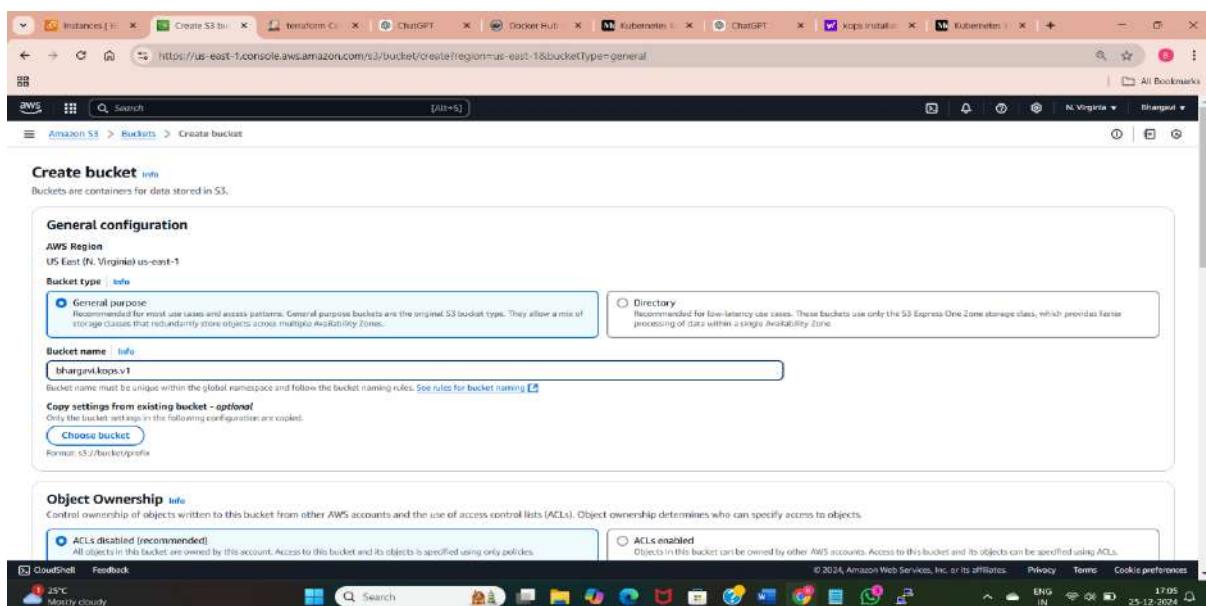


- After running the docker image we need to use the public IP address of the instance with the port which we used in the expose and browse we can see the application.

## METHOD-8: BUILD AND DEPLOY PYTHON APPLICATIONS WITH DOCKER AND K8'S (EKS AND KOPS) (USE DECLARATIVE MANIFEST METHOD ALONG WITH DOCKER IMAGE)

Kubernetes is an open-source platform for automating the deployment, scaling, and management of containerized applications. It helps you manage containers (like Docker containers) across a cluster of machines, providing features like load balancing, self-healing, and scaling, ensuring that your applications run reliably and efficiently in a production environment.

- To setup Kubernetes first we need to launch ec2 instance and install kops and kubectl in the instance by connecting it to terminal.



- For Kubernetes we need to setup s3 bucket.

- ♣ Create a general purpose s3 bucket with unique name.

- ♣ Here I have connected the instance to terminal and installed kubectl and kops using the above commands.
  - ♣ Give the aws credentials to creates the clusters in the aws.
  - ♣ Export the created s3 bucket.

- Here am creating the kops clusters using “`kops create cluster --name bhargavi.k8s.local --zones us-east-1a --master-size t2.medium --node-size t2.micro --node-count=1`”
  - Update the cluster using “`kops update cluster --name bhargavi.k8s.local --yes --admin`”.
  - We can check with the created clusters using “`kubectl get nodes`”.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability
kops	i-05d5efac...	Running	t2.micro	2/2 checks p	View alarms	us-east-1
ter	i-097a8bf1...	Terminated	t2.medium	-	View alarms	us-east-1
nodes-us-east-1a.bhargavibairagoni.k8s.local	i-01ae64c5...	Running	t2.micro	2/2 checks p	View alarms	us-east-1
control-plane-us-east-1a.masters.bhargavibairagoni.k8s.local	i-0d7a1878...	Running	t2.medium	2/2 checks p	View alarms	us-east-1

- We can see in the AWS management console two clusters are created one is node and other is control plane.

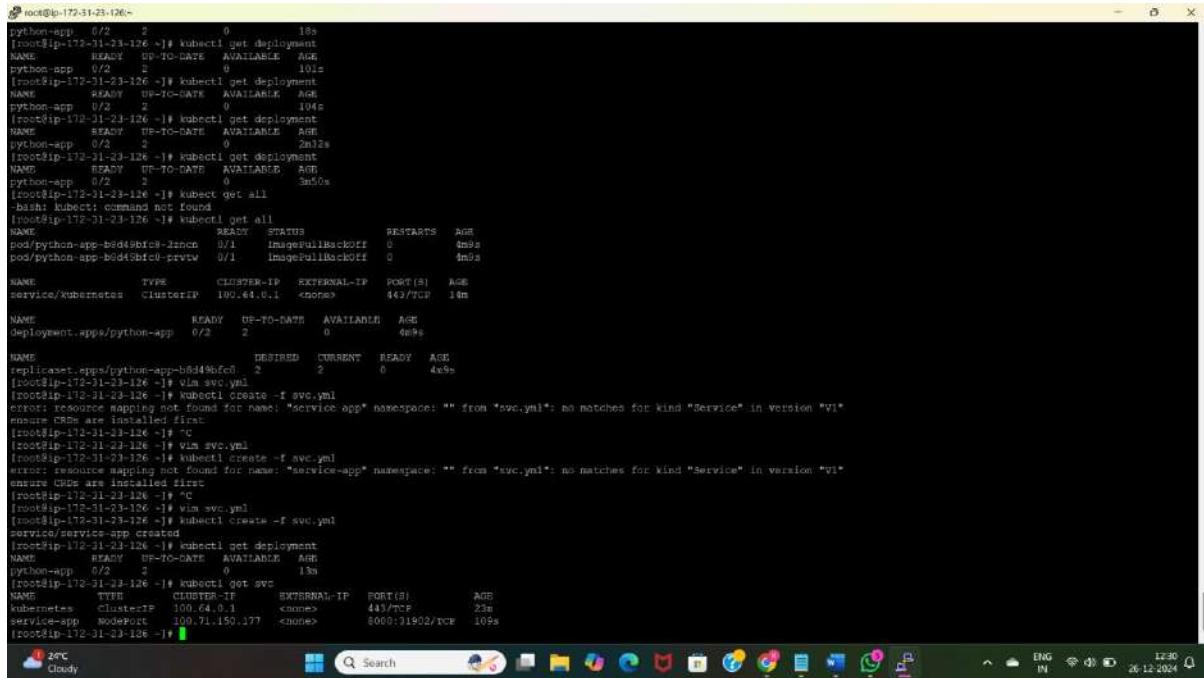
```
root@ip-172-31-29-216:~#
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fish-deployment
spec:
  replicas: 2
  selector:
    matchLabels:
      tier: python-app
  template:
    metadata:
      labels:
        tier: python-app
    spec:
      containers:
      - name: container
        image: bhargavibairagoni/fish-app:python-deployment
      ports:
      - containerPort: 80
```

- Create a deployment file with the application image.
- In deployment file am creating a pod/container with the name “python-app”.
- Specifying the replicas so that particular number of pods are available for every time.
- Give the container port.
- Now run the deployment file using “kubectl create -f deploymentfile name” or “kubectl apply -f deploymentfilename”.



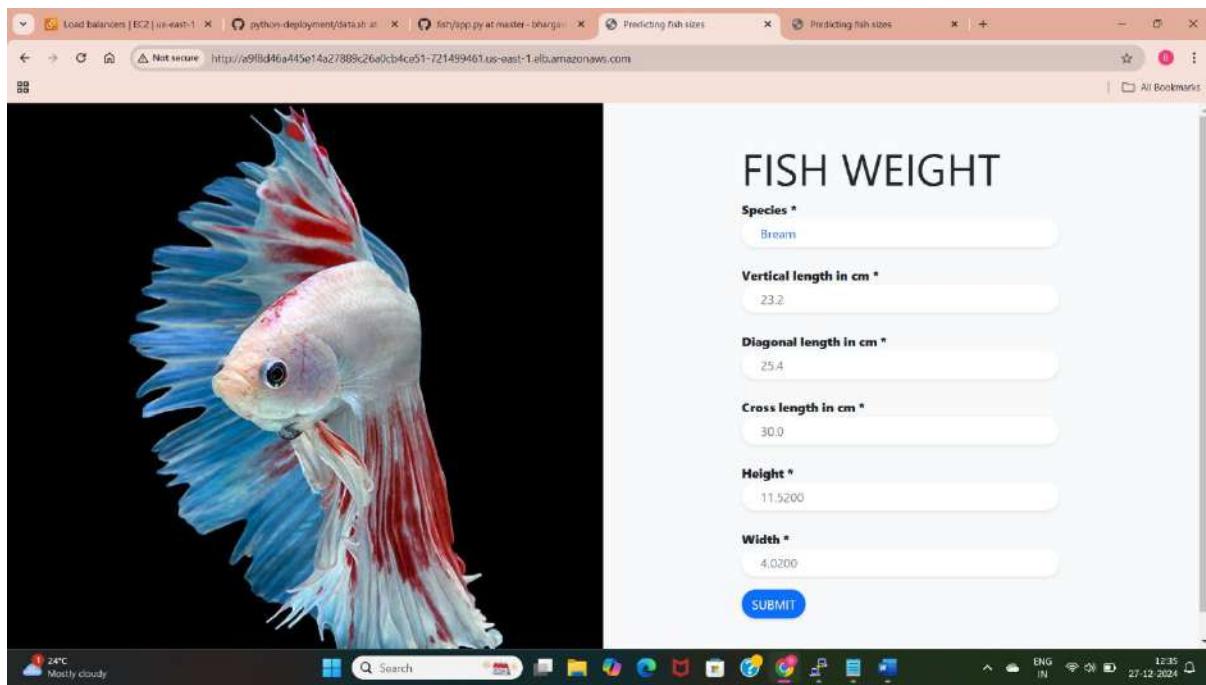
```
root@ip-172-31-29-216:~  
  
apiVersion: v1  
kind: Service  
metadata:  
  name: fish-service  
spec:  
  selector:  
    tier: python-app  
  ports:  
    - protocol: TCP  
      port: 80  
      targetPort: 8000  
  type: LoadBalancer  
~  
~
```

- ♣ Creating a service file with the loadbalancer type and provide the pod name which is used in the deployment, give the target port which matches the application port.
- ♣ We can run the service file using “kubectl create -f servicefilename” or “kubectl apply -f servicefilename”.



```
root@ip-172-31-23-126:~  
python-app 0/2 2 0 18s  
[root@ip-172-31-23-126 ~]# kubectl get deployment  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
python-app     0/2     2           0           18s  
[root@ip-172-31-23-126 ~]# kubectl get deployment  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
python-app     0/2     2           0           194s  
[root@ip-172-31-23-126 ~]# kubectl get deployment  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
python-app     0/2     2           0           2m12s  
[root@ip-172-31-23-126 ~]# kubectl get deployment  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
python-app     0/2     2           0           3m50s  
[root@ip-172-31-23-126 ~]# kubectl get all  
-bash: kubectl: command not found  
[root@ip-172-31-23-126 ~]# kubectl get all  
NAME          READY   STATUS    RESTARTS   AGE  
pod/python-app-b8d49fbfc8-2nnen 0/1   ImagePullBackOff  0   3m6s  
pod/python-app-b8d49fbfc0-prtvw 0/1   ImagePullBackOff  0   3m6s  
  
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE  
service/kubernetes  ClusterIP  109.64.0.1   <none>       443/TCP   3m  
  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
deployment.apps/python-app  0/2     2           0           3m9s  
  
NAME          DESIRED  CURRENT  READY   AGE  
replicaset.apps/python-app-b8d49fbfc0  2     2     0     4s9s  
[root@ip-172-31-23-126 ~]# vim svc.yml  
[root@ip-172-31-23-126 ~]# kubectl create -f svc.yml  
error: resource mapping not found for name: "service/app" namespace: "" from "svc.yml": no matches for kind "Service" in version "v1"  
ensure CRD(s) are installed first  
[root@ip-172-31-23-126 ~]# ^C  
[root@ip-172-31-23-126 ~]# vim svc.yml  
[root@ip-172-31-23-126 ~]# kubectl create -f svc.yml  
error: resource mapping not found for name: "service/app" namespace: "" from "svc.yml": no matches for kind "Service" in version "v1"  
ensure CRD(s) are installed first  
[root@ip-172-31-23-126 ~]# ^C  
[root@ip-172-31-23-126 ~]# vim svc.yml  
[root@ip-172-31-23-126 ~]# kubectl create -f svc.yml  
service/service created  
[root@ip-172-31-23-126 ~]# kubectl get deployment  
NAME          READY   UP-TO-DATE   AVAILABLE   AGE  
python-app     0/2     2           0           13s  
[root@ip-172-31-23-126 ~]# kubectl get svc  
NAME          TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)   AGE  
kubernetes  ClusterIP  109.64.0.1   <none>       443/TCP   23s  
service-app  NodePort   100.71.150.177  <none>       8000:31902/TCP  109s  
[root@ip-172-31-23-126 ~]
```

- ♣ To check the created deployments use “kubectl get deployments”.
- ♣ To check the created pods use “kubectl get pods”.
- ♣ To check the created services use “kubectl get svc”.
- ♣ To check all these at a time use “kubectl get all”.

