

# PYTHON APPLICATION DEPLOYMENT METHODS

-- G.Vinay

## Method-1: Deploy python web applications using AWS resources.

- ❖ Now launch the instance using the amazon linux or Ubuntu but here I took amazon linux.

The screenshot shows the AWS EC2 Instances page. There are three instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPV4
flask-1	i-0ffdf4f7a3f935a60	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-34-20...
flask-1	i-019c5c8516a8a3b42	Terminated	t2.micro	-	View alarms +	us-east-1b	-
python dep-1	i-0f8382c3eb740902b	Terminated	t2.micro	-	View alarms +	us-east-1b	-

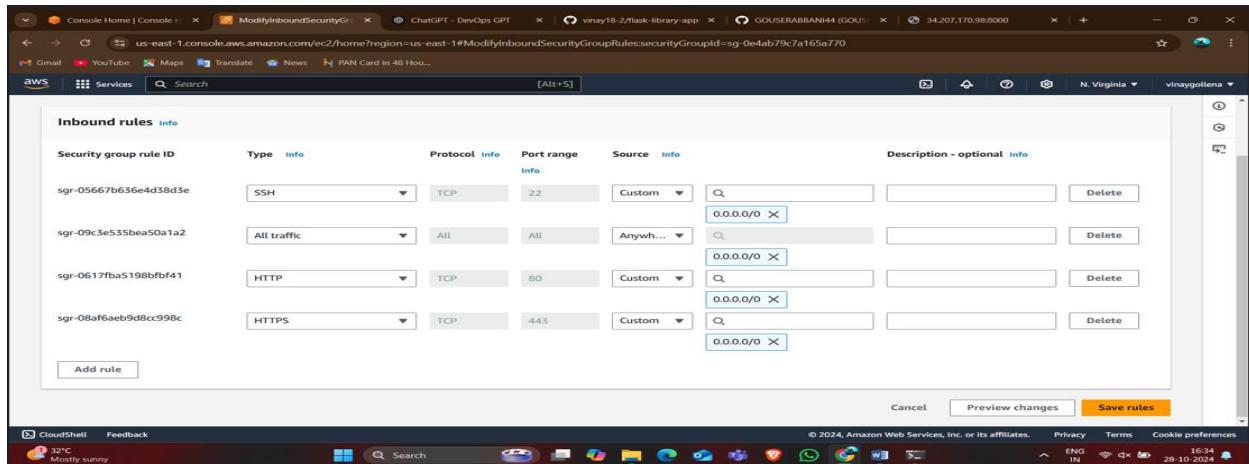
A tooltip "Public IPv4 address copied" is displayed over the Public IPv4 field for the first instance (i-0ffdf4f7a3f935a60). The tooltip text is "34.207.170.98 | open address".

The screenshot shows the AWS EC2 Instances page. There are two instances listed:

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPV4
python-1	i-0b545d059df654c50	Running	t2.micro	2/2 checks passed	View alarms +	us-east-1b	ec2-34-20...
flask-1	i-0ffdf4f7a3f935a60	Stopped	t2.micro	-	View alarms +	us-east-1b	-

A tooltip "Public IPv4 address copied" is displayed over the Public IPv4 field for the first instance (i-0b545d059df654c50). The tooltip text is "34.207.223.137 | open address".

- ❖ In the security group inbound rules we have to give the all traffic along with ssh,http,https.



- ❖ Now connect to the terminal by using git bash or command prompt.

```
ec2-user@ip-172-31-40-235:~ + 
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd downloads

C:\Users\user>ssh -i "vinay.pem" ec2-user@ec2-34-207-170-98.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-170-98.compute-1.amazonaws.com (34.207.170.98)' can't be established.
ED25519 key fingerprint is SHA256:1ElecSELcsQQLZ2Exq0Nbmp1gK+7afxt+3Si996Nl+k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-170-98.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

      _\##_
     /_##_#_
    /##_##_
   /#_--_
  V`---->
   /_/_/ A newer version of Amazon Linux is available!
  /_/_/_/ Amazon Linux 2023, GA and supported until 2028-03-15.
  /_/_/_/_ https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-40-235 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-40-235 ~]$ sudo yum full-upgrade -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No such command 'full-upgrade'. Please use /bin/yum --help
[ec2-user@ip-172-31-40-235 ~]$ sudo yum upgrade -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-40-235 ~]$ sudo yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64

32°C Mostly sunny  Search  CloudShell  Feedback  ENG IN  16:29  28-10-2024
```

- ❖ Update the instance by using the command “sudo yum update -y”
- ❖ Install the git by using “sudo yum install git -y”.

```

ec2-user@ip-172-31-40-235:~ + 
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd downloads

C:\Users\user\Downloads>ssh -i "vinay.pem" ec2-user@ec2-34-207-170-98.compute-1.amazonaws.com
The authenticity of host 'ec2-34-207-170-98.compute-1.amazonaws.com (34.207.170.98)' can't be established.
ED25519 key fingerprint is SHA256:1E1ecselCsQ0L2Exq0Nbmp1gK+7afXT+3Si996Nl+k.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-207-170-98.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

#_
  _###_      Amazon Linux 2
~~ \####\    AL2 End of Life is 2025-06-30.
~~  \##\    A newer version of Amazon Linux is available!
~~   \'#-->
~~   V`-->
~~   /     Amazon Linux 2023, GA and supported until 2028-03-15.
~~   /     https://aws.amazon.com/linux/amazon-linux-2023/
~/m/`-->

[ec2-user@ip-172-31-40-235 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-40-235 ~]$ sudo yum full-upgrade -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No such command: full-upgrade. Please use /bin/yum --help
[ec2-user@ip-172-31-40-235 ~]$ sudo yum upgrade -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-40-235 ~]$ sudo yum install git -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:2.40.1-1.amzn2.0.3 will be installed
--> Processing Dependency: git-core = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: git-core-doc = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl-Git = 2.40.1-1.amzn2.0.3 for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Git) for package: git-2.40.1-1.amzn2.0.3.x86_64
--> Processing Dependency: perl(Term::ReadKey) for package: git-2.40.1-1.amzn2.0.3.x86_64

32°C Mostly sunny  Search  File  Home  Downloads  Applications  Network  Mail  Google  Web  Help  ENG IN  16:29  28-10-2024

```

❖ Clone the repository from github by using git command “git clone repo link”

```

ec2-user@ip-172-31-40-235:~ + 
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 perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-40-235 ~]$ git clone https://github.com/vinay18-2/flask-library-app.git
Cloning into 'flask-library-app'...
remote: Enumerating objects: 194, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 194 (delta 9), reused 2 (delta 2), pack-reused 178 (from 1)
Receiving objects: 100% (194/194), 858.96 KiB | 38.68 MiB/s, done.
Resolving deltas: 100% (112/112), done.
[ec2-user@ip-172-31-40-235 ~]$ cd flask-library-app/
[ec2-user@ip-172-31-40-235 flask-library-app]$ ll
total 52
-rw-rw-r-- 1 ec2-user ec2-user 9928 Oct 28 10:47 app.py
-rw-rw-r-- 1 ec2-user ec2-user 176 Oct 28 10:47 data.sh
-rw-rw-r-- 1 ec2-user ec2-user 16384 Oct 28 10:47 library.db
-rw-rw-r-- 1 ec2-user ec2-user 1581 Oct 28 10:47 old.py
-rw-rw-r-- 1 ec2-user ec2-user 795 Oct 28 10:47 pipeline
-rw-rw-r-- 1 ec2-user ec2-user 23 Oct 28 10:47 Procfile
drwxrwxr-x 2 ec2-user ec2-user 32 Oct 28 10:47 __pycache__
-rw-rw-r-- 1 ec2-user ec2-user 981 Oct 28 10:47 README.md
-rw-rw-r-- 1 ec2-user ec2-user 270 Oct 28 10:47 requirements.txt
drwxrwxr-x 2 ec2-user ec2-user 123 Oct 28 10:47 screenshots
drwxrwxr-x 5 ec2-user ec2-user 41 Oct 28 10:47 static
drwxrwxr-x 2 ec2-user ec2-user 308 Oct 28 10:47 templates
[ec2-user@ip-172-31-40-235 flask-library-app]$ pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting certifi==2022.12.7
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
    |████████| 155 kB 19.3 MB/s
Collecting charset-normalizer==2.0.7
  Downloading charset_normalizer-2.0.7-py3-none-any.whl (38 kB)
Collecting click==8.0.3
  Downloading click-8.0.3-py3-none-any.whl (97 kB)

32°C Mostly sunny  Search  File  Home  Downloads  Applications  Network  Mail  Google  Web  Help  ENG IN  16:29  28-10-2024

```

- ❖ Got ot the cloned file bys uding the command “cd /cloned files/”

```
ec2-user@ip-172-31-40-235:~ + - x
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 perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-40-235 ~]$ git clone https://github.com/vinay18-2/flask-library-app.git
Cloning into 'flask-library-app'...
remote: Enumerating objects: 194, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 194 (delta 9), reused 2 (delta 2), pack-reused 178 (from 1)
Receiving objects: 100% (194/194), 858.96 KiB | 30.68 MiB/s, done.
Resolving deltas: 100% (112/112), done.
[ec2-user@ip-172-31-40-235 ~]$ cd flask-library-app/
[ec2-user@ip-172-31-40-235 flask-library-app]$ ll
total 52
-rw-rw-r-- 1 ec2-user ec2-user 9928 Oct 28 10:47 app.py
-rw-rw-r-- 1 ec2-user ec2-user 176 Oct 28 10:47 data.sh
-rw-rw-r-- 1 ec2-user ec2-user 16384 Oct 28 10:47 library.db
-rw-rw-r-- 1 ec2-user ec2-user 1581 Oct 28 10:47 old.py
-rw-rw-r-- 1 ec2-user ec2-user 795 Oct 28 10:47 pipeline
-rw-rw-r-- 1 ec2-user ec2-user 23 Oct 28 10:47 Procfile
drwxrwxr-x 2 ec2-user ec2-user 32 Oct 28 10:47 __pycache__
-rw-rw-r-- 1 ec2-user ec2-user 981 Oct 28 10:47 README.md
-rw-rw-r-- 1 ec2-user ec2-user 274 Oct 28 10:47 requirements.txt
drwxrwxr-x 2 ec2-user ec2-user 123 Oct 28 10:47 screenshots
drwxrwxr-x 5 ec2-user ec2-user 41 Oct 28 10:47 static
drwxrwxr-x 2 ec2-user ec2-user 308 Oct 28 10:47 templates
[ec2-user@ip-172-31-40-235 flask-library-app]$ pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writable
Collecting certifi==2022.12.7
  Downloading certifi-2022.12.7-py3-none-any.whl (155 kB)
|██████████| 155 kB 19.3 MB/s
Collecting charset-normalizer==2.0.7
  Downloading charset_normalizer-2.0.7-py3-none-any.whl (38 kB)
Collecting click==8.0.3
  Downloading click-8.0.3-py3-none-any.whl (97 kB)

32°C
Mostly sunny
Windows Search Bar
ENG IN
16:29
10/20/2024
```

- ❖ Now run the command to install the pip3 to run requirement file.  
“pip3 install –r requirements.txt” and if any changes should do in the requirement.txt file than open the file by using “sudo vi requirement.txt” than change according to the required versions , than run the command “pip3 install –r requirement.txt”.

```
* Running on http://172.31.40.235:8000/ (Press CTRL+C to quit)
 * Restarting with stat
/home/ec2-user/.local/lib/python3.7/site-packages/flask_sqlalchemy/__init__.py:873: FSADeprecationWarning: SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and will be disabled by default in the future. Set it to True or False to suppress this warning.
'SQLALCHEMY_TRACK_MODIFICATIONS adds significant overhead and '
 * Debugger is active!
 * Debugger PIN: 739-527-533
103.213.202.11 - - [28/Oct/2024 10:55:56] "GET / HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:55:57] "GET /static/css/style.css HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:55:57] "GET /static/js/script.js HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:55:58] "GET /favicon.ico HTTP/1.1" 404 -
103.213.202.11 - - [28/Oct/2024 10:56:15] "GET /books HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:56:15] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:32] "GET /script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:33] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:34] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:46] "GET / HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:56:47] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:47] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:49] "GET /customers HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:56:50] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:50] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:56:58] "GET /customers/edit/1 HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:56:59] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:57:43] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:57:44] "GET /add_books HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:57:44] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:57:57] "POST /add_books HTTP/1.1" 302 -
103.213.202.11 - - [28/Oct/2024 10:57:58] "GET /books HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:57:58] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:58:10] "GET /customers HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:58:11] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:58:11] "GET /static/js/script.js HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:58:20] "GET / HTTP/1.1" 200 -
103.213.202.11 - - [28/Oct/2024 10:58:20] "GET /static/css/style.css HTTP/1.1" 304 -
103.213.202.11 - - [28/Oct/2024 10:58:20] "GET /static/js/script.js HTTP/1.1" 304 -
* [ec2-user@ip-172-31-40-235 Flask-library-app]$ sudo vi app.py
[ec2-user@ip-172-31-40-235 Flask-library-app]$
```