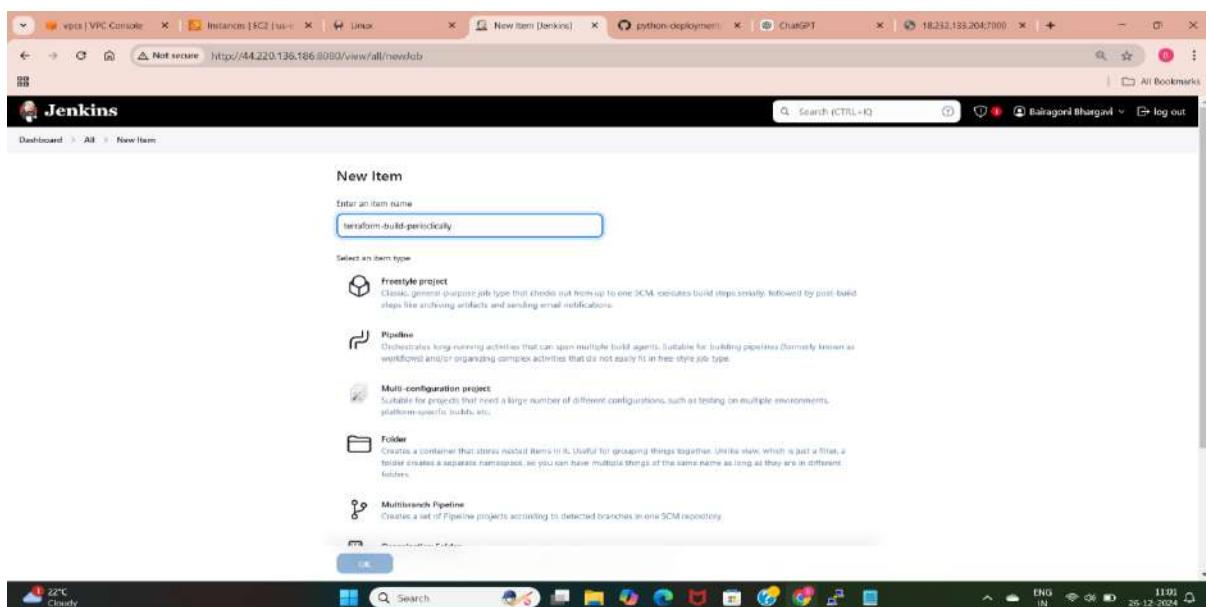


- This is the python application deployed using k8s with the docker image. We can access this application using the load balancer url.

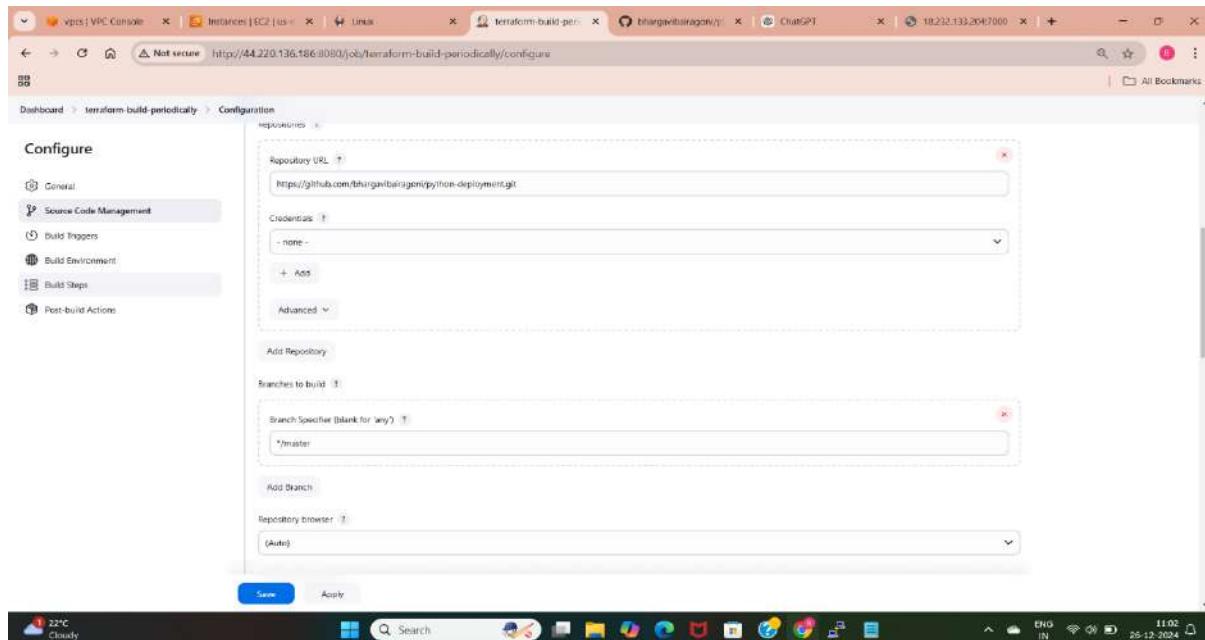
## METHOD-9: BUILD AND DEPLOY PYTHON APPLICATIONS WITH THE GIT, GITHUB, JENKINS AND TERRAFORM (WITH BUILD PERIODICALLY, POLLSCM AND WEBHOOKS)

Terraform allows you to write code to define the infrastructure you need, and then it automatically takes care of creating, updating, and managing the resources on cloud platforms like AWS, Azure, Google Cloud, or even on-premises servers.

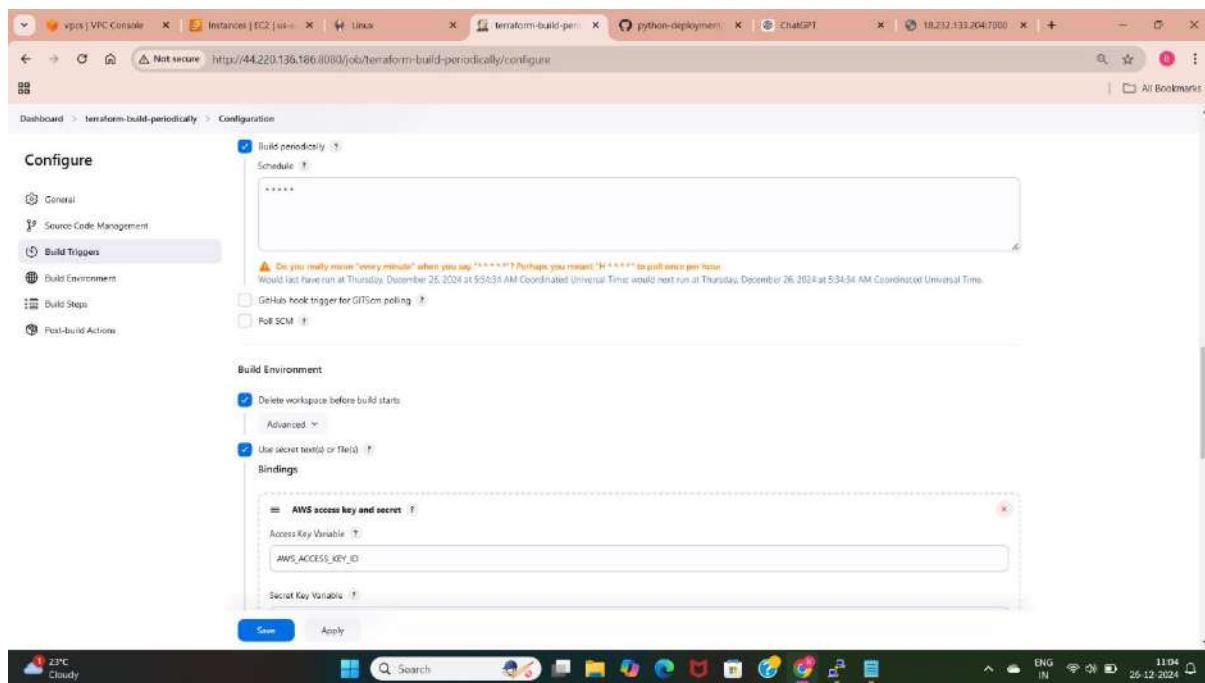
To deploy python application using git, github, Jenkins and terraform first we need to launch a EC2 instance and setup Jenkins in that instance by connecting it to the terminal, and install git.



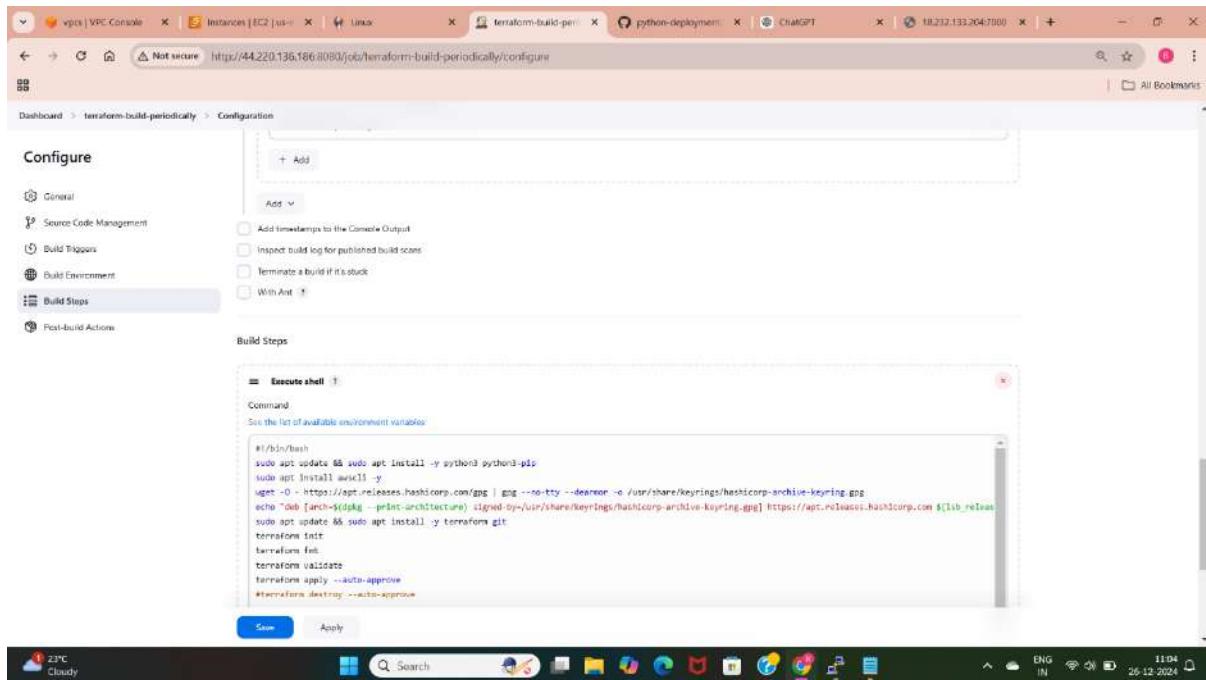
- ❖ Now create a Jenkins job for deploying the terraform files. Give the name of the job, and select freestyle project.



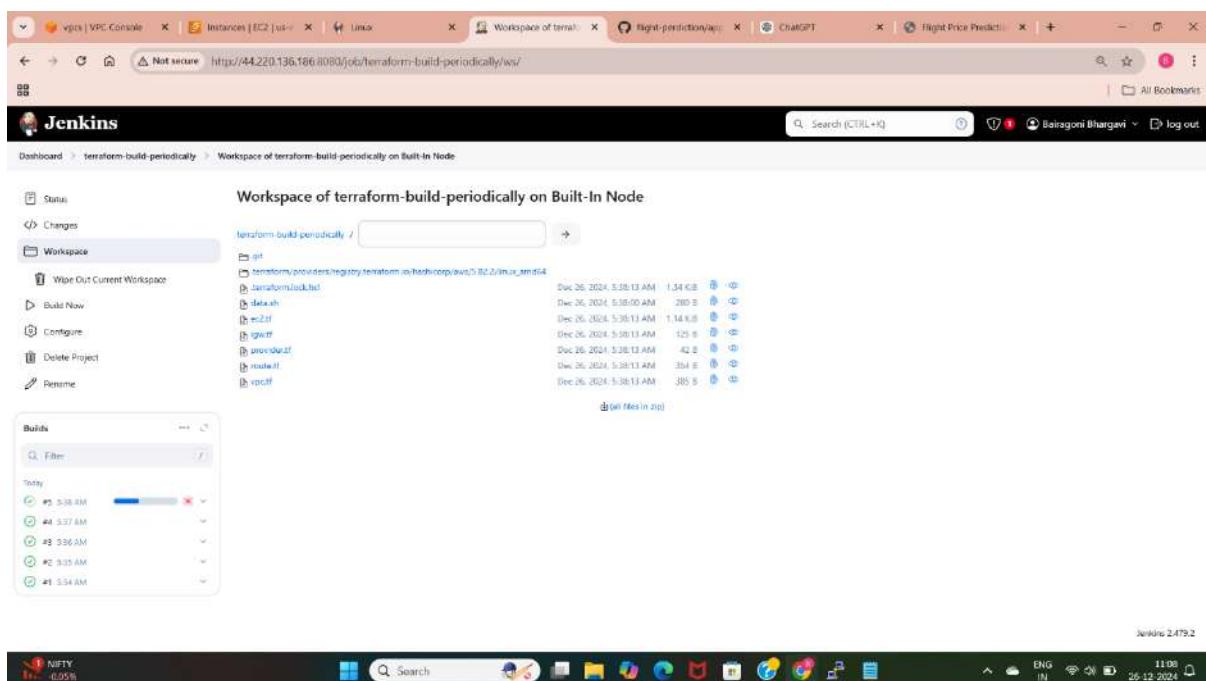
- ❖ Select the “git” and take the terraform files from github using the url, select the branch where we are having the files.



- ❖ Select the “build periodically” option and give the crontab syntax according to the requirement.
- ❖ Build periodically is used to build the jobs irrespective of changes in the code.
- ❖ I have given the build periodically syntax “\* \* \* \* \*” which builds the job for every minute.
- ❖ Select “delete workspace before build starts” this option is used to delete the files which are downloaded from previous build.
- ❖ Give the AWS credentials using the “secret text or file”.



- ❖ Now , select build steps > execute shell.
- ❖ In execute shell give the commands to install python, pip, awscli and terraform.
- ❖ I have used the commands “terraform init “ in the directory where terraform files are cloned.
- ❖ “terraform fmt” to provide the indentation for the terraform files.
- ❖ “terraform validate” to check with the code in the terraform files.
- ❖ “terraform apply “ for applying the changes that we have done in configuration files of terraform and select save button.



- ❖ Now select “Build Now” option. We will get either “green right symbol” for successful build, and for failure build we will get “cross mark in red color”.
- ❖ We can see that the above builds are successful.
- ❖ We can also check with the builds, that they are running for every minute according to “build periodically”.

```

Dashboard > terraform-build-periodically > #4 > Console Output

[{"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 7 to add, 0 to change, 0 to destroy."}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 7 resources: Creating: 8[imPlan:2]m"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[vpc]: vpc: Creation complete after 1s [id=vpc-05ad037c51f4c72]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[subnet]: subnet:1: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[internet_gateway]: internet_gateway: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[security_group]: security_group: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[internet_gateway]: internet_gateway: creation complete after 0s [id=igw-00be000000042a879]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[route_table]: route_table: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[route_table]: route_table: creation complete after 0s [id=rta-044f8851ba382da5]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[security_group]: security_group: creation complete after 2s [id=sg-0c10f870aa0620]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[subnet]: subnet:1: creating... [id=subnet-04270d9858f18a582]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[volume]: volume: public_subnet-1: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[route_table_association]: route_table_association: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[instance_public_subnet]: instance_public_subnet:1@#8: creating...@#8@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[instance_public_subnet]: instance_public_subnet:1@#8: creation complete after 12s [id=i-0cb942a4cd0ca050]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[instance]: instance:1@#8: creation complete after 12s [id=i-0cb942a4cd0ca050]@#8"}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[apply]: Apply complete! Resources: 7 added, 0 changed, 0 destroyed."}, {"id": "2024-01-26T10:00:00Z", "label": "Resource Changes", "text": "2[imPlan:2]m 8[finished]: SUCCESS"}]

```

- ❖ We can check with the console output that the terraform files are running with creating vpc, ec2 instance, route table, internet gateway and subnet.

Name	Instance ID	Instance State	Instance Type	Status Check	Alarm Status	Availability Zone
ter	i-097a80f1...	Running	t2.medium	2/2 checks pass	View alarms	us-east-1c
My public Instance	i-0c42ff05...	Running	t2.micro	Initializing	View alarms	us-east-1a
<b>My public Instance</b>	<b>i-0cb942a4...</b>	<b>Running</b>	<b>t2.micro</b>	<b>Initializing</b>	<b>View alarms</b>	<b>us-east-1a</b>

- ❖ After running above terraform files a new ec2 instance with ubuntu os, t2.micro instance type is created.
- ❖ For every minute a new EC2 instance is created.

The screenshot shows the AWS VPC Dashboard with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#vpcs>. The dashboard lists four VPCs under the heading "Your VPCs (4/5) Info". The columns include Name, VPC ID, State, Block, IPv4 CIDR, IPv6 CIDR, DHCP option set, and Main route table. All VPCs are in an "Available" state with "Off" block settings. The VPC IDs are vpc-05773c95ed8706096, vpc-0aac5a4fb09cd106, vpc-033574884c95b1190, and vpc-05ade4375c2f4c72. The main route table for each is rtb-032710d20ff5d1baef, rtb-081ab40b0bc97f9c5, rtb-0292a519d4625f1f5, and rtb-036997889871e421 respectively.

- ❖ New VPC's are creating according to build periodically. Build periodically is running the terraform files for every minute.

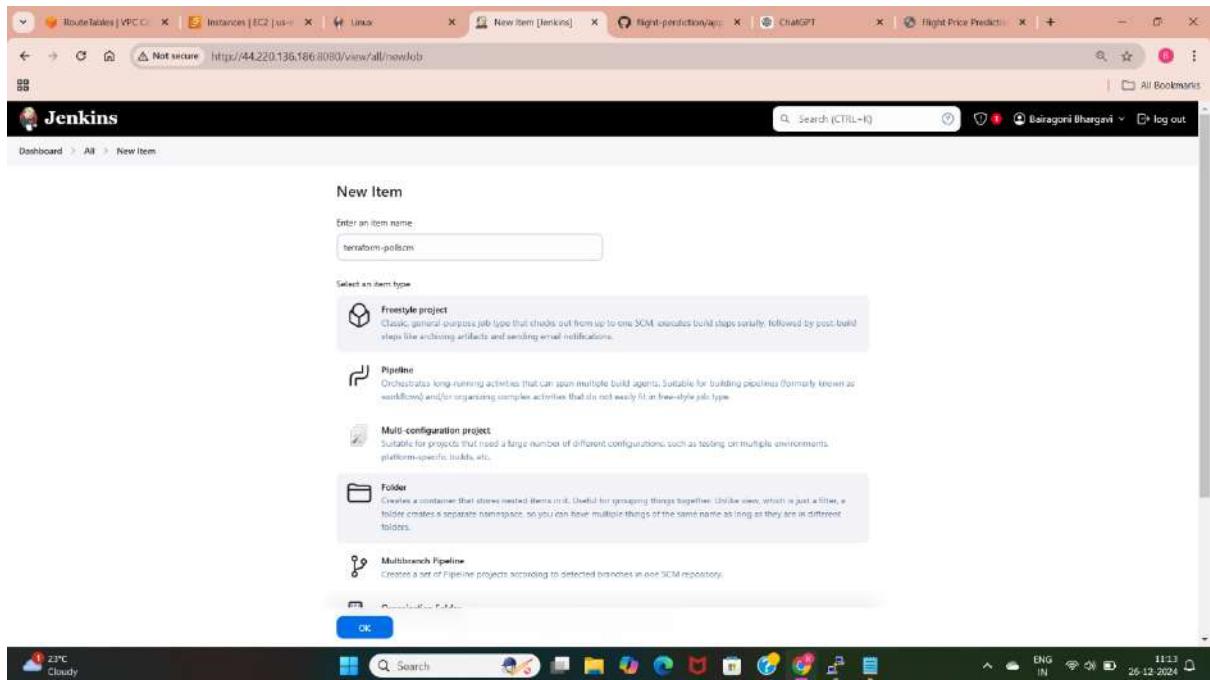
The screenshot shows the AWS Subnets Dashboard with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#subnets>. The dashboard lists ten subnets under the heading "Subnets (10) Info". The columns include Name, Subnet ID, State, VPC, IPv4 CIDR, IPv6 CIDR, IPv6 CIDR association ID, and Available IPv4 address space. The subnets are categorized under "Web Subnet 1". The subnets are: subnet-01657fa4557..., subnet-0a0407be5e37..., subnet-0bf503e5b007..., subnet-0bb6e1e4ec93..., subnet-0307e1f5f61d..., subnet-09e5e5856f..., subnet-0875dbfcdb8..., subnet-0427039858f1..., subnet-0bd0d55fcf67..., and subnet-02184702857.... All subnets are in an "Available" state. The VPC for all subnets is vpc-03f5f1. The IPv4 CIDRs range from 172.51.48.0/20 to 172.51.32.0/20. The IPv6 CIDR association ID and Available IPv4 address space are also listed.

- ❖ Subnets are creating when we get the build is successful.
- ❖ Here four subnets are created with the builds.

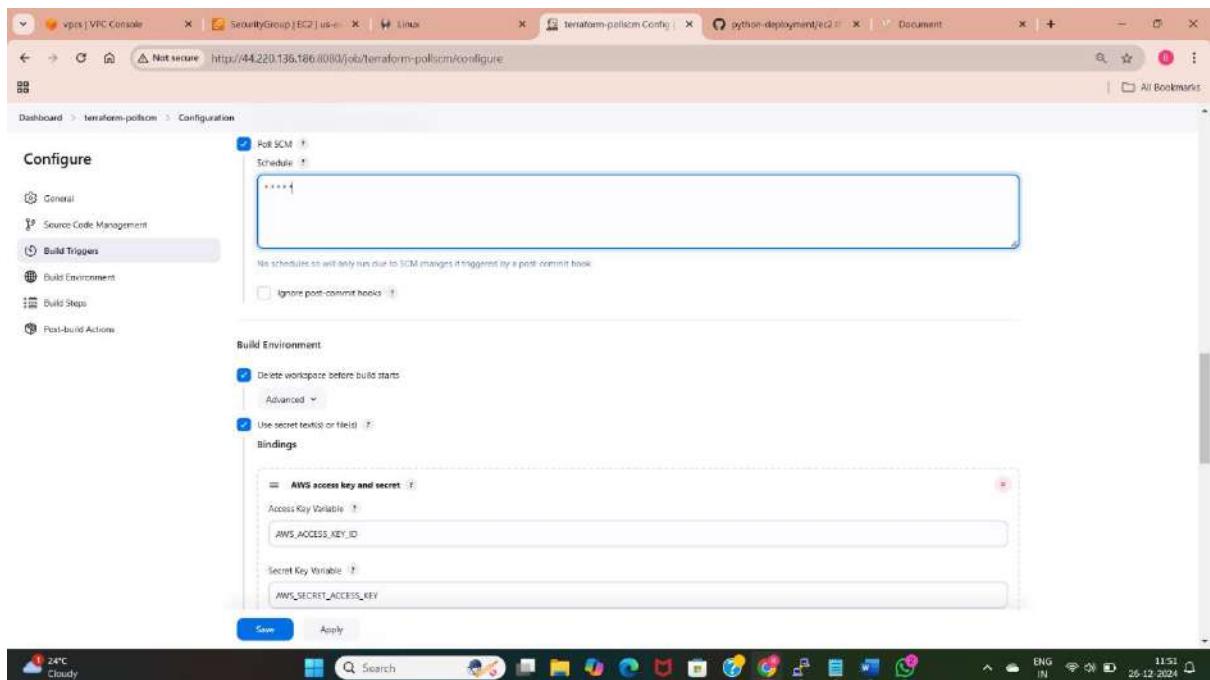
Name	Route table ID	Explanation	Edge associations	Main	VPC	Owner ID
Route to Internet	rtb-0a208258ef72cd...	subnet...	-	No	vpc-0bf21e4fc0a508ebf6	90541b365089
-	rtb-0b17de17ff670a...	-	-	Yes	vpc-0f5fe50c55d01103	90541b365089
-	rtb-0c934df62a1721e...	-	-	Yes	vpc-0bf21e4fc0a508ebf6	90541b365089
Route to Internet	rtb-0cb525baaf5f...	subnet...	-	No	vpc-031ad75c24129ed56	90541b365089
-	rtb-04981ce3f94b2a...	-	-	Yes	vpc-051ad75c24129ed56	90541b365089
-	rtb-052710d2fc5d1...	-	-	Yes	vpc-05773c95ed8706096   DEMO VPC	90541b365089
Route to Internet	rtb-02262f5c1ef19...	subnet...	-	No	vpc-0533748849561190   DEMO VPC	90541b365089
-	rtb-08149a400dc93f7...	-	-	Yes	vpc-0aac5a4fb0c95d106   DEMO VPC	90541b365089
Route to Internet	rtb-0449805bb2a3828...	subnet...	-	No	vpc-05ade437c5c214c72   DEMO VPC	90541b365089
-	rtb-0359971889871e...	-	-	Yes	vpc-05ade437c5c214c72   DEMO VPC	90541b365089
Route to Internet	rtb-022f101cbed00...	subnet...	-	No	vpc-0aac5a4fb0c95d106   DEMO VPC	90541b365089
-	rtb-0292433d4625f...	-	-	Yes	vpc-0333748849561190   DEMO VPC	90541b365089
Route to Internet	rtb-07915ca81e3e259...	subnet...	-	No	vpc-05773c95ed8706096   DEMO VPC	90541b365089

- ❖ Route table is created with the provided terraform configuration.

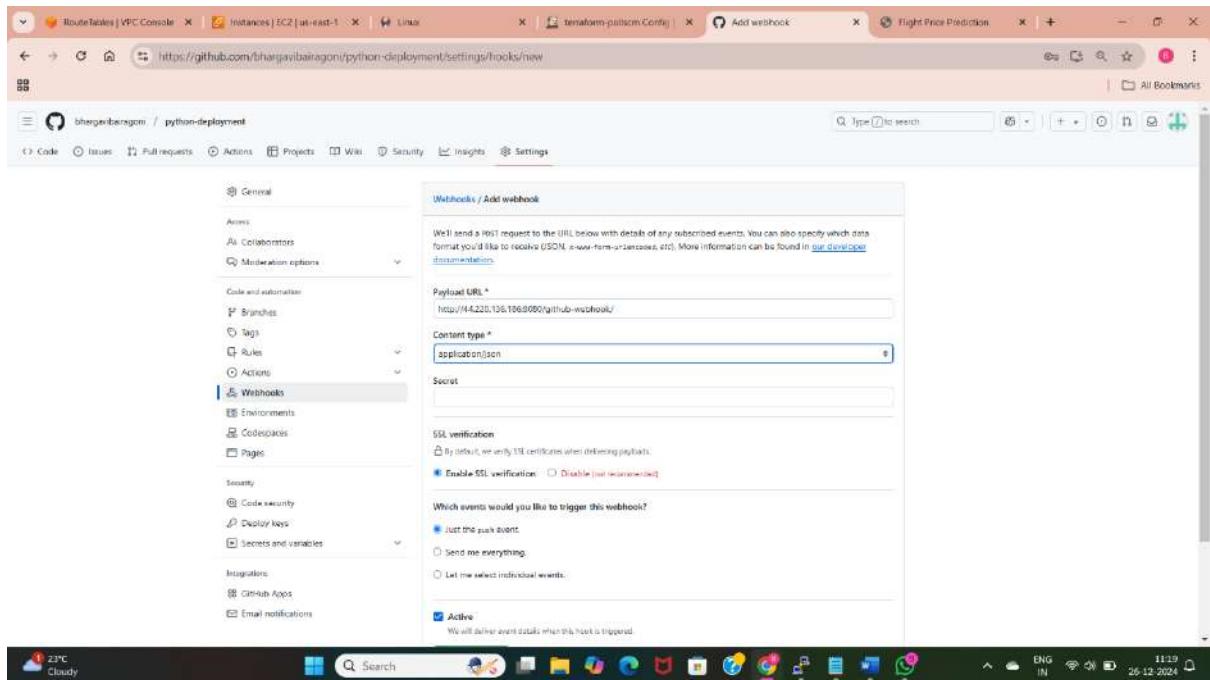
- ❖ After taking the Public IP address of the newly created instance with application port we can able to access the application.
- ❖ Here I have deployed application using build periodically.