- ❖ After the requirement.txt file has installed than open the app.py file “sudo vi app.py”. Here we have to change the last line of the code inorder to host the python application.  
“app.run(host = “0.0.0.0/0”,port = 8000,debug=True).

```

ec2-user@ip-172-31-40-235: ~ + v
    search=request.form['search_box']
except:
    pass
if not search=="":
    r = requests.get('https://frappe.io/api/method/frappe-library?title='+search+'&page=' + str(id))
# result = r.json()
# print(r)
# return render_template('books_store.html',books=json.loads(r.text)[‘message’],search=search,page_no=id+1)
else:
    return render_template('books_store.html')

else:
    return render_template('books_store.html')

@app.route('/books/import<int:id>', methods=['GET', 'POST'])
def import_book(id):
    r = requests.get('https://frappe.io/api/method/frappe-library?isbn=' + str(id))
    book_details = json.loads(r.text)[‘message’][0]
    book_title=book_details[‘title’]
    book_author=book_details[‘authors’]
    book_stockQty=request.form[‘qty’]
    new_book = Books(title=book_title,authors=book_author,stockQty=book_stockQty,timesIssued=0)
    db.session.add(new_book)
    db.session.commit()
    return redirect('/books')
    #return render_template('books.html')

if __name__ == '__main__':
    try:
        new_book = Books(book_id=1,title="book_title",authors="book_author",stockQty=0,timesIssued=0)
        db.session.add(new_book)
        db.session.commit()
        new_customer = Customer(cust_id=1,name="name")
        db.session.add(new_customer)
        db.session.commit()
    except:
        pass
    app.run(host='0.0.0.0/0',port=8000,debug=True)
-- INSERT --

```

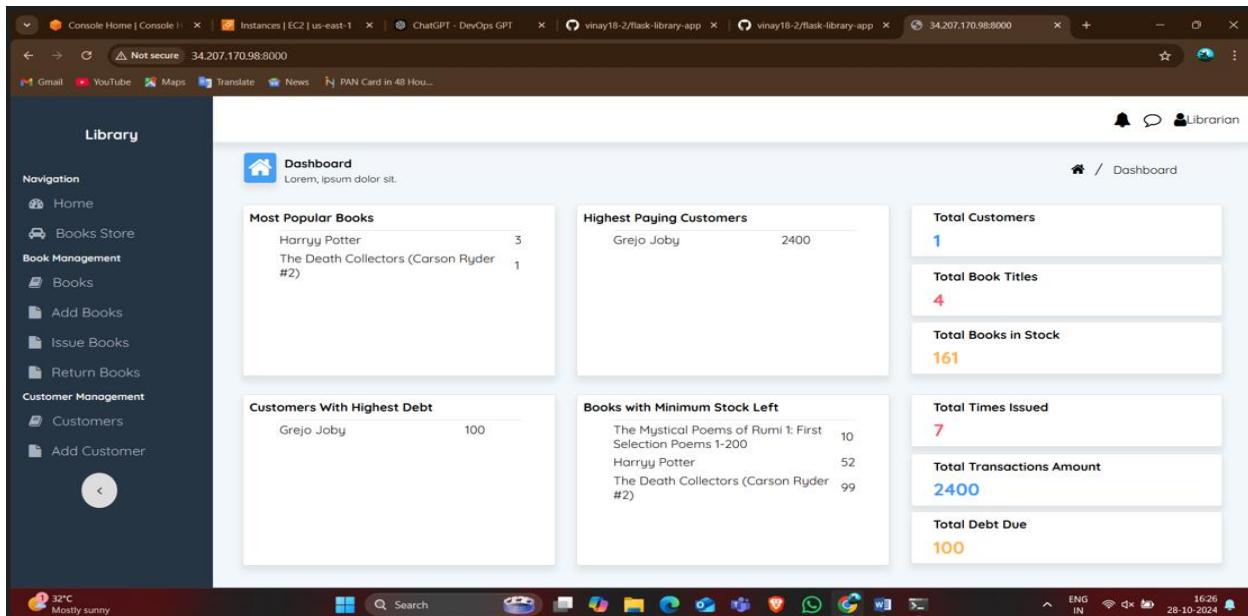
268, 38      Bot

32°C Mostly sunny

Search      Home      Task View      Start      File Explorer      File History      Taskbar

28-10-2024

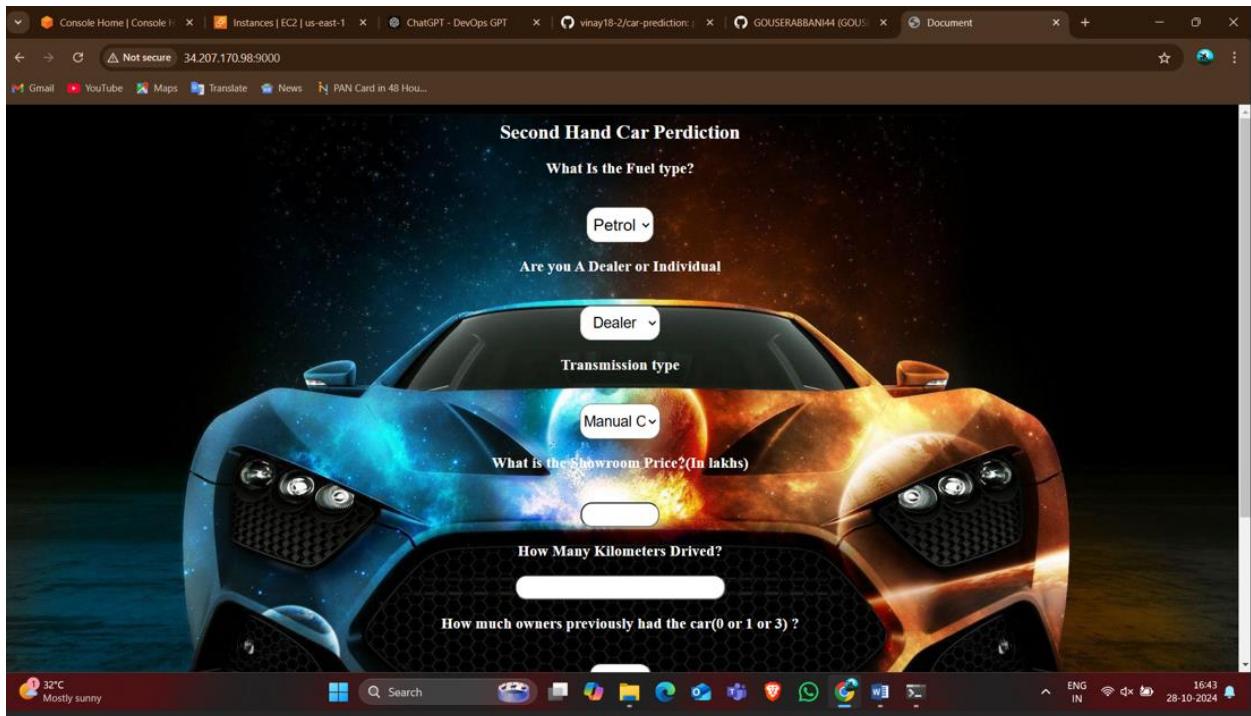
- ❖ Than run the command after done with that line in app.py.  
“python3 app.py” or screen -m -d python3 app.py”
- ❖ After than copy the public ip of the instance and paste it in the browser than browse that ip address with the given port number “34.207.170.98:8000”.
- ❖ And here the result of the python application i.e flask-library-app.



- ❖ Like the same process I have hosted the 13 applications.
- ❖ Everytime we have to check the requirement.txt file and give the required versions.
- ❖ Than open the app.py and go tot the line than edit the line with the ip address and port number than save it and run with the command “python3 app.py”
- ❖ After that copy the id address of the instance in the browser and search it.
- ❖ Here the applications I hosted.
  1. Car-prediction
  2. Fish
  3. Flight prediction
  4. Penguin
  5. inNews
  6. medical insurance
  7. portfolio
  8. agri
  9. indian-Liver-patients
  10. hearing
  11. fuel-consumption-rating
  12. USA-Houasing
  13. MyFuel

- ❖ Here the results of the applications I hosted.

1. Car-prediction.



## 2. Fish.



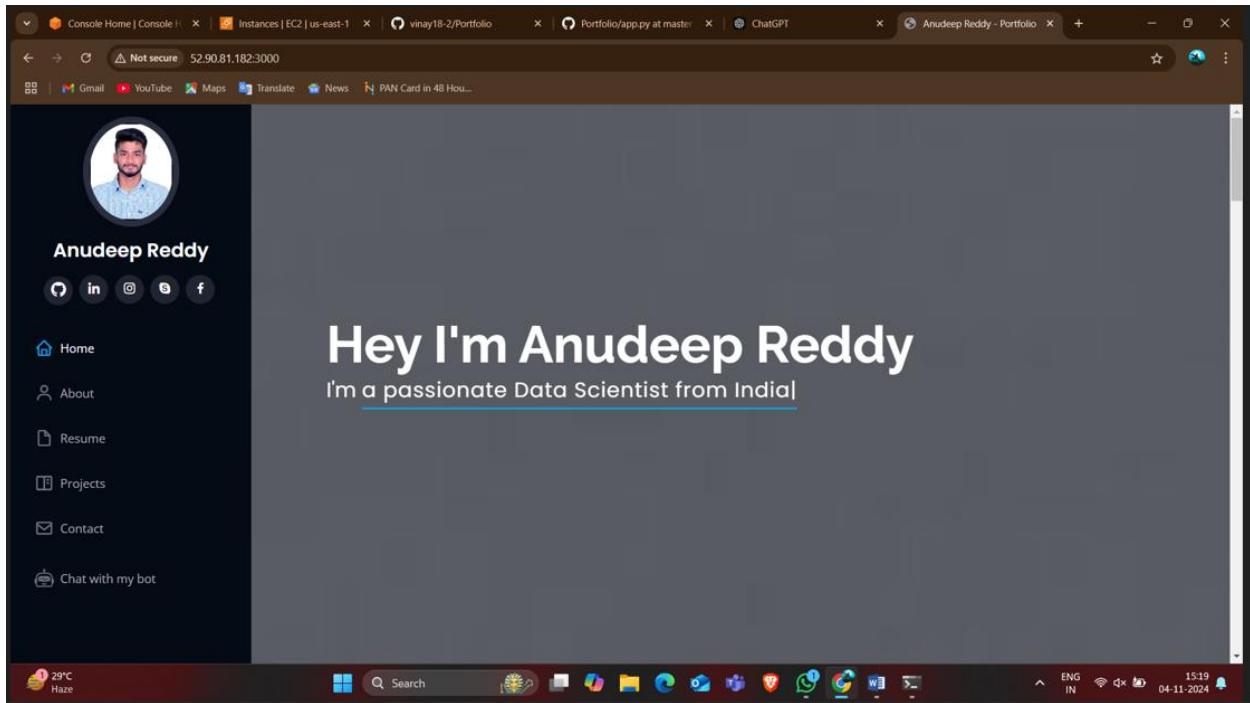
A screenshot of a web browser showing a form titled "FISH WEIGHT". The form includes fields for "Species" (Bream), "Vertical length in cm" (23.2), "Diagonal length in cm" (25.4), "Cross length in cm" (30.0 - highlighted in red with an error message "Please fill out this field."), "Height" (11.5200), and "Width" (4.0200). A "SUBMIT" button is at the bottom.