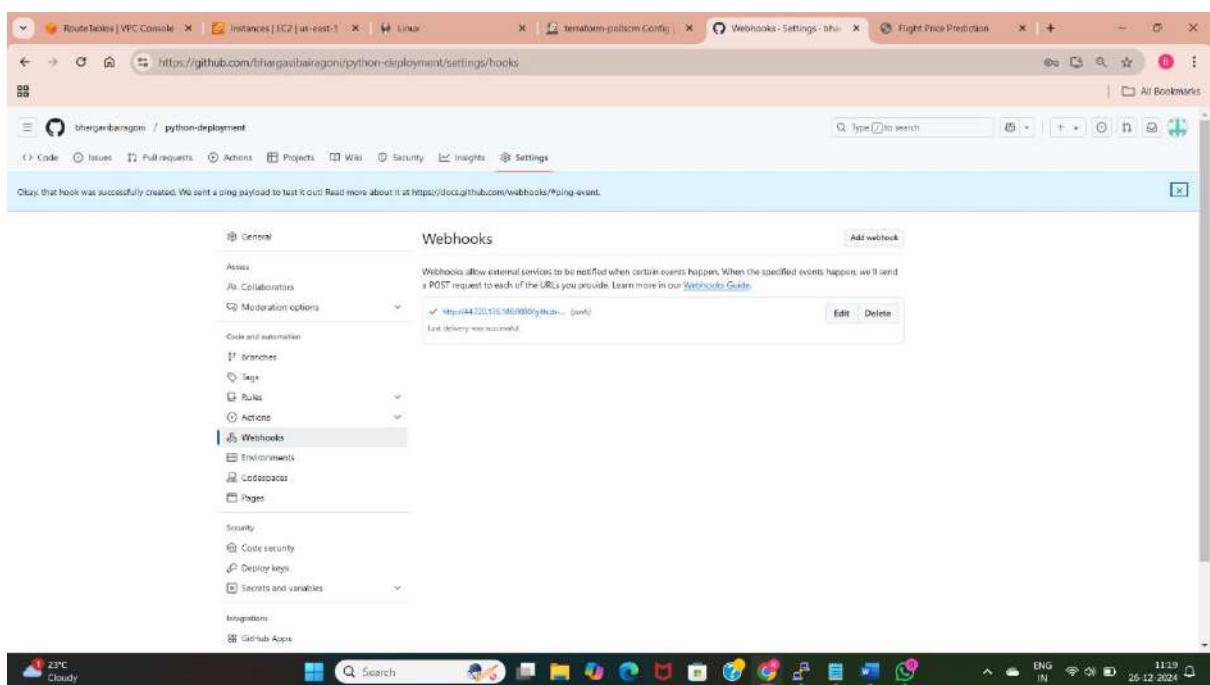
- ❖ Now create a new item with the name “terraform-pollscm”, select the freestyle project.



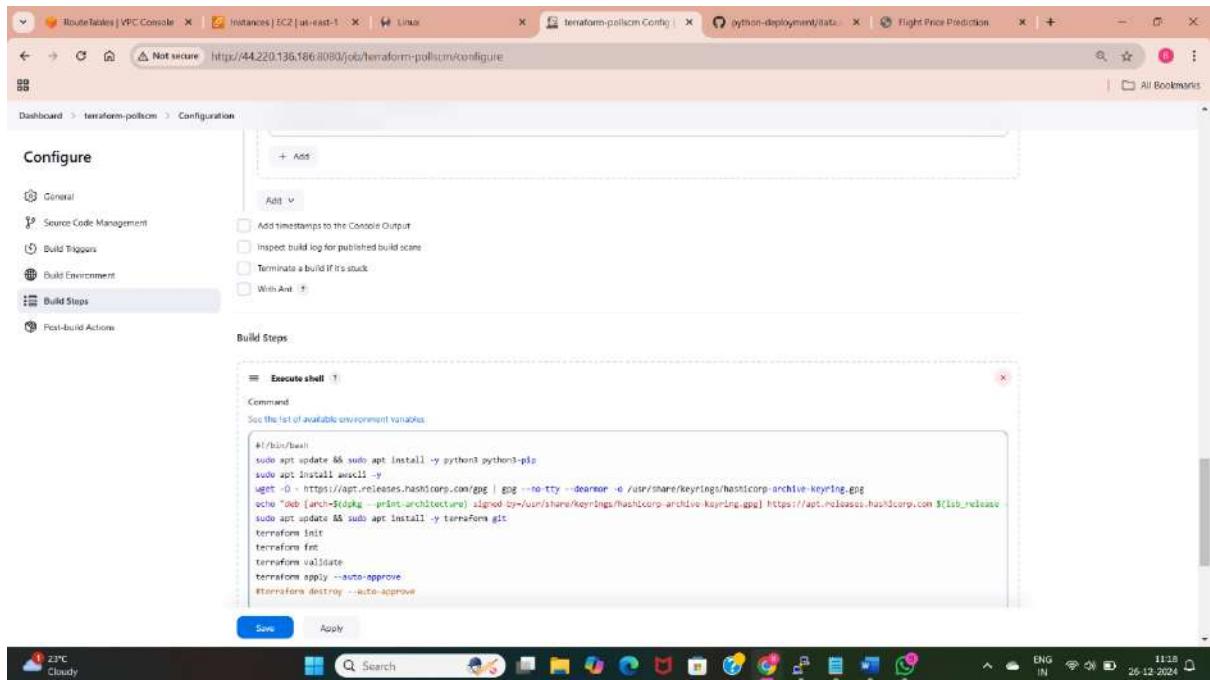
- ❖ Select the “poll SCM” option and give the crontab syntax. I have given “\* \* \* \* \*” syntax for building the code.
- ❖ Poll SCM is used to trigger the builds when we change in github code.
- ❖ Here first we need to create the webhook in the github.



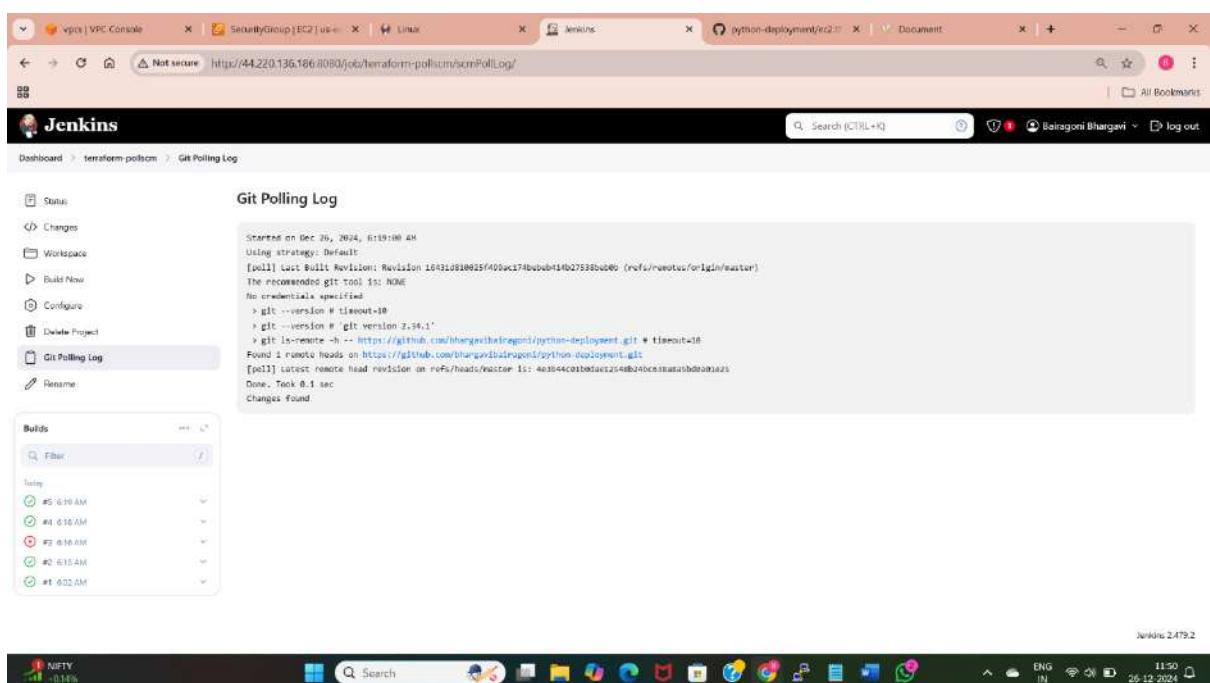
- ❖ Now navigate to github, go inside the repository which you are using in the Jenkins, select the “setting” of repository.
- ❖ Select “webhook” option, select add webhook.
- ❖ At the place of Playload url give the “Jenkins url/github-webhook/”, select the application/json as content type and create webhook.



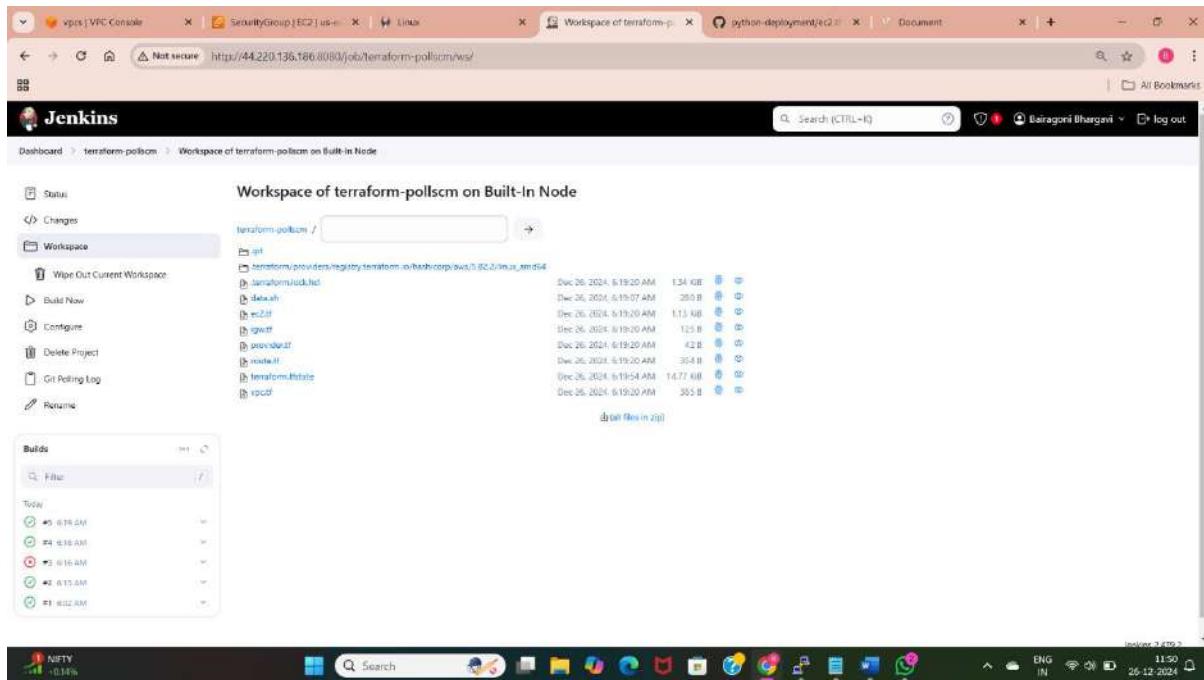
- The webhook is created successfully.



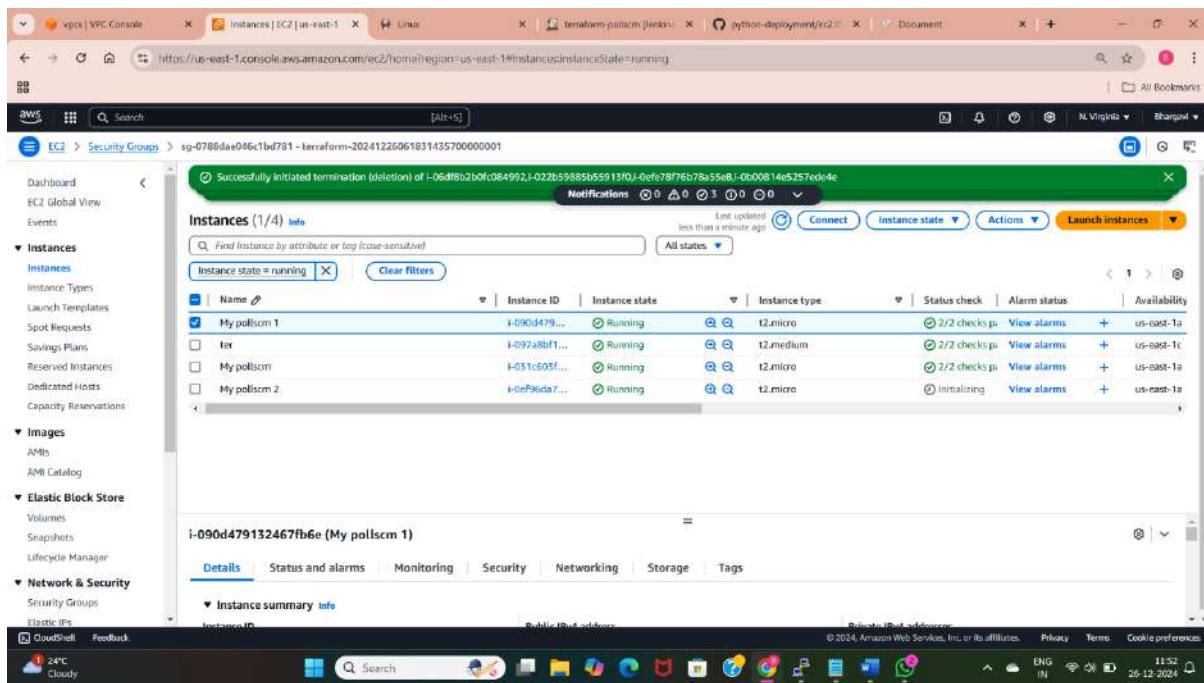
- ❖ Now go to build steps in the Jenkins job and select execute shell and give the commands.



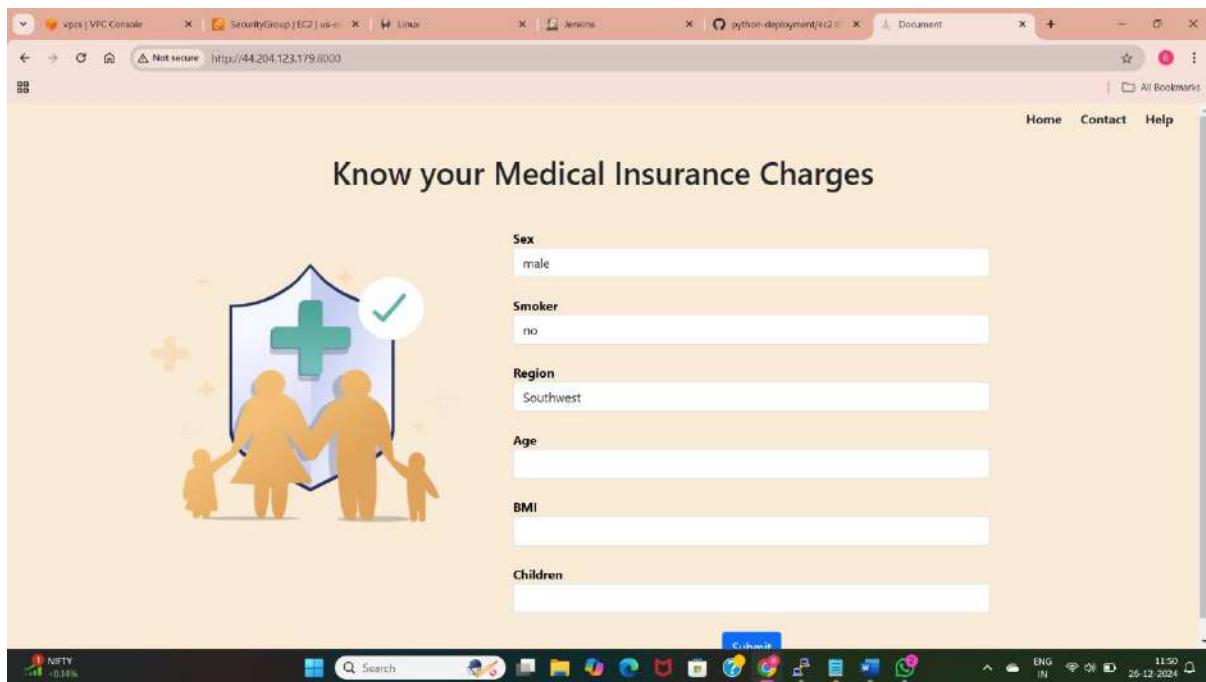
- ❖ Now we can see the builds when we change the code in github. I have made some changes in the github code so the build are done automatically by triggering github changes by following the poll scm syntax.



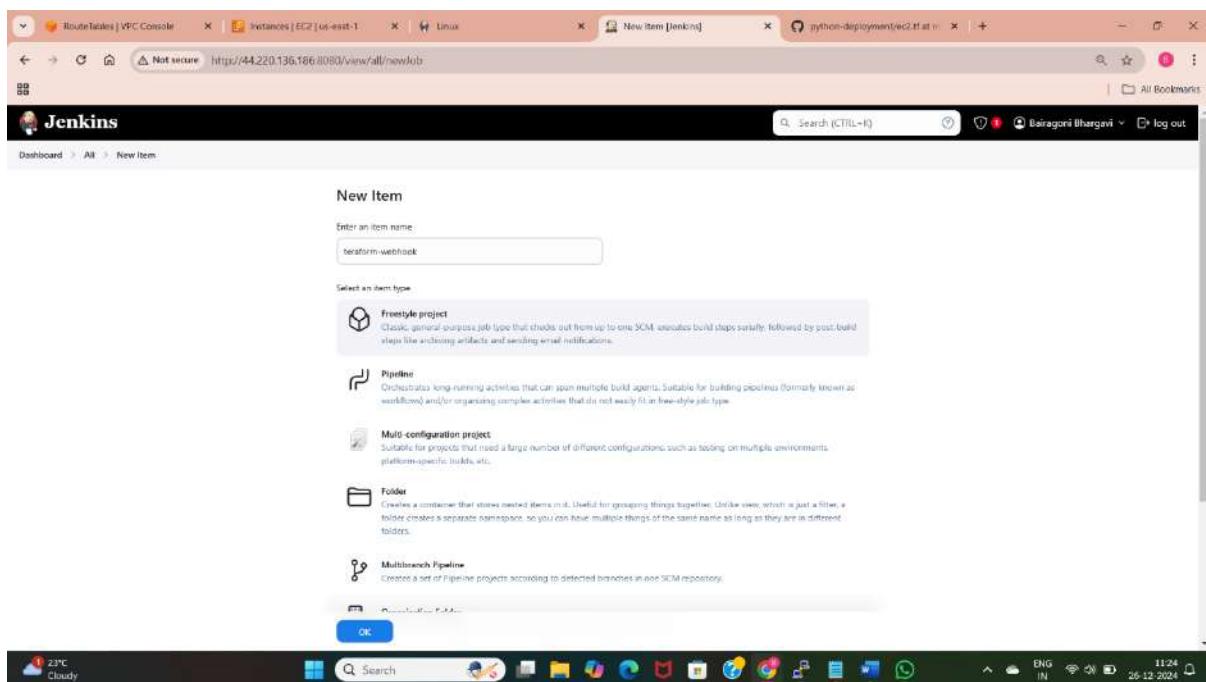
- ❖ After build is successful we can able to see the terraform files in the workspace of Jenkins.



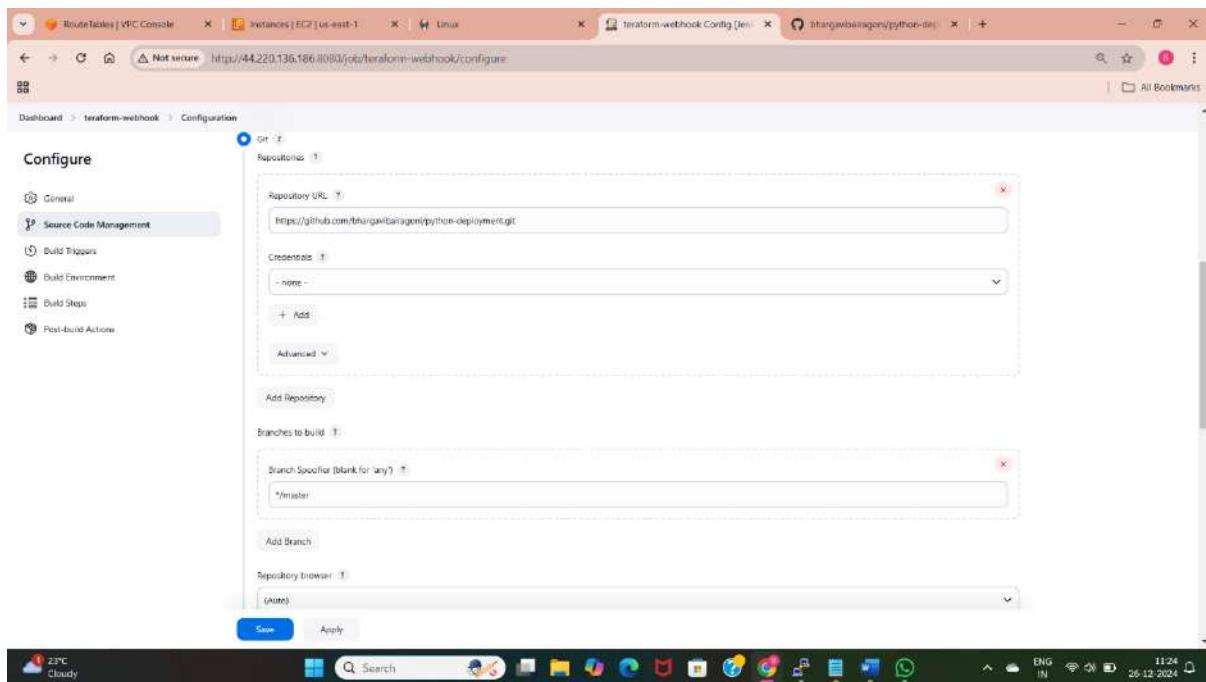
- ❖ New instances are created when the build is successful. Here the build is triggered three times using poll scm.



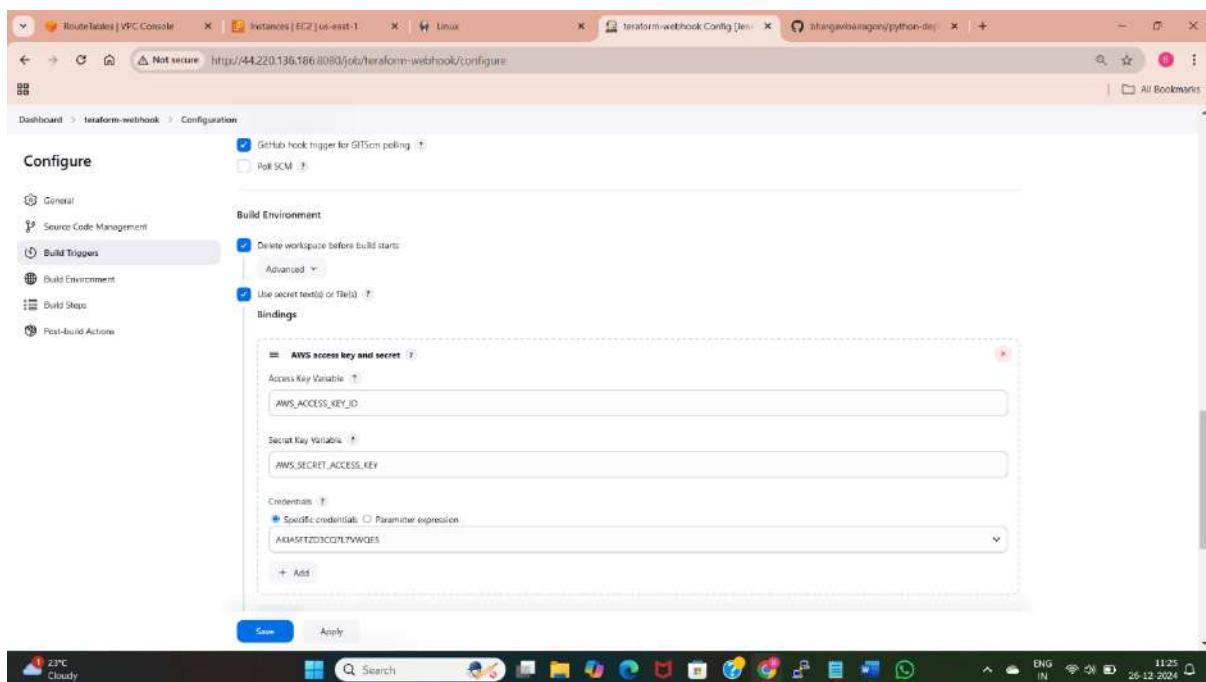
- ❖ After creating new instance using terraform and Jenkins pollscm copy the public IP address of new instance and paste in browser with port number of application.



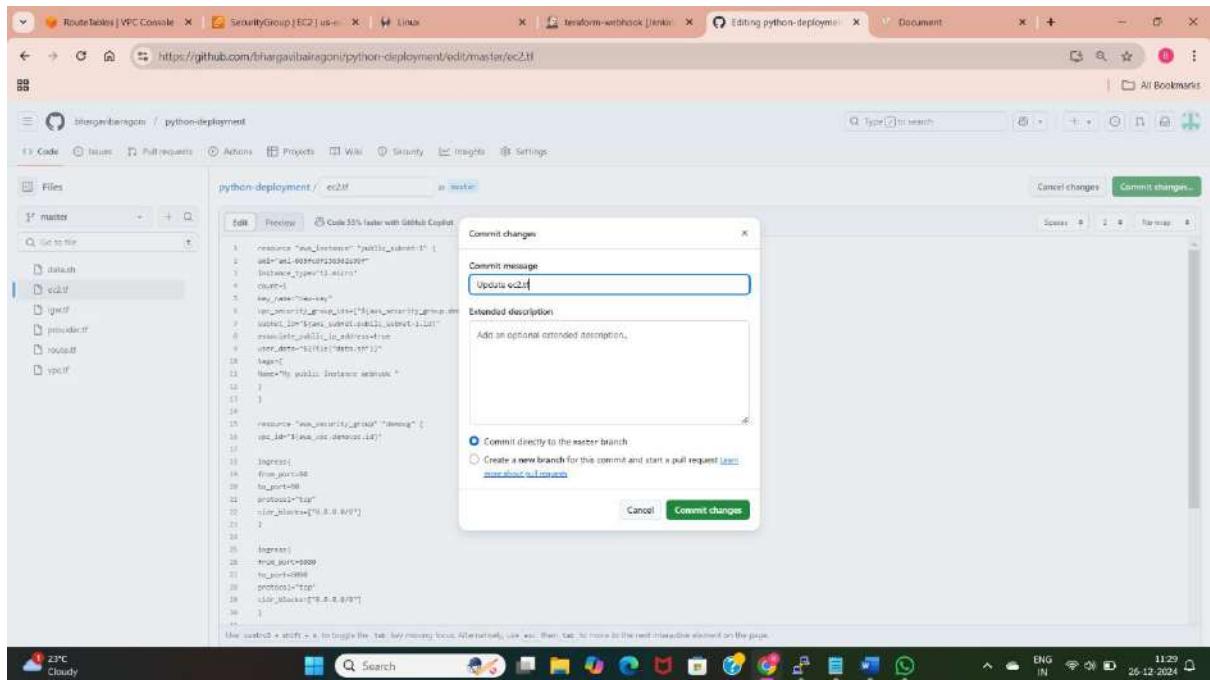
- ❖ Now create a another job with the name “terraform webhook” with freestyle job and click on “ok”.