## 3. Mecial-insurance.

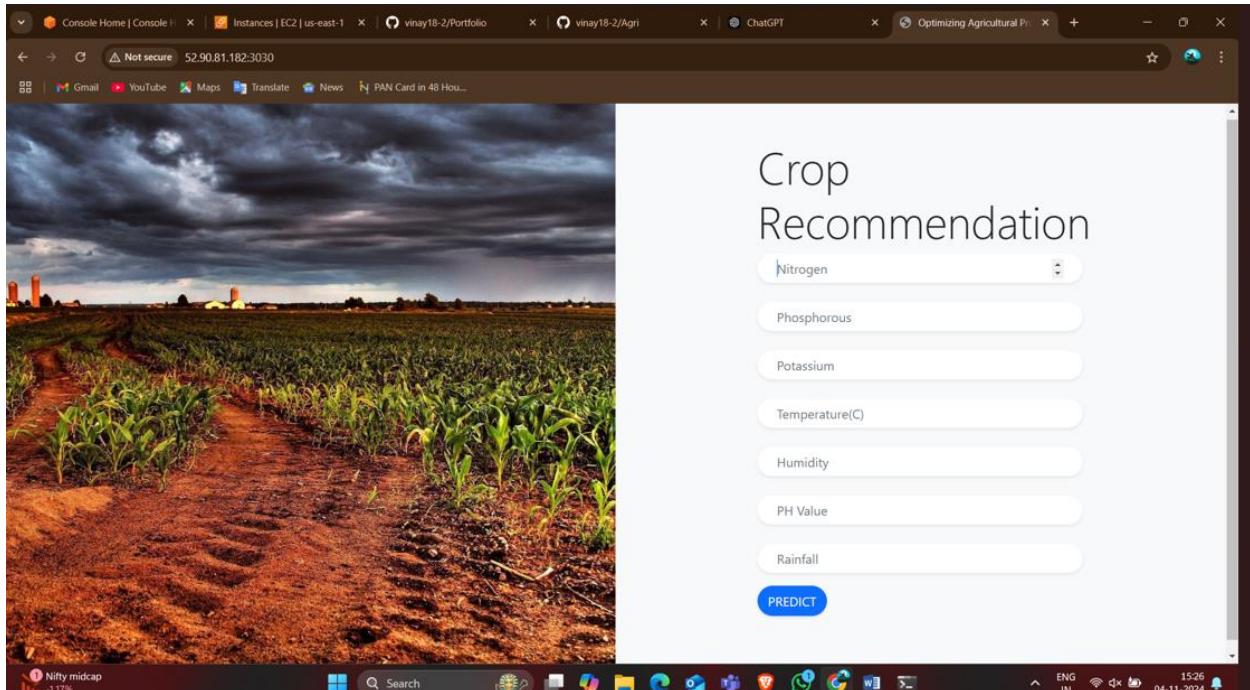


A screenshot of a web browser showing a form titled "Know your Medical Insurance Charges". The form includes fields for "Sex" (male), "Smoker" (no), "Region" (Southwest), "Age" (empty), "BMI" (empty), and "Children" (empty). A "Submit" button is at the bottom. The browser interface shows it's not secure and the URL is 52.90.81.182:7000.

#### 4. Portfolio.



#### 5. Agri.



## 6. Liver patient.

A screenshot of a web browser window titled "LIVER PATIENT". The page contains several input fields arranged in a grid. The first row has two fields: the first is labeled "Female" and the second is empty. The second row has two empty fields. The third row has two empty fields. The fourth row has two empty fields. The fifth row has two empty fields. A blue "Submit!" button is located at the bottom center of the form. The browser's address bar shows the URL "52.90.81.182:7070/predict". The status bar at the bottom indicates "29°C Haze" and the date "04-11-2024".

## 7. Hearing test.

A screenshot of a web browser window titled "Hearing Test". The page has a large orange header with the text "Hearing Test". Below the header, there are two input fields: "Age \* Acura" and "Physical Score \* ILX". A blue "Predict Now!" button is located below these fields. The browser's address bar shows the URL "52.90.81.182:6060". The status bar at the bottom indicates "Air: Moderate Now" and the date "04-11-2024".

## 8. Fuel consumption.

The screenshot shows a web browser window titled "Fuel Consumption". The page features a form with several input fields and dropdown menus. On the left, there are fields for "Make" (Acura), "Model" (empty), "Vehicle Class" (Compact), "Transmission" (AM8), and "Fuel Type" (Z). On the right, there are fields for "Engine Size" (empty), "Cylinders" (empty), "CO2 emission" (empty), "CO2 Rating" (empty), and "SmokeRate" (empty). Below the form is a large image of a fuel gauge with the needle pointing between E and F. At the bottom of the form is a blue "Predict Now!" button. The browser's address bar shows the URL "52.90.81.182:5050". The taskbar at the bottom of the screen includes icons for various applications like Gmail, YouTube, Maps, Translate, News, and ChatGPT.

## 9. USA-Housing.

The screenshot shows a web browser window titled "USA Housing". The page features a form with four input fields. On the left, there are fields for "Avg. Area Income" (empty), "Avg. Area House Age" (empty), and "Avg. Area Number of Rooms" (empty). On the right, there are fields for "Avg. Area Number of Bedrooms" (empty) and "Area Population" (empty). In the center, there is a large image of the United States Capitol building at night. At the bottom of the form is a blue "Submit" button. The browser's address bar shows the URL "52.90.81.182:8060". The taskbar at the bottom of the screen includes icons for various applications like Gmail, YouTube, Maps, Translate, News, and ChatGPT.

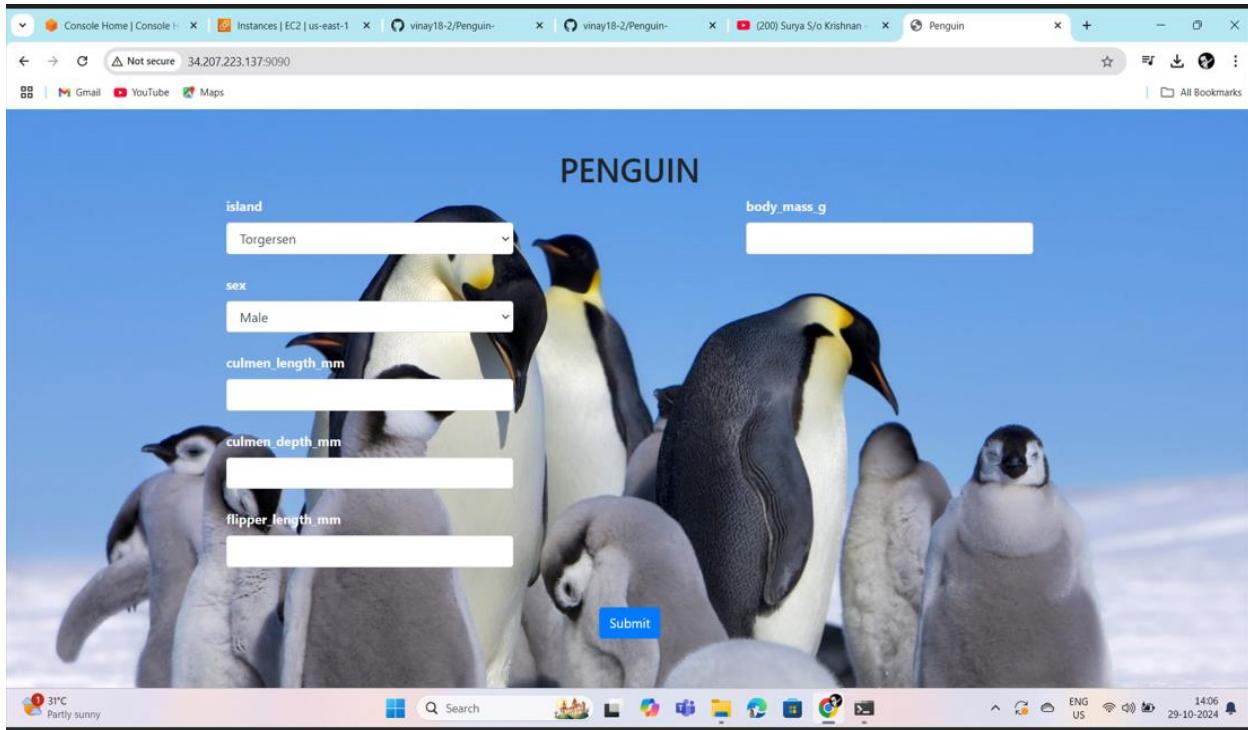
## 10. Fuel consumption.

The screenshot shows a web browser window titled "Fuel Consumption". On the left, there are several input fields for vehicle specifications: "Make" (Acura), "Engine Size", "Model", "Cylinders", "Vechile Class" (Compact), "CO2", "Transmission" (AM8), "SmokeRate", and "Fuel Type" (Z). To the right of these fields is a photograph of a Petro-Canada gas station with several fuel pumps. The browser's address bar shows the URL "52.90.81.182:5090". The taskbar at the bottom displays various icons and the date "04-11-2024".

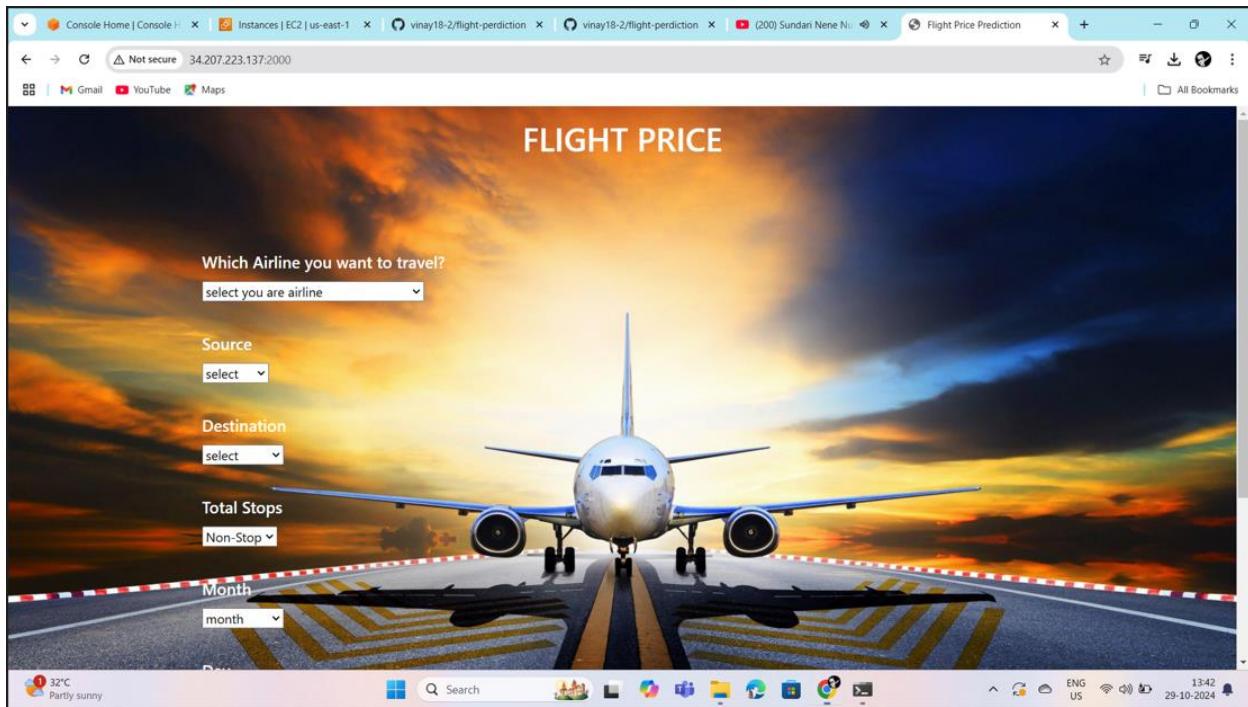
## 11. InNews.

The screenshot shows a web browser window titled "InNewsIN: A Summarised News". The main visual element is a large orange circle containing a white newspaper icon with the word "NEWS" on it. Below this is a dark blue header with the text "InNewsIN: A Summarised News" and a small news icon. Underneath the header is a dropdown menu labeled "Select your Category" with the placeholder "-Select-". Below the dropdown is a red rectangular button with the text "Please select Type!!". The browser's address bar shows the URL "52.90.81.182:8501". The taskbar at the bottom displays various icons and the date "04-11-2024".