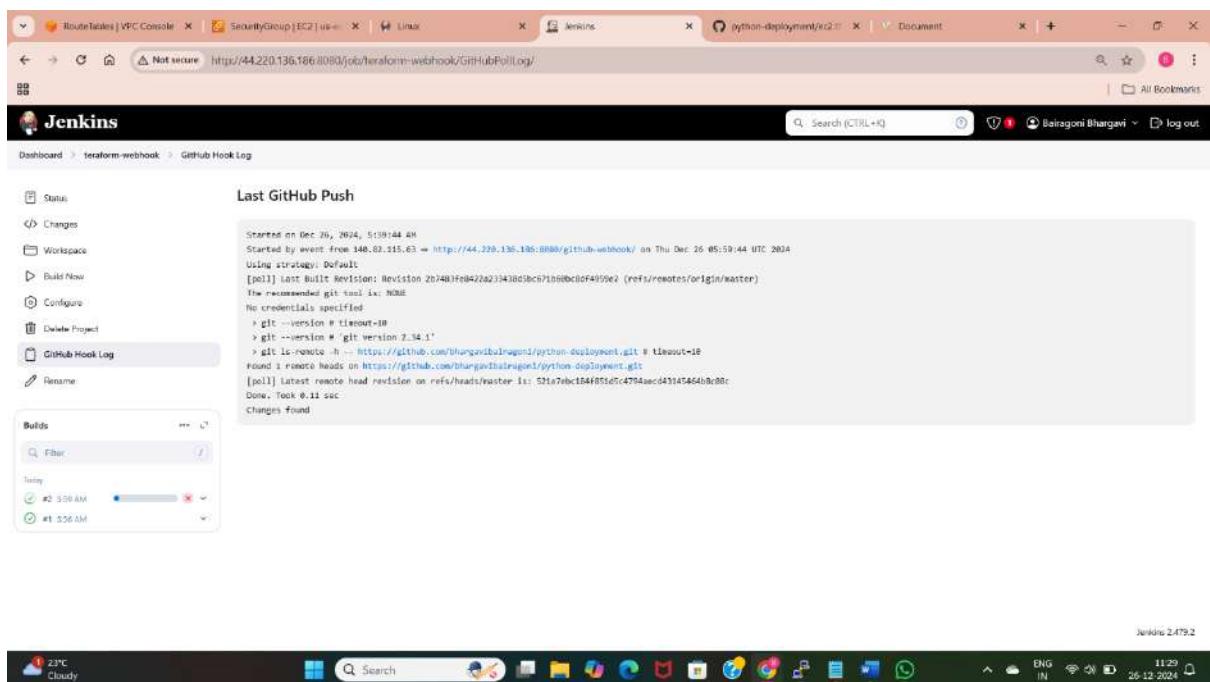
- ❖ Select the “git” option and provide the github code url.



- ❖ Now select the “Build Environment” in the Jenkins job and select <secret text or file” which takes the aws credentials.
- ❖ Here we need to install the aws credential plugin by moving into manage Jenkins>plugin.
- ❖ After navigate to “credentials” and give the credentials by selecting “global credentials”.



- ❖ Here am changing the code in the github.



- ❖ After finding the changes in the github, webhook is triggered changes to the Jenkins and build is started in Jenkins.

The screenshot shows the Jenkins interface for a workspace named 'terraform-webhook'. The left sidebar has options like Status, Changes, Workspace (which is selected), Wipe Out Current Workspace, Build Now, Configure, Delete Project, GitHub Hook Log, and Rename. The main area displays a list of files and their details:

	Date	Size	Actions
terraform.providers.registry.terraform.us-east-1.v2.0.0-rc.0.tar.gz	Dec 26, 2024, 6:01:09 AM	1.34 KB	...
.jenkinsci-builds	Dec 26, 2024, 6:01:00 AM	305 B	...
.terraformstate.lockfile	Dec 26, 2024, 6:01:02 AM	200 B	...
.datafile	Dec 26, 2024, 6:01:05 AM	1.55 KB	...
.env	Dec 26, 2024, 6:01:03 AM	125 B	...
.providerfile	Dec 26, 2024, 6:01:05 AM	42 B	...
.rootfile	Dec 26, 2024, 6:01:05 AM	354 B	...
terraform.state	Dec 26, 2024, 6:04:02 AM	0 B	...
vocat	Dec 26, 2024, 6:03:59 AM	305 B	...

Below this is a 'Builds' section with a timeline from 6:01 AM to 6:03:59 AM. At the bottom right, it says 'Jenkins 2.479.2'.

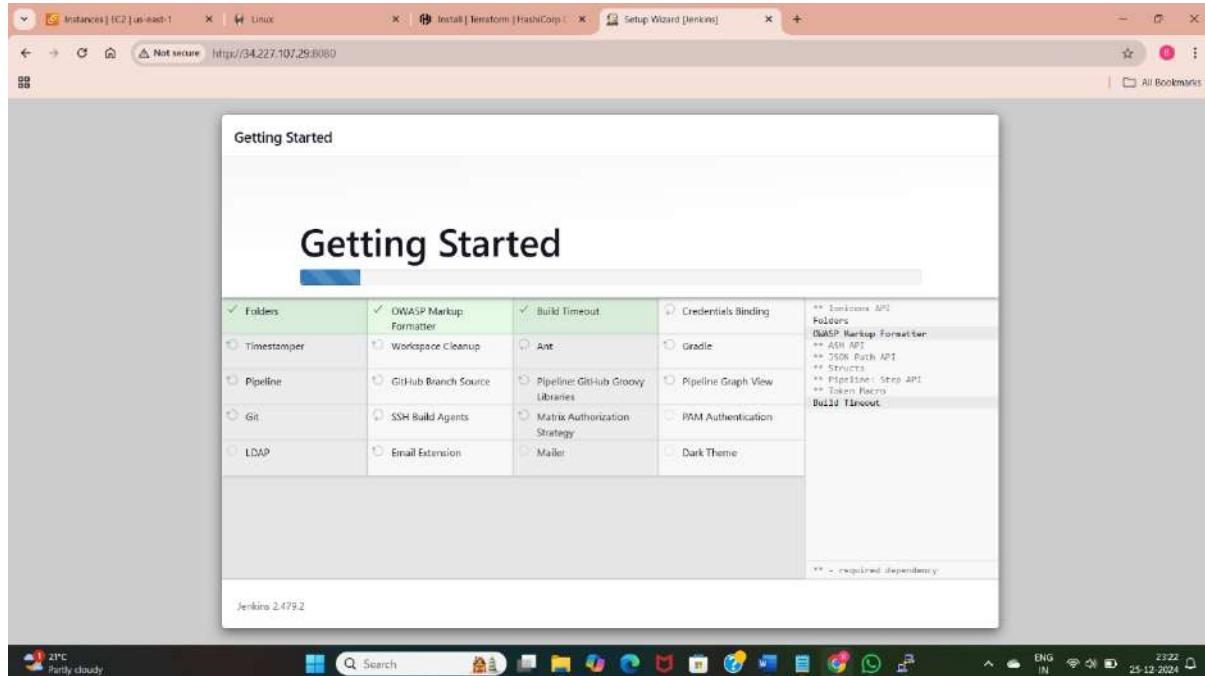
- ❖ We can see the files in the workspace of Jenkins.

The screenshot shows a web application titled 'Know your Medical Insurance Charges'. It features a background illustration of a family holding hands in front of a shield with a green cross. On the right, there are input fields for 'Sex' (male), 'Smoker' (no), 'Region' (Southwest), 'Age', 'BMI', and 'Children'. Below these fields is a blue 'Calculate' button. The top navigation bar includes links for Home, Contact, and Help. The bottom status bar shows the date and time as 26-12-2024, 11:29.

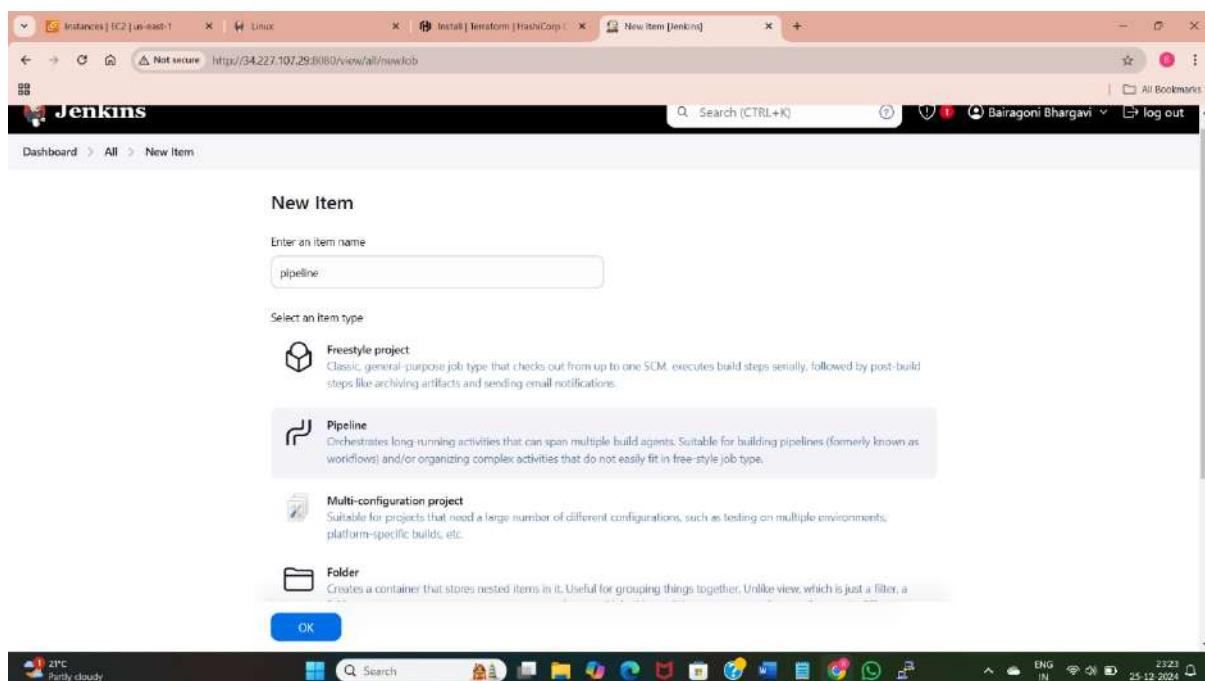
- ❖ After build is successful, we can access the application using the new instance IP address.

## METHOD -10: BUILD AND DEPLOY PYTHON APPLICATIONS WITH PIPELINE METHOD (CREATE CLONE JOB AND BUILD JOB) WITH BUILD PERIODICALLY, POLLSCM AND WEBHOOKS

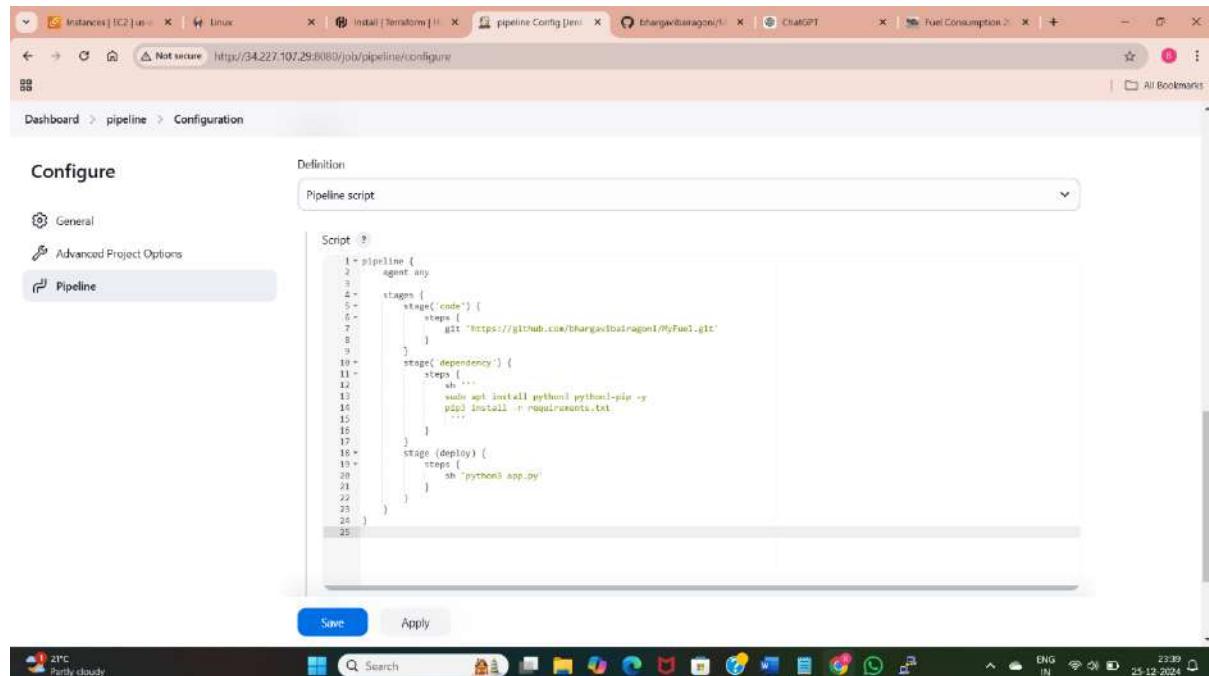
To deploy python application with pipeline method we need to install the EC2 instance and setup Jenkins.