## 12. Penguin.

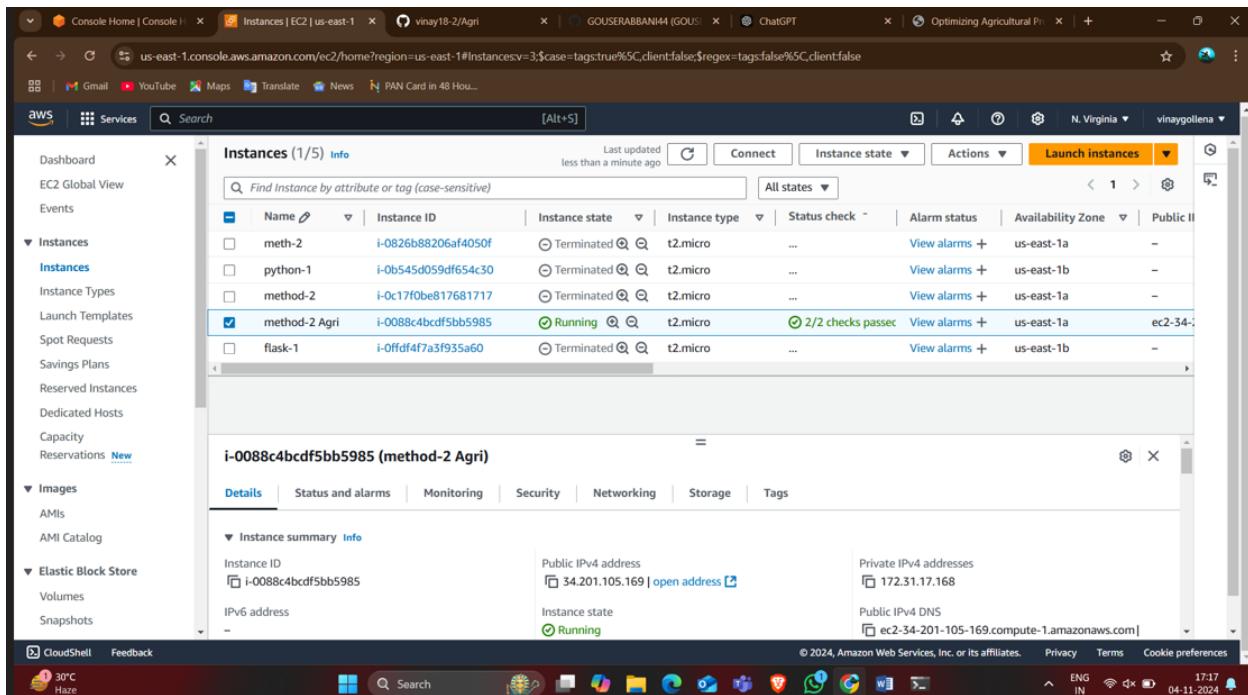


## 13. Flight prediction

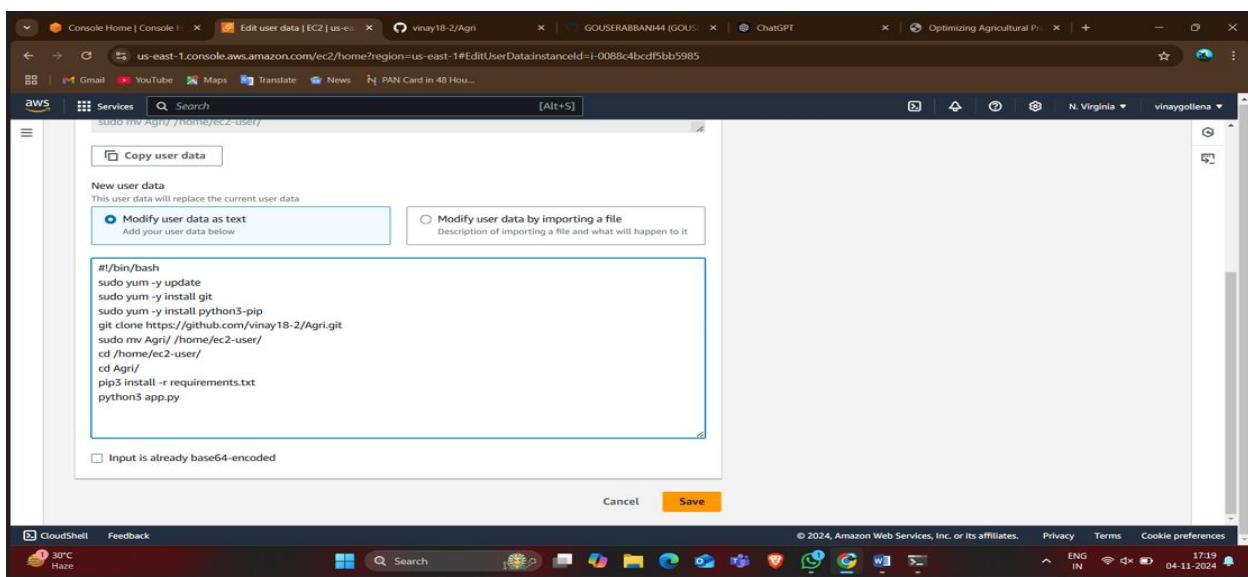


# METHOD-2: PYTHON APPLICATION DEPLOY USING USER DATA SCRIPT

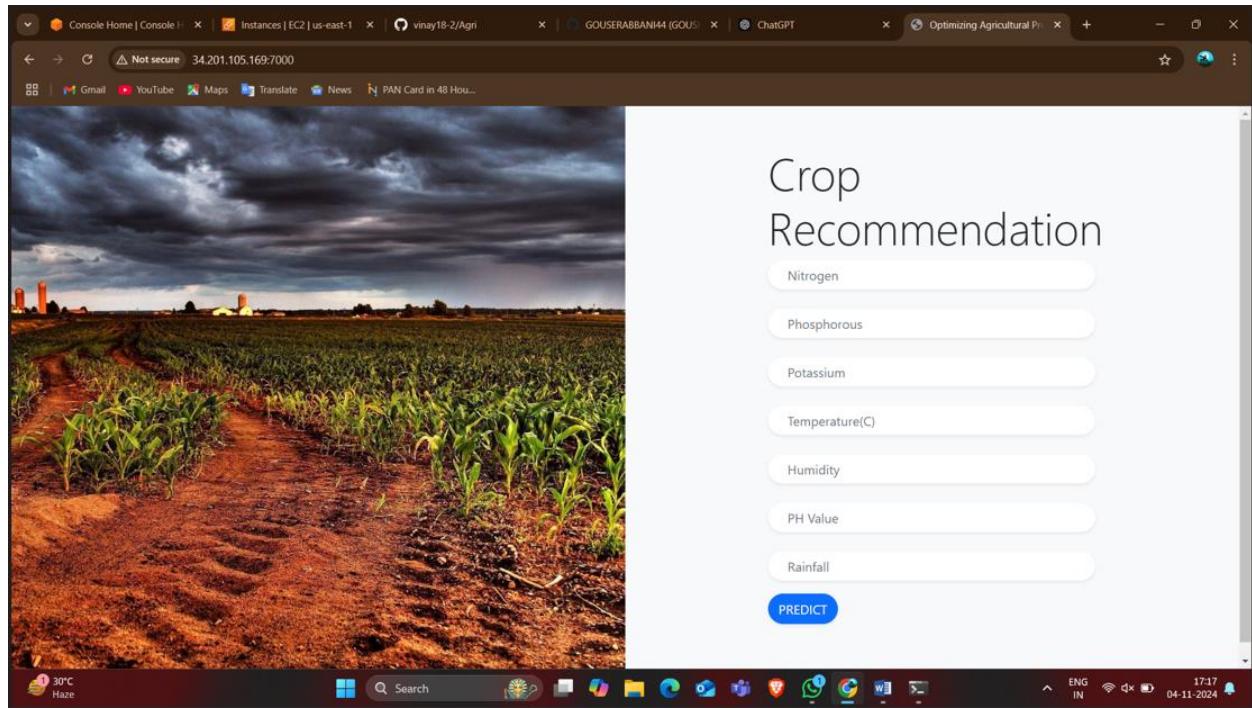
- ❖ Launch an instance using amazon linux and connect to the terminal.



- ❖ Here the bash script which I have write in the ec2 instance userdata while launching the instance.

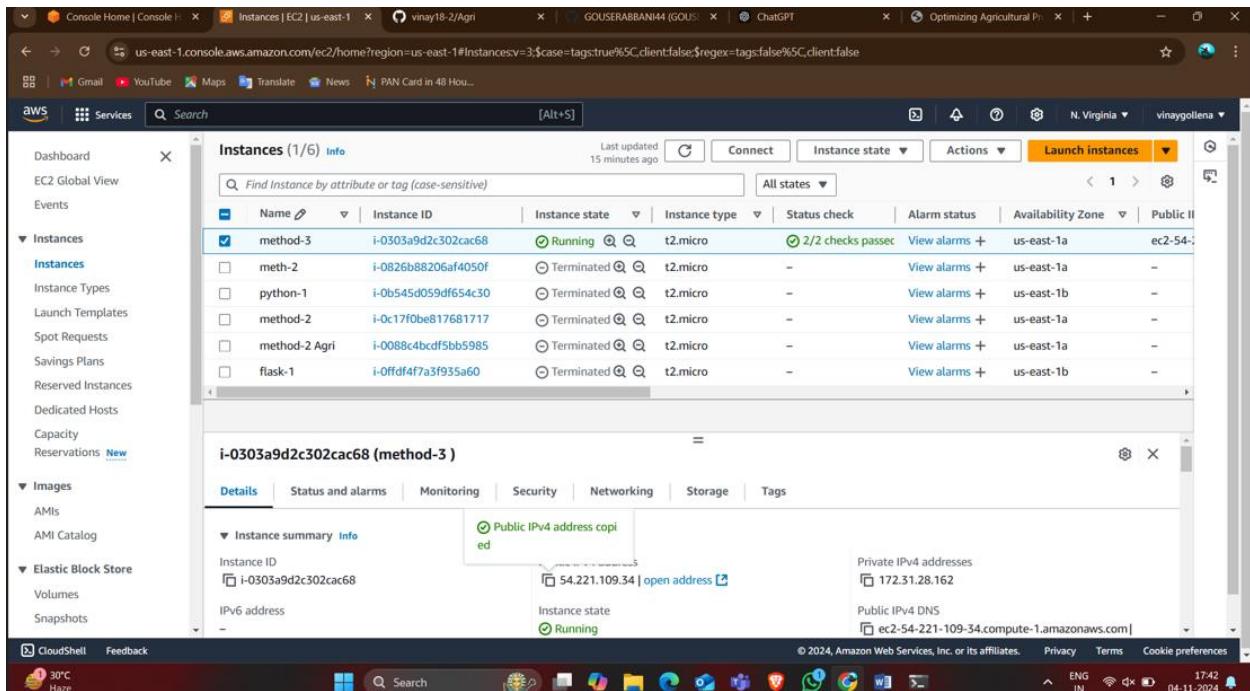


- ❖ Here the output I got while browsing the public ip address with port number in browser.



# METHOD-3 DEPLOYING PYTHON APPLICATION BY USING THE BASH SCRIPT

- ❖ Launch the instance using amazon linux and connect to the terminal.



- ❖ Update by using the command.

```
ec2-user@ip-172-31-28-162:~$ ssh -i "vinay.pem" ec2-user@ec2-54-221-109-34.compute-1.amazonaws.com
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd downloads

C:\Users\user\Downloads>ssh -i "vinay.pem" ec2-user@ec2-54-221-109-34.compute-1.amazonaws.com
The authenticity of host 'ec2-54-221-109-34.compute-1.amazonaws.com (54.221.109.34)' can't be established.
ED25519 key fingerprint is SHA256:ZOE3imb81hQezpJ9mpoytYYh/cavN6IhfKcHAWcJNsk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-54-221-109-34.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

    _\_\_ #####
    _\_\_ \###\_
    _\_\_ \|#\|
    _\_\_ \#`--->
    _\_\_ V`---->
          A newer version of Amazon Linux is available!
    _\_\_ /`-->
    _\_\_ /`-->
          Amazon Linux 2023, GA and supported until 2028-03-15.
          https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-28-162 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
--> Running transaction check
--> Package amazon-ss-agent.x86_64 0:3.3.859.0-1.amzn2 will be updated
--> Package amazon-ss-agent.x86_64 0:3.3.987.0-1.amzn2 will be an update
--> Package kernel.x86_64 0:5.10.227-219.884.amzn2 will be installed
--> Package libdwarf.x86_64 0:20130207-4.amzn2.0.2 will be updated
--> Package libelf.x86_64 0:20130207-4.amzn2.0.3 will be an update
--> Package openssl.x86_64 1:1.0.2k-24.amzn2.0.14 will be updated
--> Package openssl-libs.x86_64 1:1.0.2k-24.amzn2.0.13 will be updated
--> Package openssl-libs.x86_64 1:1.0.2k-24.amzn2.0.14 will be an update
--> Package python.x86_64 0:2.7.18-1.amzn2.0.8 will be updated
--> Package python.x86_64 0:2.7.18-1.amzn2.0.9 will be an update
--> Package python-devel.x86_64 0:2.7.18-1.amzn2.0.8 will be updated
--> Package python-devel.x86_64 0:2.7.18-1.amzn2.0.9 will be an update
--> Package python-idna.noarch 0:2.4-1.amzn2 will be updated
```

- ❖ Create a file for data by using command “sudo vi fname.sh”.
- ❖ Writhe the script and save it.

```
#!/bin/bash
sudo yum -y update
sudo yum -y install git
sudo yum -y install python3-pip
git clone https://github.com/vinay18-2/Agri.git
sudo mv Agri/ /home/ec2-user/
cd /home/ec2-user/Agri/
pip3 install -r requirements.txt
python3 app.py

"v3.sh" 10L, 238B
```

The terminal window shows the user executing commands to set up a Python application. It includes cloning a GitHub repository, changing directory, installing dependencies using pip3, and running the application script. The status bar at the bottom right indicates the date as 04-11-2024 and the time as 17:41.

- ❖ Give the permission to the scripted file and run the file.

```
sudo chmod 777 fname.sh (permission)
sudo ./fname.sh (to run data.sh files)
```

```
[ec2-user@ip-172-31-28-162: ~]$ sudo rm -rf agri.sh
[ec2-user@ip-172-31-28-162 ~]$ ll
total 0
drwxr-xr-x 5 root root 247 Nov  4 11:59 Agri
[ec2-user@ip-172-31-28-162 ~]$ sudo rm -rf Agri
[ec2-user@ip-172-31-28-162 ~]$ ll
total 0
[ec2-user@ip-172-31-28-162 ~]$ sudo vi v3.sh
[ec2-user@ip-172-31-28-162 ~]$ sudo chmod 777 v3.sh
[ec2-user@ip-172-31-28-162 ~]$ sudo ./v3.sh
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
No packages marked for update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package git-2.40.1-1.amzn2.0.3.x86_64 already installed and latest version
Nothing to do
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package python3-pip-20.2.2-1.amzn2.0.7.noarch already installed and latest version
Nothing to do
Cloning into 'Agri'...
remote: Enumerating objects: 10771, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (13/13), done.
remote: Total 10771 (delta 6), reused 7 (delta 2), pack-reused 10755 (from 1)
Receiving objects: 100% (10771/10771), 81.97 MiB | 23.85 MiB/s, done.
Resolving deltas: 100% (816/816), done.
Updating files: 100% (10267/10267), done.
mv: 'Agri/' and '/home/ec2-user/Agri' are the same file
WARNING: Running pip install with root privileges is generally not a good idea. Try 'pip3 install --user' instead.
Requirement already satisfied: click in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 1)) (8.1.7)
Requirement already satisfied: colorama in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 2)) (0.4.6)
```

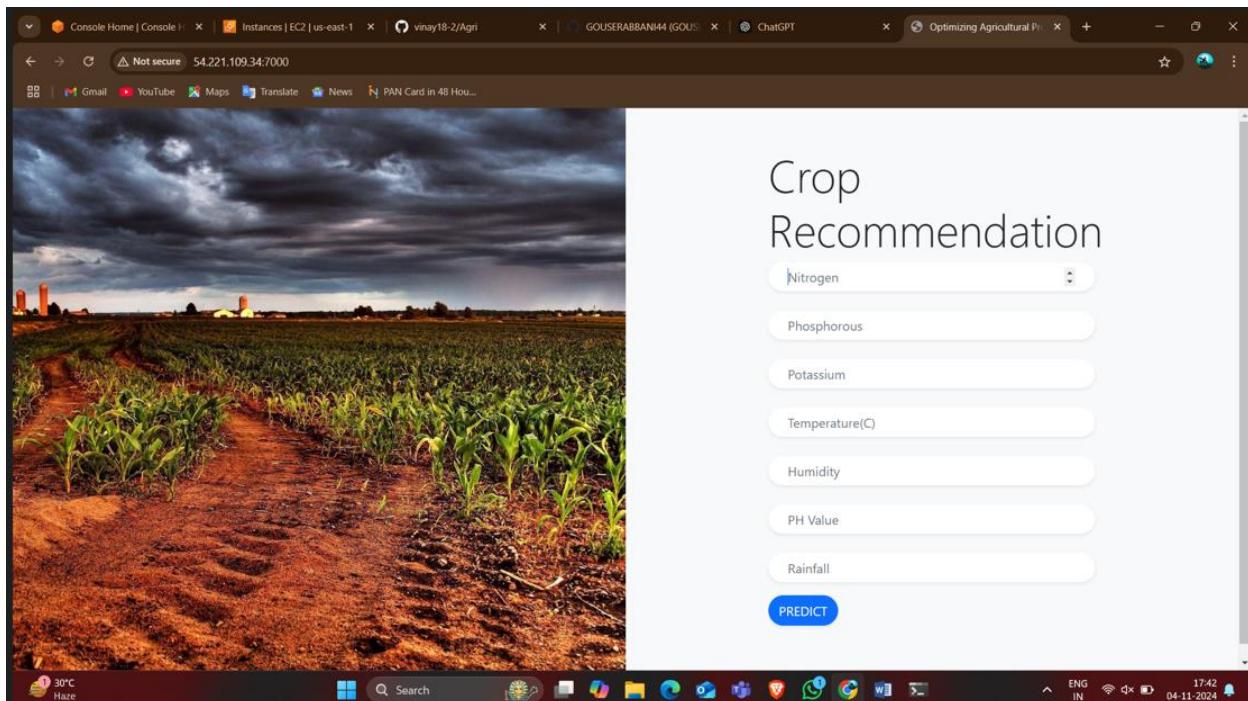
The terminal window shows the user deleting the 'Agri' directory, creating a new 'v3.sh' file, making it executable, and running it. The script appears to be a simple shell script. The status bar at the bottom right indicates the date as 04-11-2024 and the time as 17:41.

```

ec2-user@ip-172-31-28-162: ~ + %
Requirement already satisfied: importlib-metadata in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (6.7.0)
Requirement already satisfied: itsdangerous in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 5)) (2.1.2)
Requirement already satisfied: Jinja2 in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (3.1.4)
Requirement already satisfied: joblib in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (1.3.2)
Requirement already satisfied: MarkupSafe in /usr/local/lib64/python3.7/site-packages (from -r requirements.txt (line 8)) (2.1.5)
Requirement already satisfied: numpy in /usr/local/lib64/python3.7/site-packages (from -r requirements.txt (line 9)) (1.21.6)
Requirement already satisfied: pandas in /usr/local/lib64/python3.7/site-packages (from -r requirements.txt (line 10)) (1.3.5)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 11)) (2.9.0.post0)
Requirement already satisfied: pytz in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 12)) (2024.2)
Requirement already satisfied: scikit-learn in /usr/local/lib64/python3.7/site-packages (from -r requirements.txt (line 13)) (1.0.2)
Requirement already satisfied: scipy in /usr/local/lib64/python3.7/site-packages (from -r requirements.txt (line 14)) (1.7.3)
Requirement already satisfied: six in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 15)) (1.16.0)
Requirement already satisfied: threadpoolctl in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 17)) (3.1.0)
Requirement already satisfied: Werkzeug in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 18)) (2.2.3)
Requirement already satisfied: zipp in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 19)) (3.15.0)
Requirement already satisfied: gunicorn in /usr/local/lib/python3.7/site-packages (from -r requirements.txt (line 20)) (23.0.0)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /usr/local/lib/python3.7/site-packages (from importlib-metadata->-r requirements.txt (line 4)) (4.7.1)
Requirement already satisfied: packaging in /usr/local/lib/python3.7/site-packages (from gunicorn->-r requirements.txt (line 20)) (24.0)
/usr/local/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 http://172.31.28.162:7000
Press CTRL+C to quit
* Restarting with stat
/usr/local/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,
* Debugger is active!
* Debugger PIN: 626-773-229
103.55.213.30 - - [04/Nov/2024 12:11:03] "GET / HTTP/1.1" 200 -
103.55.213.30 - - [04/Nov/2024 12:11:03] "GET /static/styles.css HTTP/1.1" 404 -
103.55.213.30 - - [04/Nov/2024 12:11:04] "GET /favicon.ico HTTP/1.1" 404 -

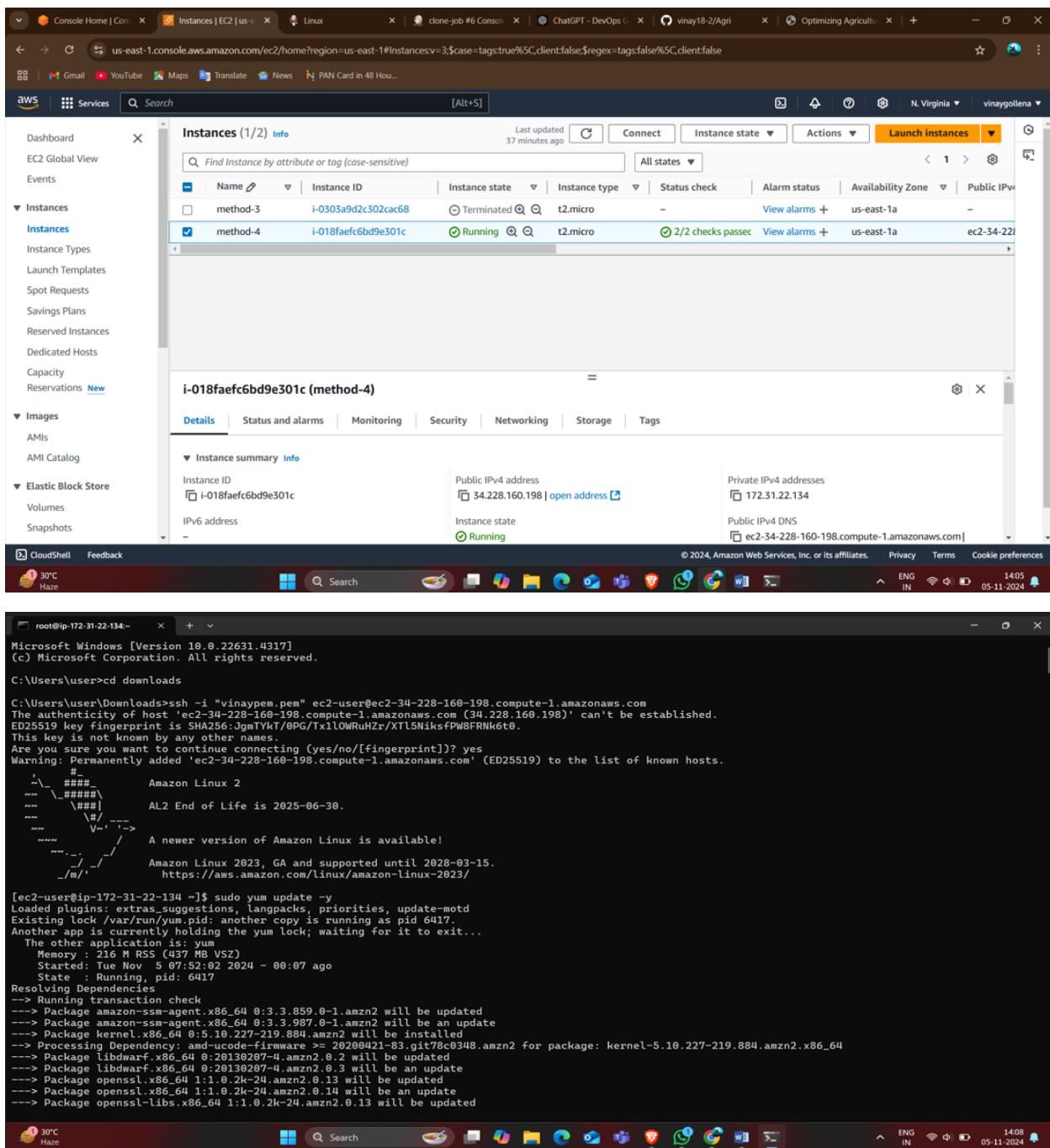
```

❖ Than search the public ip address woth port number .



## METHOD-4 Build and deploy python applications with Git, Github and Jenkins (execute shell)

- ❖ Launch the instance with the Amazon Linux and connect to the terminal.



- ❖ Than install the Jenkins in the terminal by searching install Jenkins in the google.

```
sudo wget -O /etc/yum.repos.d/jenkins.repo \
```

```
https://pkg.jenkins.io/redhat-stable/jenkins.repo
```

```
sudo rpm --import https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key
```

```
sudo yum upgrade
```

```
# Add required dependencies for the jenkins package
```

```
sudo yum install fontconfig java-17-openjdk
```

```
sudo yum install jenkins
```

```

root@ip-172-31-22-134: ~ + ~
Cleanup : amazon-ssm-agent-3.3.859.0-1.amzn2.x86_64 16/24
Cleanup : python3-libs-3.7.16-1.amzn2.0.6.x86_64 17/24
Cleanup : python3-pip-20.2.2-1.amzn2.0.6.noarch 18/24
Cleanup : python3-3.7.16-1.amzn2.0.6.x86_64 19/24
Cleanup : python-2.7.18-1.amzn2.0.8.x86_64 20/24
Cleanup : python-libs-2.7.18-1.amzn2.0.8.x86_64 21/24
Cleanup : 1:openssl-1.0.2k-24.amzn2.0.13.x86_64 22/24
Cleanup : 1:openssl-libs-1.0.2k-24.amzn2.0.13.x86_64 23/24
Cleanup : libdwarf-20130207-4.amzn2.0.2.x86_64 24/24
Verifying : libdwarf-20130207-4.amzn2.0.3.x86_64 1/24
Verifying : python-libs-2.7.18-1.amzn2.0.9.x86_64 2/24
Verifying : python3-3.7.16-1.amzn2.0.7.x86_64 3/24
Verifying : python-2.7.18-1.amzn2.0.9.x86_64 4/24
Verifying : 1:openssl-libs-1.0.2k-24.amzn2.0.14.x86_64 5/24
Verifying : kernel-5.10.227-219.884.amzn2.x86_64 6/24
Verifying : python-devel-2.7.18-1.amzn2.0.9.x86_64 7/24
Verifying : python3-pip-20.2.2-1.amzn2.0.7.noarch 8/24
Verifying : amazon-ssm-agent-3.3.987.0-1.amzn2.x86_64 9/24
Verifying : python3-libs-3.7.16-1.amzn2.0.7.x86_64 10/24
Verifying : python-idna-2.4-1.amzn2.0.1.noarch 11/24
Verifying : 1:openssl-1.0.2k-24.amzn2.0.14.x86_64 12/24
Verifying : libdwarf-20130207-4.amzn2.0.8.x86_64 13/24
Verifying : python-devel-2.7.18-1.amzn2.0.8.x86_64 14/24
Verifying : python3-pip-20.2.2-1.amzn2.0.6.noarch 15/24
Verifying : python-libs-2.7.18-1.amzn2.0.8.x86_64 16/24
Verifying : python-2.7.18-1.amzn2.0.8.x86_64 17/24
Verifying : amazon-ssm-agent-3.3.859.0-1.amzn2.x86_64 18/24
Verifying : 1:openssl-1.0.2k-24.amzn2.0.13.x86_64 19/24
Verifying : libdwarf-20130207-4.amzn2.0.2.x86_64 20/24
Verifying : python-idna-2.4-1.amzn2.noarch 21/24
Verifying : python3-libs-3.7.16-1.amzn2.0.6.x86_64 22/24
Verifying : 1:openssl-libs-1.0.2k-24.amzn2.0.13.x86_64 23/24
Verifying : python3-3.7.16-1.amzn2.0.6.x86_64 24/24

Installed:
  kernel.x86_64 0:5.10.227-219.884.amzn2

Dependency Installed:
  and-ucode-firmware.noarch 0:20200421-83.git78c0348.amzn2

Updated:

```

- ❖ Before installing the Jenkins or run the sudo yum install Jenkins we have to install the run time for Jenkins i.e “java or java-17” version by using the command “sudo yum install java-17\* -y”.