- Here am installing the plugins in the Jenkins.



- Now create a new item with the name “pipeline” as pipeline job.



- Now create a pipeline to deploy the python application. Here am using three stages code, dependency, deploy.
  - In code stage taking the code from github using url.
  - In dependency stage install the requirements of myfuel application requirements.
  - In deploy stage deploy the python application using “screen -m -d python3 ./app.py && or python3 app.py”.

```
ubuntu@ip-172-31-21-4: ~
$ nano 5.0                                     /etc/sudoers_top
Defaults    assume_path="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"
Defaults    use_pty

# This preserves proxy settings from user environments of root
#(equivalent users (group sudo))
Defaults:sudo env_keep += "http_proxy https_proxy ftp_proxy all_proxy no_proxy"

# This allows running arbitrary commands, but so does ALL, and it means
# different subjects have their choice of editor respected.
#Defaults:sudo env_keep += "EDITOR"

# Completely harmless preservation of a user preference.
#Defaults:sudo env_keep += "TERM colors"

# While you shouldn't normally run GIT as root, you need to with gitkeeper
#Defaults:sudo env_keep += "GIT_AUTHOR GIT_COMMITTER"

# Per-user preferences: root won't have sensible values for them.
#Defaults:sudo env_keep += "EMAIL BERTHOMAIN BERTHOMAIN"

# "sudo scp" or "sudo rsync" should be able to use your SSH agent.
#Defaults:sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Print for GPG agent
#Defaults:sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the adm group may gain root privileges
%admin  ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "%include" directives:
%include /etc/sudoers.d

jenkins ALL=(ALL) NOPASSWD: ALL
Have modified buffer?
  [ ] Yes
  [ ] No
  [ ] Cancel
```

- In visudo file give the sudo permissions in Jenkins.

**Stage View**

	code	dependency	deploy
Average stage times:	5s	15s	1min 37s
#3	935ms	3s	4min 52s
#2	865ms	42s	483ms failed
#1	13s	680ms failed	212ms failed

**Builds**

- #3 - 6:03 PM
- #2 - 6:01 PM
- #1 - 5:58 PM

**Permalinks**

- Last build (#3), 2 min 32 sec ago
- Last failed build (#2), 2 min 32 sec ago
- Last unsuccessful build (#2), 2 min 32 sec ago
- Last completed build (#2), 2 min 32 sec ago

- Now build the pipeline . We can see that build is successful after two builds.

**Fuel Consumption**

Make: Acura

Model: [dropdown]

Vechile Class: Compact

Transmission: AM8

Fuel Type: Z

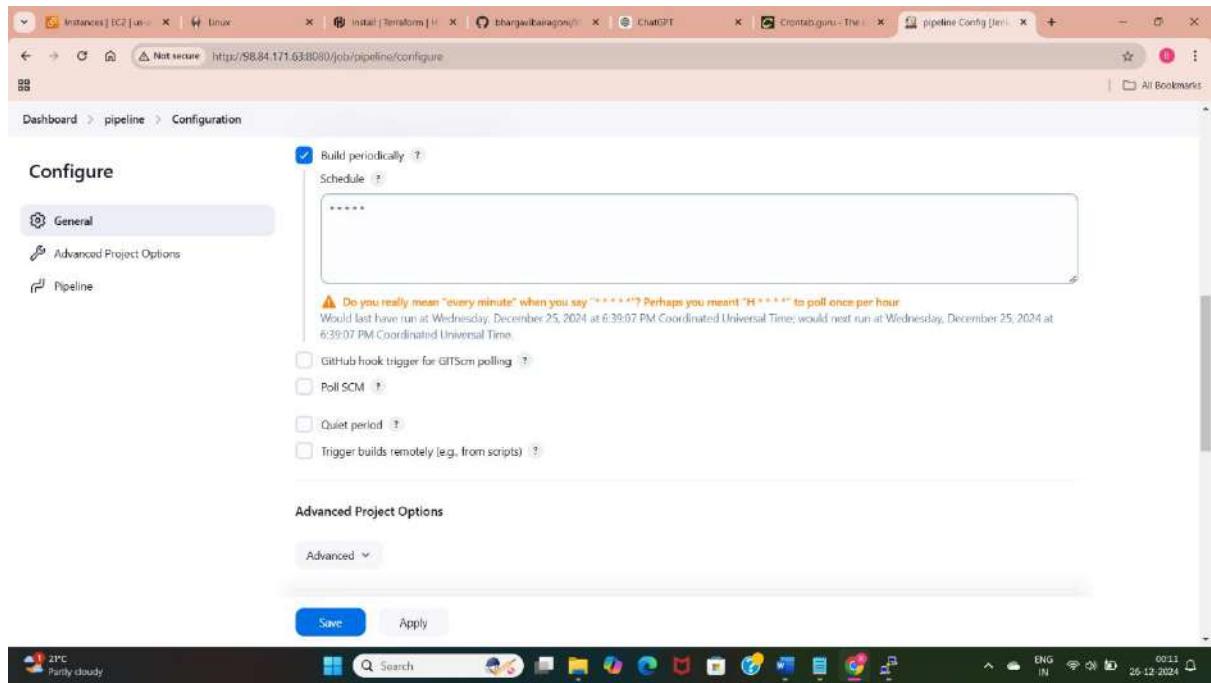
Engine Size: [input field]

Cylinders: [input field]

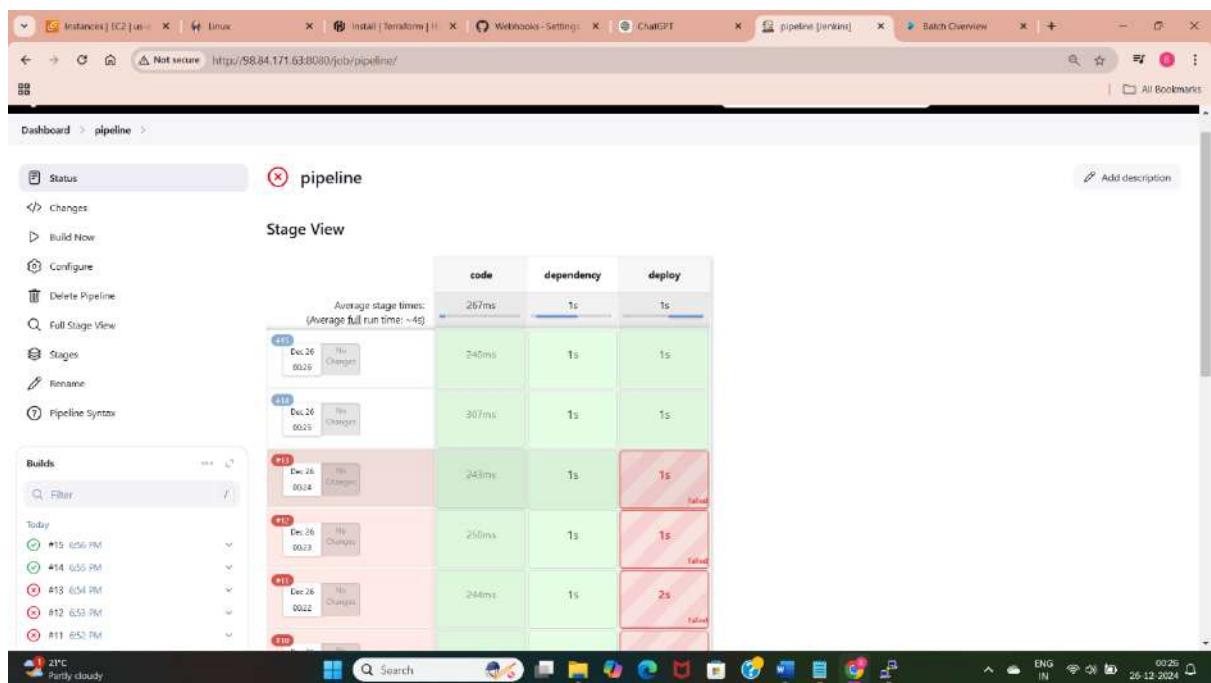
CO2: [input field]

SmokeRate: [input field]

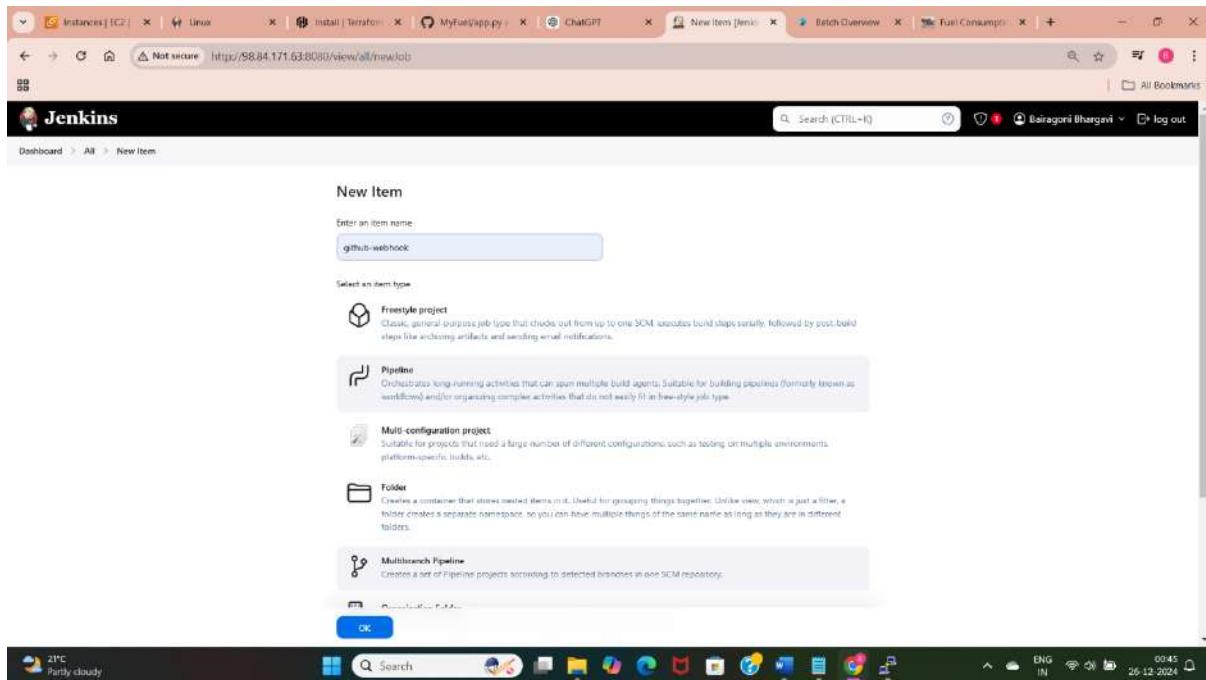
- Now we can able to access the application using public Ip address of instance and port number.



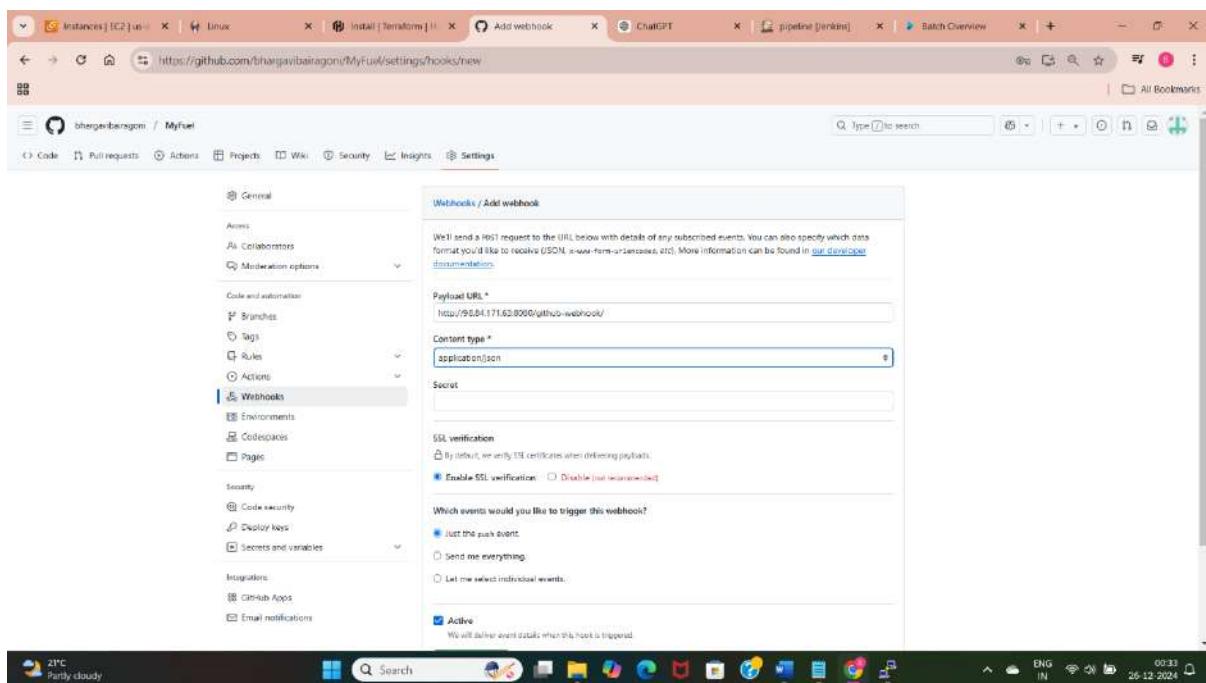
- Now use the build periodically option to build the pipeline. Here am setting the build periodically for every minute using `*****`.