- ❖ Than after that install the Jenkins and start the Jenkins, enable the Jenkins and check the status of the Jenkins.

Sudo systemctl start Jenkins

Sudo systemctl enable Jenkins

Sudo systemctl status Jenkins

- ❖ And also edit the visudo file by using “sudo visudo” go to the down and write one line their i.e Jenkins ALL=(ALL) NOPASSWD=ALL”

- ❖ And restart the Jenkins.

```

# Adding HOME to env_keep may enable a user to run unrestricted
# commands via sudo.
#
# Defaults    env_keep += "HOME"
Defaults    secure_path = /sbin:/bin:/usr/sbin:/usr/bin

## Next comes the main part: which users can run what software on
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##     user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root      ALL=(ALL)          ALL

## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS

## Allows people in group wheel to run all commands.
%wheel  ALL=(ALL)          ALL

## Same thing without a password
# %wheel    ALL=(ALL)        NOPASSWD: ALL
jenkins   ALL=(ALL)        NOPASSWD: ALL
## Allows members of the users group to mount and umount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
# %users  localhost=/sbin/shutdown -h now

## Read drop-in files from /etc/sudoers.d (the # here does not mean a comment)
#includedir /etc/sudoers.d
-- INSERT --
111,33      Be

 30°C Haze          Q Search          13:55 05-11-2024 ENG IN WiFi 111,33 Be

 30°C Haze          Q Search          13:55 05-11-2024 ENG IN WiFi 111,33 Be

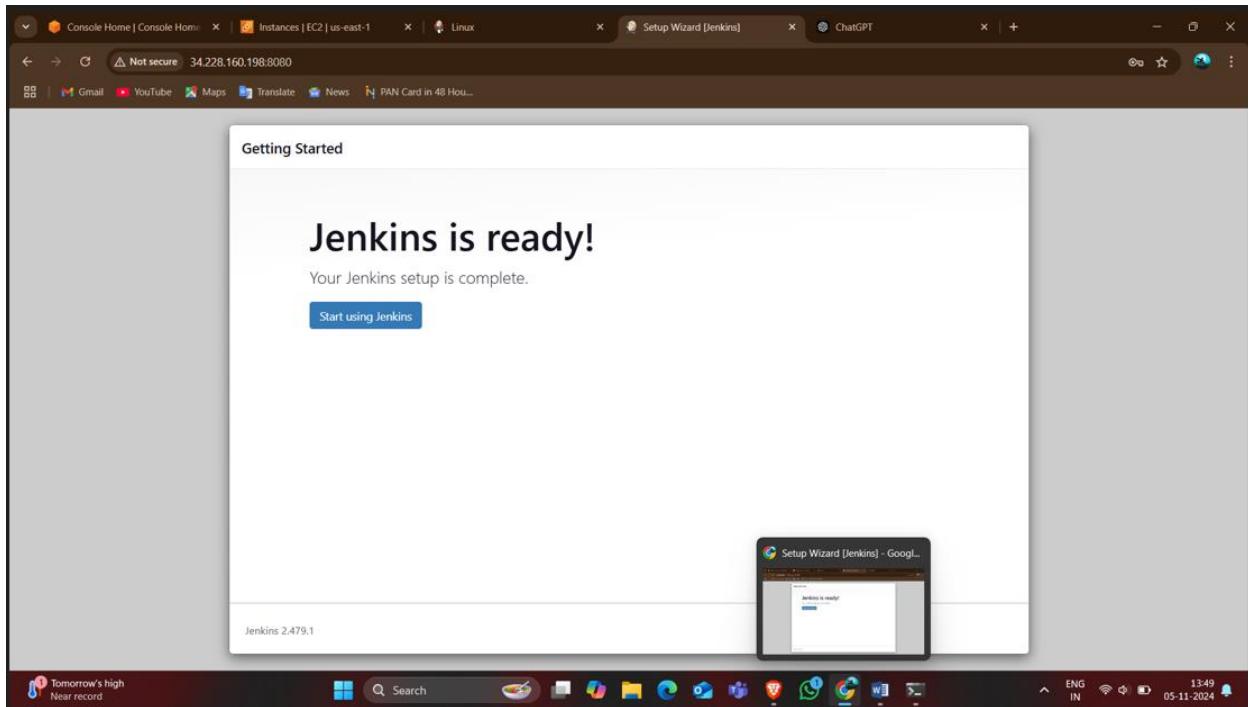
root@ip-172-31-22-134:~ % + v
--> Finished Dependency Resolution

Dependencies Resolved

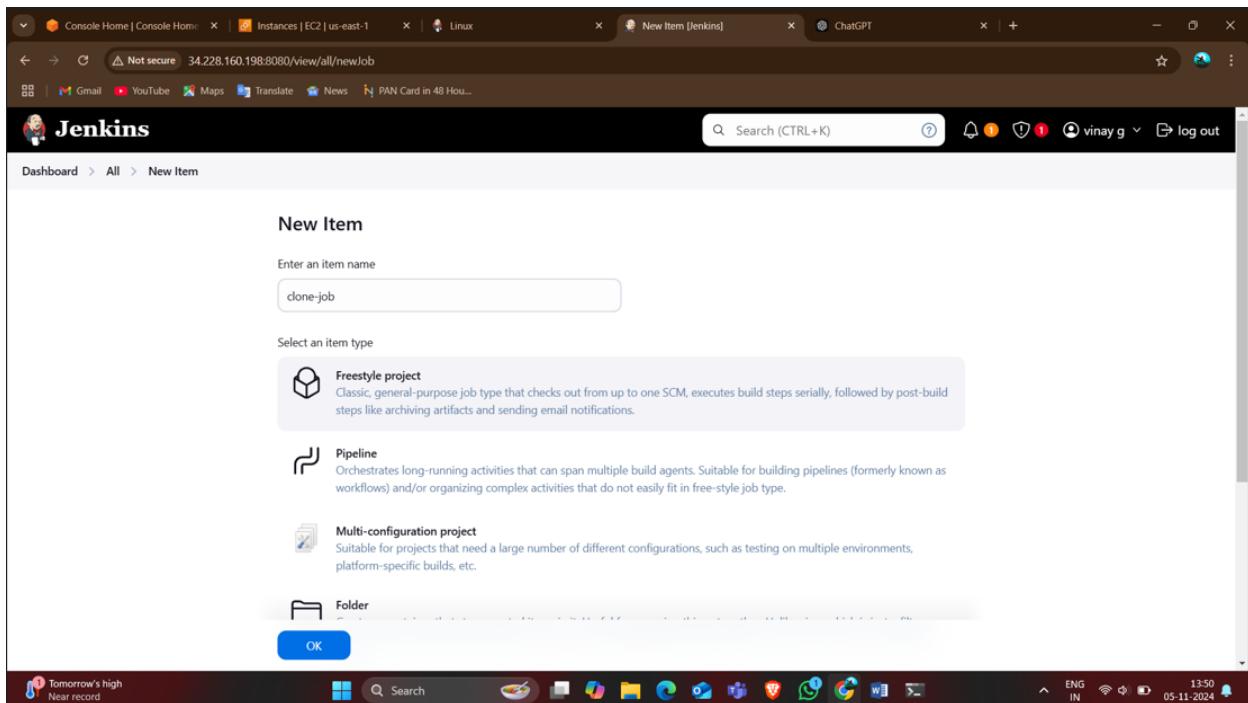
=====
| Package           | Arch       | Version      | Repository | Size |
|=====             | =====     | =====       | =====      | ===== |
| Installing:      |            |             |            |       |
| jenkins          | noarch    | 2.479.1-1.1 | jenkins    | 91 M  |
| Transaction Summary
|=====
|=====             |             |             |             |       |
| Install 1 Package
|=====             |             |             |             |       |
| Total download size: 91 M
| Installed size: 92 M
| Is this ok [y/d/N]: y
| Downloading packages:
| jenkins-2.479.1-1.1.noarch.rpm
| Running transaction check
| Running transaction test
| Transaction test succeeded
| Running transaction
|   Installing : jenkins-2.479.1-1.1.noarch
|   Verifying   : jenkins-2.479.1-1.1.noarch
|=====
|=====             |             |             |             |       |
| Installed:
| jenkins.noarch 0:2.479.1-1.1
| Complete!
[ec2-user@ip-172-31-22-134 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-22-134 ~]$ sudo systemctl enable jenkins
Created symlink from /etc/systemd/system/multi-user.target.wants/jenkins.service to /usr/lib/systemd/system/jenkins.service.
[ec2-user@ip-172-31-22-134 ~]$ sudo systemctl status jenkins
● jenkins.service - Jenkins Continuous Integration Server
  Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
  Active: active (running) since Tue 2024-11-05 07:57:50 UTC; 18s ago
    Main PID: 10227 (java)
   CGroup: /system.slice/jenkins.service
           └─10227 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080
 14:08 05-11-2024 ENG IN WiFi 111,33 Be

```

- ❖ Than browse the public ip of instance with port 8080 in the browser and access it.
- ❖ Login the Jenkins page.



- ❖ Now create a new job i.e clone-job to clone the repo from github.



- ❖ Now select the source as git and give the repo link in that.

The screenshot shows the Jenkins job configuration page for a job named "clone-job". The "Source Code Management" section is selected, showing the "Git" option is chosen. The "Repository URL" field contains "https://github.com/vinay18-2/Agri.git". A red error message below the URL field says "Please enter Git repository." The Jenkins interface includes a navigation bar at the top and a taskbar at the bottom.

- ❖ Write the below commands in the execute shell in build steps.

The screenshot shows the Jenkins job configuration page for a job named "clone-job". The "Build Steps" section is selected, showing a large text area for entering commands. The commands listed are:

```

#!/bin/bash
# Update and install pip for Python 3
sudo yum -y install python3-pip

# Move the Agri directory to the target location
sudo mv Agri/* /home/ec2-user/

# Navigate to the Agri directory
cd /home/ec2-user/Agri/

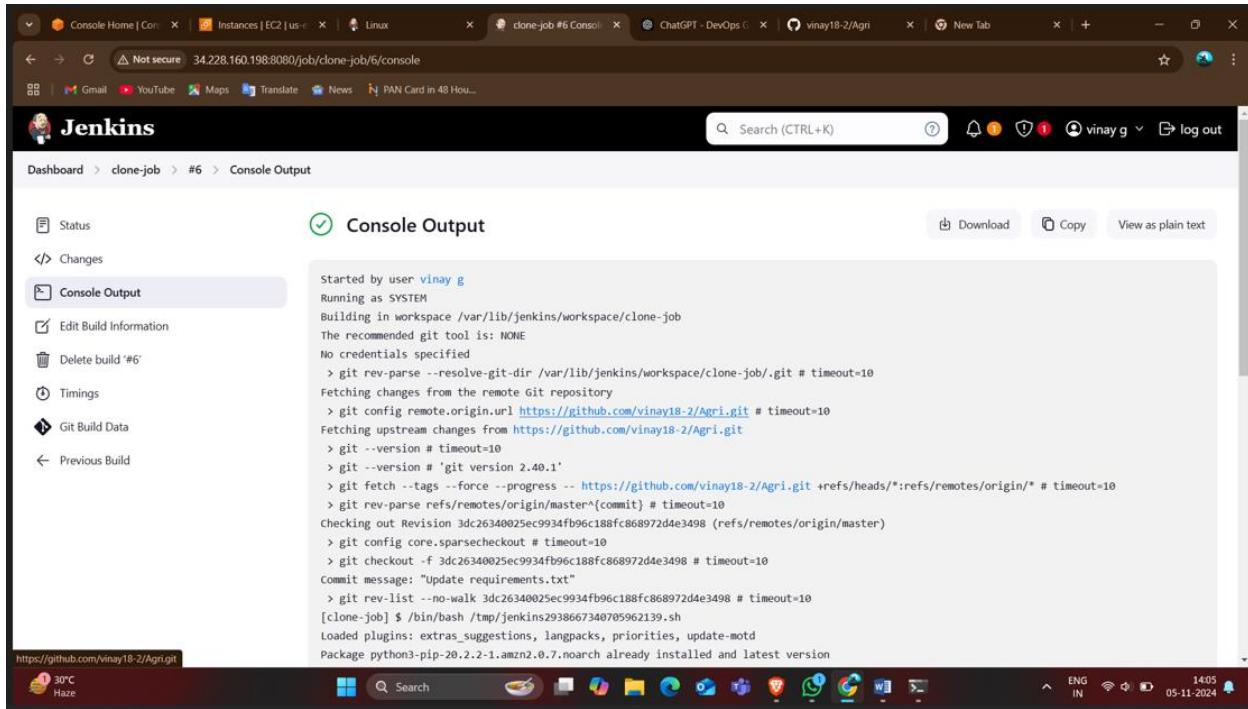
# Install the dependencies listed in requirements.txt
sudo pip3 install -r requirements.txt

# Run the app
screen -m -d python3 app.py

```

The Jenkins interface includes a navigation bar at the top and a taskbar at the bottom.

- ❖ Than save it and build the job, after than if build is success than it get in green in colour.



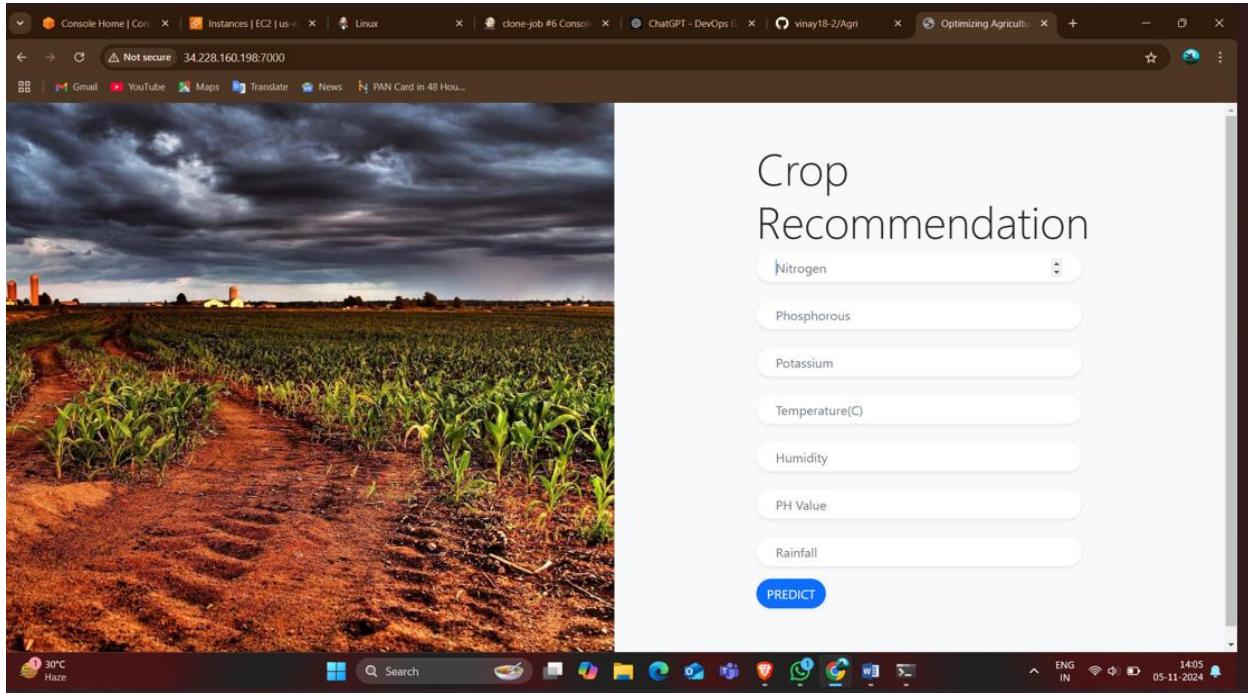
The screenshot shows a Jenkins console output window. The log starts with "Started by user vinay g" and "Running as SYSTEM". It details the git cloning process from a GitHub repository, including fetching upstream changes and checking out a specific revision. The log ends with "Package python3-pip-20.2.2-1.amzn2.0.7.noarch already installed and latest version". The Jenkins interface includes a sidebar with options like Status, Changes, and Console Output, and a navigation bar at the top.

```

Started by user vinay g
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/clone-job
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/clone-job/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/vinay18-2/Agri.git # timeout=10
Fetching upstream changes from https://github.com/vinay18-2/Agri.git
> git --version # timeout=10
> git --version # 'git' version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/vinay18-2/Agri.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master{commit} # timeout=10
Checking out Revision 3dc26340025ec9934fb96c188fc868972d4e3498 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f 3dc26340025ec9934fb96c188fc868972d4e3498 # timeout=10
Commit message: "Update requirements.txt"
> git rev-list --no-walk 3dc26340025ec9934fb96c188fc868972d4e3498 # timeout=10
[clone-job] $ /bin/bash /tmp/jenkins2938667340705962139.sh
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package python3-pip-20.2.2-1.amzn2.0.7.noarch already installed and latest version

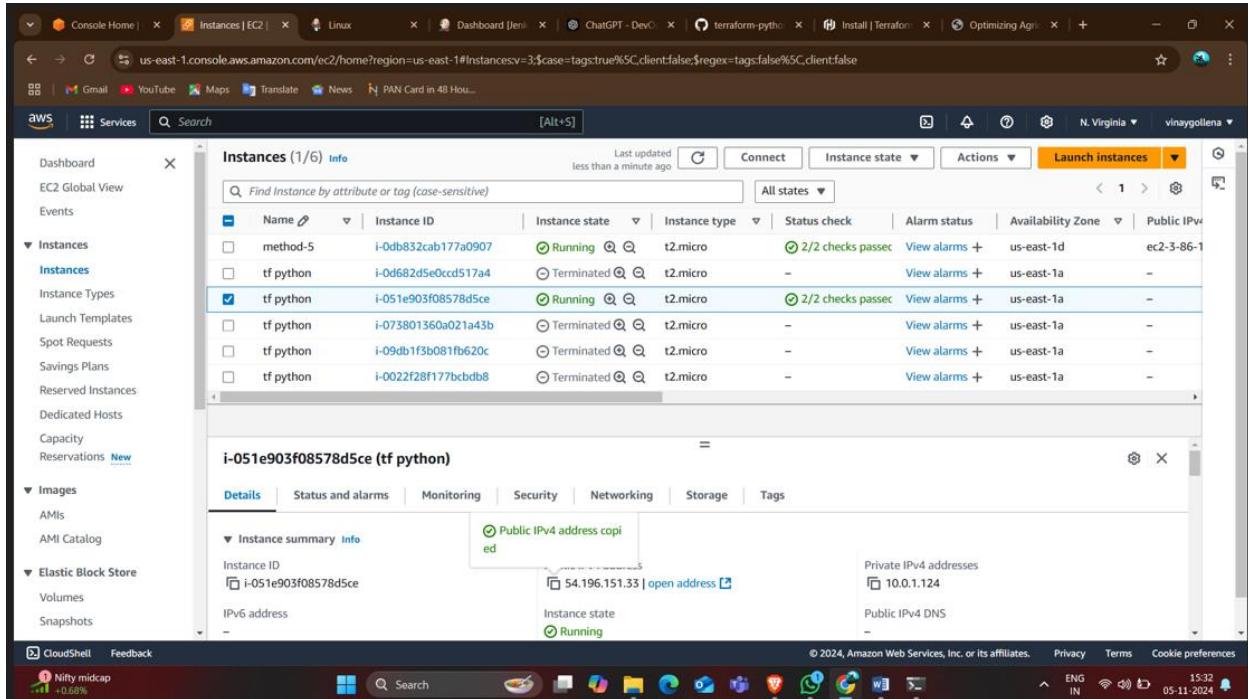
```

- ❖ Than browse the ip address with port provided in the application.
- ❖ Observe the results.



# METHOD-5: Build and deploy python applications with the Terraform(data.sh) and push the Terraform Scripted files in Github

- ❖ Launch the instance using amazon.



- ❖ Connect to the terminal and update it.
- ❖ Install git and terraform .
- ❖ Clone the github link which the terraform scripts have in that.
- ❖ Or else write then hcl script to create the instance.vpc,route tables,subnets and security groups,etc.
- ❖ Then provide the terraform commands  
terraform init  
terraform validate  
terraform plan  
terraform apply –auto-approve
- ❖ We can observe the below images what happens after running this commands.

```
[ec2-user@ip-172-31-87-159 ~] [ec2-user@ip-10-0-1-124 ~] + 
Destroy complete! Resources: 17 destroyed.
[ec2-user@ip-172-31-87-159 terraform=python]$ ll
total 72
-rw-rw-r-- 1 ec2-user ec2-user 1740 Nov  5 09:47 errored.tfstate
-rw-rw-r-- 1 ec2-user ec2-user 114 Nov  5 09:05 igw.tf
-rw-rw-r-- 1 ec2-user ec2-user 397 Nov  5 09:05 instance.tf
-rw-rw-r-- 1 ec2-user ec2-user 239 Nov  5 09:45 python-data.sh
-rw-rw-r-- 1 ec2-user ec2-user 18 Nov  5 09:05 README.md
-rw-rw-r-- 1 ec2-user ec2-user 1434 Nov  5 09:05 routetable.tf
-rw-rw-r-- 1 ec2-user ec2-user 629 Nov  5 09:05 sg.tf
-rw-rw-r-- 1 ec2-user ec2-user 1796 Nov  5 09:05 subnets.tf
-rw-rw-r-- 1 ec2-user ec2-user 182 Nov  5 09:56 terraform.tfstate
-rw-rw-r-- 1 root      root    27839 Nov  5 09:55 terraform.tfstate.backup
-rw-rw-r-- 1 ec2-user ec2-user 695 Nov  5 09:05 variable.tf
-rw-rw-r-- 1 ec2-user ec2-user 303 Nov  5 09:13 vpc.tf
[ec2-user@ip-172-31-87-159 terraform=python]$ sudo vi python-data.sh

[ec2-user@ip-172-31-87-159 terraform=python]$ terraform init
Initializing the backend...
Initializing provider plugins...
- Reusing previous version of hashicorp/aws from the dependency lock file
- Using previously-installed hashicorp/aws v5.74.0

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,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
[ec2-user@ip-172-31-87-159 terraform=python]$ terraform validate
Success! The configuration is valid.

[ec2-user@ip-172-31-87-159 terraform=python]$ terraform plan

Terraform used the selected providers to generate the following execution plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

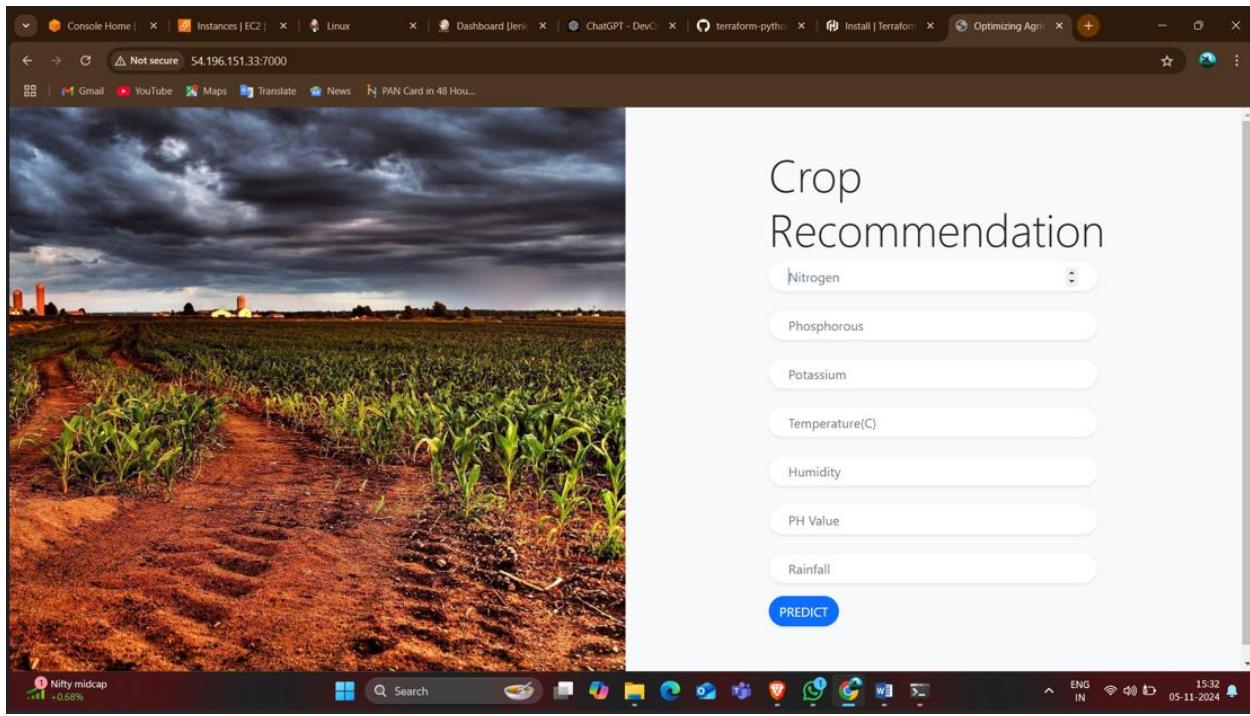