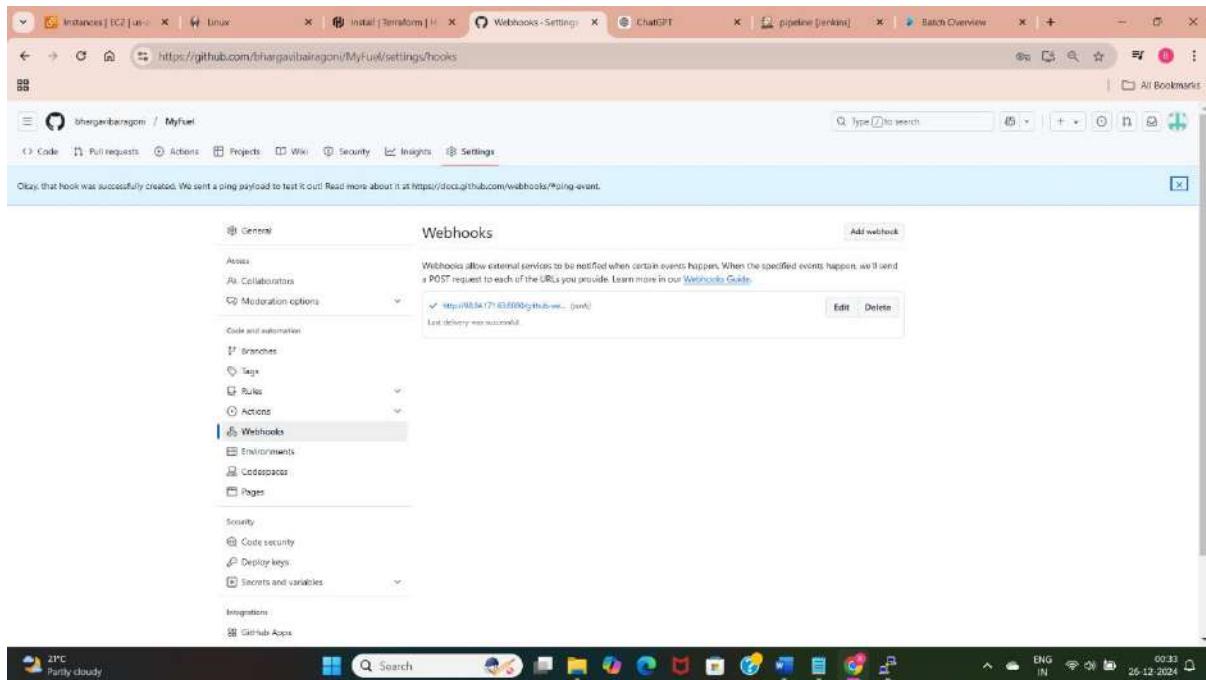
- Now we can see the builds. These are running for every minute.



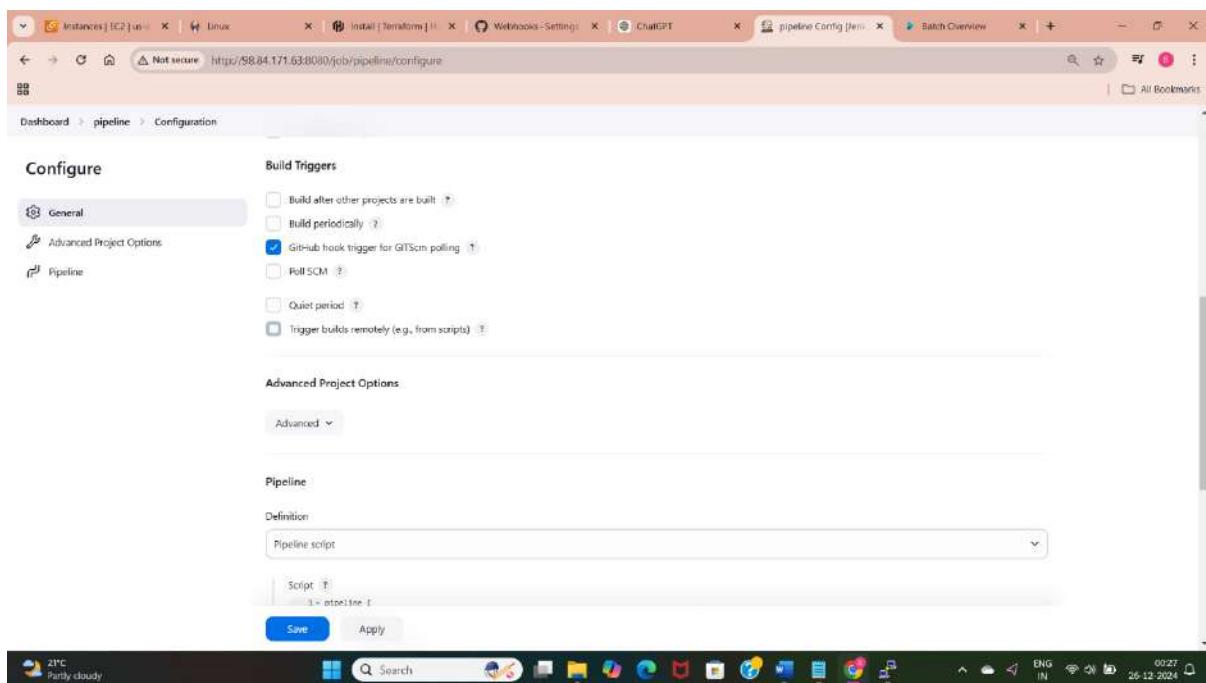
- Creating a new pipeline job with the name “github-webhook” to deploy python application.



- Now create a webhook in github application which you want to deploy using Jenkins.
- Give the Jenkins url in the payload url.
- Select the application/json as content type and save the webhook.



- Webhook is successfully created.



- Now am deploying the application using the “Github hook trigger for GITScm polling”.
- Webhook is used to trigger the changes from github to Jenkins.
- When I change the code in github , webhook is triggered and a new build will start in jenkins.

The screenshot shows the Jenkins interface with the URL <http://98.84.171.63:8080/job/pipeline/GitHubPollLog/>. The main content area displays the "Last GitHub Push" log, which includes the following details:

```
Started on Dec 25, 2024, 6:59:28 PM
Started by event from 10.82.115.89 -> http://98.84.171.63:8080/github-webhook/
Using strategy: Default
[poll] last Built Revision: Revision e8ad8ac10258b37c98ff3863a4911a62ea24ad1 (refs/remotes/origin/master)
The recommended git tool is: NONE
No credentials specified
> git --version # timeout=10
> git --version # 'git version 2.34.1'
> git ls-remote -h -- https://github.com/BhargavBairagi/InNews.git # timeout=10
Found 1 remote heads on https://github.com/BhargavBairagi/InNews.git
[poll] Latest remote head revision on refs/heads/master is: 8e4defed4f012a5fb0d36e1f33347a0999a0624
Done. Took 0.1 sec.
Changes Found
```

The sidebar on the left lists various Jenkins management options: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and GitHub Hook Log (which is currently selected).

- Now we can see in the pipeline job that webhook is triggered and builds are scheduled.
- Here we can check the “last github push”.

The screenshot shows the Jenkins Pipeline configuration page with the URL <http://98.84.171.63:8080/job/pipeline/configure>. The "Configure" section is expanded, specifically the "Poll SCM" tab. The configuration includes:

- Build periodically (unchecked)
- GitHub hook trigger for GITScm polling (unchecked)
- Poll SCM (checked):
  - Schedule: \* \* \* \* \*
  - No schedules so will only run due to SCM changes if triggered by a post-commit hook (unchecked)
  - Ignore post-commit hooks (unchecked)
  - Quiet period (unchecked)
  - Trigger builds remotely (e.g., from scripts) (unchecked)

Below the SCM configuration, there are sections for "Advanced Project Options" (with an "Advanced" dropdown menu) and "Pipeline". At the bottom of the configuration page are "Save" and "Apply" buttons.

- Now am deploying the python application using poll scm.
- Pollscm is used to trigger the changes from github to Jenkins using the crontab syntax.
- Here am using the every minute build crontab syntax using \* \* \* \* \* in poll scm.

The screenshot shows the Jenkins interface for a pipeline named 'pollcm'. On the left, a sidebar lists pipeline stages: Status, Changes, Build Now, Configure, Delete Pipeline, Full Stage View, Stages, Rename, Pipeline Syntax, and Git Polling Log. The 'Git Polling Log' tab is selected. The main content area displays the 'Git Polling Log' with the following text:

```
Started on Dec 25, 2024, 7:14:00 PM
Using strategy: Default
[Pipeline] Last Built Revision: Revision 13cf67d340b442b827b43fdd4186c810e5f7bdc (refs/remotes/origin/master)
The recommended git tool is: NONE
No credentials specified
> git --version & timeout=10
> git --version # git version 3.34.1'
> git ls-remote -h -- https://github.com/BaragoniBharagvi/MyFuel.git & timeout=10
Found 1 remote heads on https://github.com/BaragoniBharagvi/MyFuel.git
[origin] latest remote head revision on refs/heads/master is: 13cf67d340b442b827b43fdd4186c810e5f7bdc
Done. Took 0.1 sec
Changes found
```

Below this, a 'Builds' section shows the following build history:

Build	Time	Status
1	2024-12-25 18:14:00	Success
2	2024-12-25 18:13:00	Success
3	2024-12-25 18:12:00	Success
4	2024-12-25 18:11:00	Success
5	2024-12-25 18:10:00	Success

The system status bar at the bottom indicates it's 21°C, Party cloudy, and the date is 26-12-2024.

- We can check the builds are running for every minute when I have changed the github code.
- We can also check with the git polling logs.

The screenshot shows the Jenkins interface for the same pipeline 'pollcm'. The 'Stage View' section is highlighted. The main content area displays the 'Stage View' with the following text:

Average stage times:  
Average full run time - 27s

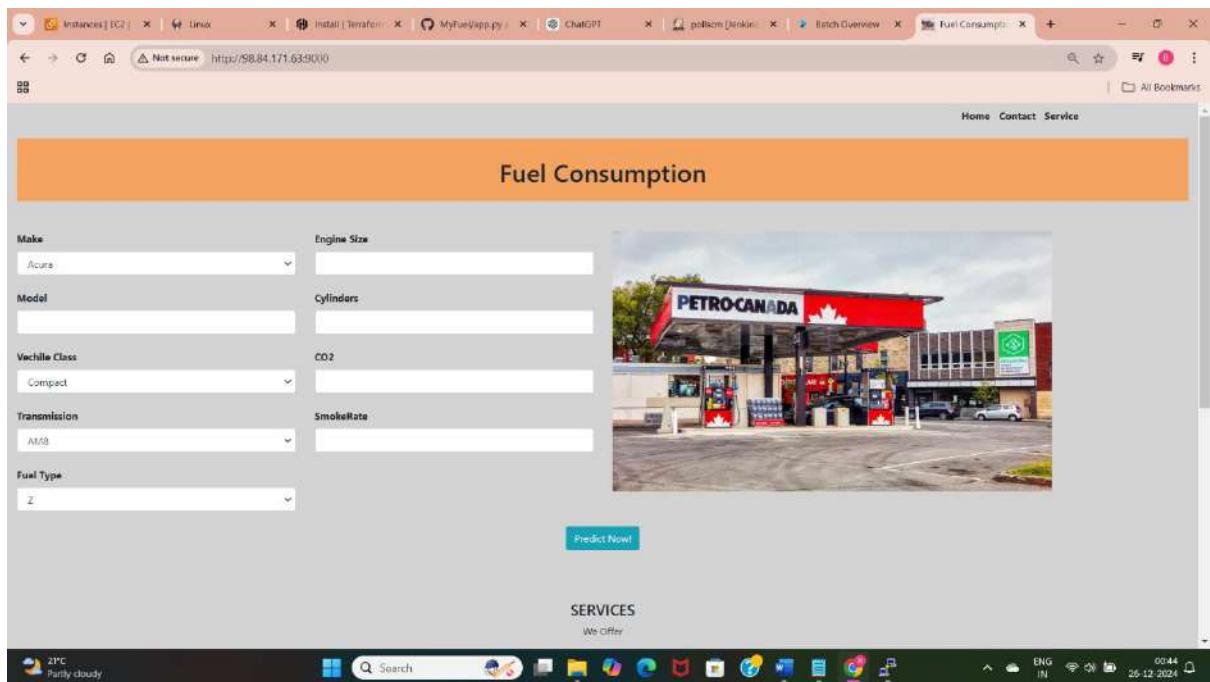
stage	code	dependency	deploy
Dec 25 2024	5s	1s	52s
Dec 26 2024	4s	1s	1s
Dec 26 2024	4s	1s	1s
Dec 26 2024	4s	1s	1s
Dec 26 2024	4s	1s	1s

Below this, a 'Builds' section shows the following build history:

Build	Time	Status
1	2024-12-25 18:14:00	Success
2	2024-12-25 18:13:00	Success
3	2024-12-25 18:12:00	Success
4	2024-12-25 18:11:00	Success
5	2024-12-25 18:10:00	Success

The system status bar at the bottom indicates it's 21°C, Party cloudy, and the date is 26-12-2024.

- This is the build stage view we can see the builds that every stage is successfully.



- After build is success am able to access the application using ip address of the instance and port of application.