```

```
ec2-user@ip-172-31-87-159: ~ ec2-user@ip-10-0-1-124: ~ + ~
aws_vpc.vinayvpc: Creation complete after 2s [id=vpc-05428fb7ef834240b]
aws_subnet.vinaysubnet1: Creating...
aws_subnet.vinaysubnet3: Creating...
aws_subnet.vinaysubnet4: Creating...
aws_subnet.vinayprsubnet5: Creating...
aws_security_group.vinaysg: Creating...
aws_subnet.vinaysubnet2: Creating...
aws_subnet.vinayprsubnet6: Creating...
aws_internet_gateway.vinaygateway: Creating...
aws_internet_gateway.vinaygateway: Creation complete after 0s [id=igw-048855b62a8981e46]
aws_route_table.vinayroute: Creating...
aws_subnet.vinayprsubnet6: Creation complete after 0s [id=subnet-04eldb7838b81b97b]
aws_subnet.vinayprsubnet5: Creation complete after 0s [id=subnet-0f84f6feef97fbe145]
aws_route_table.vinayroute: Creation complete after 1s [id=rtb-043fdbda7f3831b9f0]
aws_route_table_association.vinayrt5: Creating...
aws_route_table_association.vinayrt6: Creating...
aws_route_table_association.vinayrt6: Creation complete after 0s [id=rtbassoc-0f0184d3c987cdfe9]
aws_route_table_association.vinayrt5: Creation complete after 0s [id=rtbassoc-0abe2c9860b65ded7]
aws_security_group.vinaysg: Creation complete after 2s [id=sg-0fb83acc3be2fee03]
aws_subnet.vinaysubnet1: Still creating... [10s elapsed]
aws_subnet.vinaysubnet3: Still creating... [10s elapsed]
aws_subnet.vinaysubnet4: Still creating... [10s elapsed]
aws_subnet.vinaysubnet2: Still creating... [10s elapsed]
aws_subnet.vinaysubnet2: Creation complete after 11s [id=subnet-039bac137a624a6f2]
aws_route_table_association.vinayrt2: Creating...
aws_subnet.vinaysubnet3: Creation complete after 11s [id=subnet-0e6db0ae6bb118a9a]
aws_route_table_association.vinayrt3: Creating...
aws_subnet.vinaysubnet4: Creation complete after 11s [id=subnet-0fe31fa5552e3ea54]
aws_subnet.vinaysubnet1: Creation complete after 11s [id=subnet-0fd7e4a60328d37df]
aws_route_table_association.vinayrt4: Creating...
aws_instance.vinay-1: Creating...
aws_route_table_association.vinayrt1: Creating...
aws_route_table_association.vinayrt2: Creation complete after 0s [id=rtbassoc-0c23b2264c9ee6d44]
aws_route_table_association.vinayrt3: Creation complete after 0s [id=rtbassoc-07ef6ff1b4fd76cc7]
aws_route_table_association.vinayrt1: Creation complete after 0s [id=rtbassoc-0d681d44471b4c252]
aws_route_table_association.vinayrt4: Creation complete after 0s [id=rtbassoc-0cda31dc32091d57b]
aws_instance.vinay-1: Still creating... [10s elapsed]
aws_instance.vinay-1: Creation complete after 12s [id=i-051e903f08578d5ce]

Apply complete! Resources: 17 added, 0 changed, 0 destroyed.
[ec2-user@ip-172-31-87-159 terraform-python]$ |
```

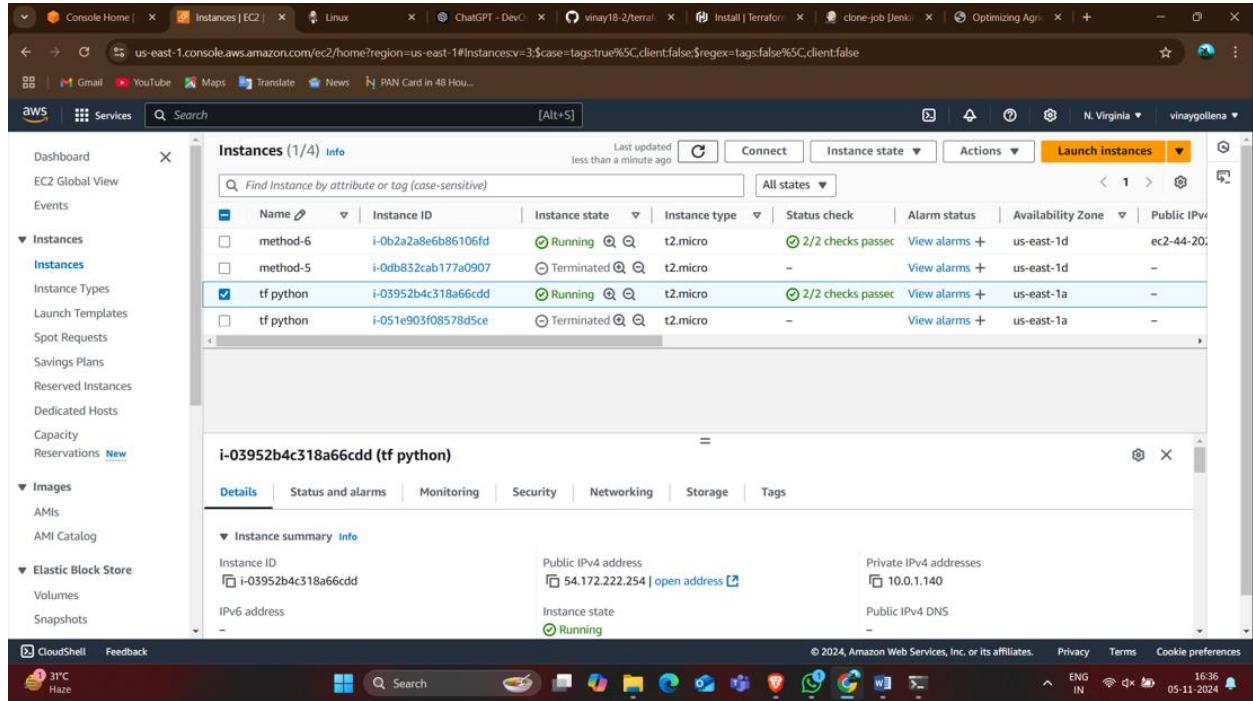
- ❖ After successfully applying this resources than go to the amazon consol and observe it created or not.
  - ❖ Browse the public ip of the created instance with given port number.

- ❖ We can observe the output.



# METHOD-6 : Build and deploy python applications with Git, github, Jenkins and Terraform

- ❖ Launch an instance with amazon linux.



- ❖ Connect to the terminal and install git.
- ❖ Than install Jenkins with the run time java.
- ❖ And tha start and enable the Jenkins.
- ❖ Edit the visudo file.
- ❖ Next restart the Jenkins.

```

ec2-user@ip-172-31-83-131:~ ec2-user@ip-10-0-1-140:~ + ~
CGroup: /system.slice/jenkins.service
└─10053 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Nov 05 10:26:14 ip-172-31-83-131.ec2.internal jenkins[10053]: ****
Nov 05 10:26:14 ip-172-31-83-131.ec2.internal jenkins[10053]: ****
Nov 05 10:26:14 ip-172-31-83-131.ec2.internal jenkins[10053]: ****
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.383+0000 [id=31] INFO jenkins.InitReactorRunner$1#...ization
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.405+0000 [id=24] INFO hudson.lifecycle.Lifecycle#0...running
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.476+0000 [id=47] INFO h.m.DownloadService$Download...staller
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.477+0000 [id=47] INFO hudson.util.Retriger#start: P...empt #1
Nov 05 10:26:28 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:16] Unknown lvalue 'StartLimitBurst' in section 'Unit'
Nov 05 10:26:28 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:17] Unknown lvalue 'StartLimitIntervalSec' in s... 'Unit'
Hint: Some lines were ellipsized, use -l to show in full.
[jec2-user@ip-172-31-83-131 ~]$ sudo systemctl start jenkins
[jec2-user@ip-172-31-83-131 ~]$ sudo systemctl enable jenkins
[jec2-user@ip-172-31-83-131 ~]$ sudo systemctl status jenkins
● Jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; enabled; vendor preset: disabled)
     Active: active (running) since Tue 2024-11-05 10:26:21 UTC; 1min 16s ago
       Main PID: 10053 (java)
      CGroup: /system.slice/jenkins.service
           └─10053 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Nov 05 10:26:14 ip-172-31-83-131.ec2.internal jenkins[10053]: ****
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.383+0000 [id=31] INFO jenkins.InitReactorRunner$1#...ization
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.405+0000 [id=24] INFO hudson.lifecycle.Lifecycle#0...running
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.476+0000 [id=47] INFO h.m.DownloadService$Download...staller
Nov 05 10:26:21 ip-172-31-83-131.ec2.internal jenkins[10053]: 2024-11-05 10:26:21.477+0000 [id=47] INFO hudson.util.Retriger#start: P...empt #1
Nov 05 10:26:28 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:16] Unknown lvalue 'StartLimitBurst' in section 'Unit'
Nov 05 10:26:28 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:17] Unknown lvalue 'StartLimitIntervalSec' in s... 'Unit'
Nov 05 10:27:34 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:16] Unknown lvalue 'StartLimitBurst' in section 'Unit'
Nov 05 10:27:34 ip-172-31-83-131.ec2.internal systemd[1]: [/usr/lib/systemd/system/jenkins.service:17] Unknown lvalue 'StartLimitIntervalSec' in s... 'Unit'
Hint: Some lines were ellipsized, use -l to show in full.
[jec2-user@ip-172-31-83-131 ~]$ sudo cat /var/lib/jenkins/secrets/initialAdminPassword
4eulaa69e624c6493a936434b3ba3ba
[jec2-user@ip-172-31-83-131 ~]$ 
[jec2-user@ip-172-31-83-131 ~]$ sudo visudo
[jec2-user@ip-172-31-83-131 ~]$ 
[jec2-user@ip-172-31-83-131 ~]$ sudo systemctl restart jenkins
[jec2-user@ip-172-31-83-131 ~]$ 

```

- ❖ After that browse the ip address:8080 and logon to the Jenkins page.
- ❖ Create a new job and provide the required information and build the job.
- ❖ Install the aws credential plugin to store the aws access keys and secret key.

Jenkins

Dashboard > Manage Jenkins > Plugins

**Plugins**

- Updates
- Available plugins
- Installed plugins
- Advanced settings
- Download progress

**Download progress**

Preparation	
• Checking internet connectivity	Success
• Checking update center connectivity	Success
• Success	Success
Amazon Web Services SDK : Minimal	Success
Amazon Web Services SDK : EC2	Success
AWS Credentials	Success
Loading plugin extensions	Success

→ Go back to the top page  
(you can start using the installed plugins right away)

→  Restart Jenkins when installation is complete and no jobs are running

REST API Jenkins 2.479.1

The screenshot shows the Jenkins 'Configuration' page for a 'clone-job'. On the left sidebar, under 'Build Environment', the 'AWS access key and secret' section is selected. It contains fields for 'Access Key Variable' (AWS\_ACCESS\_KEY\_ID) and 'Secret Key Variable' (AWS\_SECRET\_ACCESS\_KEY), both set to empty. Below these are 'Credentials' options, with 'Specific credentials' selected and a dropdown containing the value 'AKIA4MTWNVJ3F6FPZO7'. At the bottom are 'Save' and 'Apply' buttons.

- ❖ Provide the commands to install and run the terraform in the execute shell.

The screenshot shows the Jenkins 'Configuration' page for a 'clone-job'. On the left sidebar, under 'Build Steps', the 'Execute shell' step is selected. The 'Script' field contains a script to install yum-utils, add Hashicorp's repository, install Terraform, navigate to the configuration directory, initialize Terraform, format and validate files, create a plan, and apply it. The script is as follows:

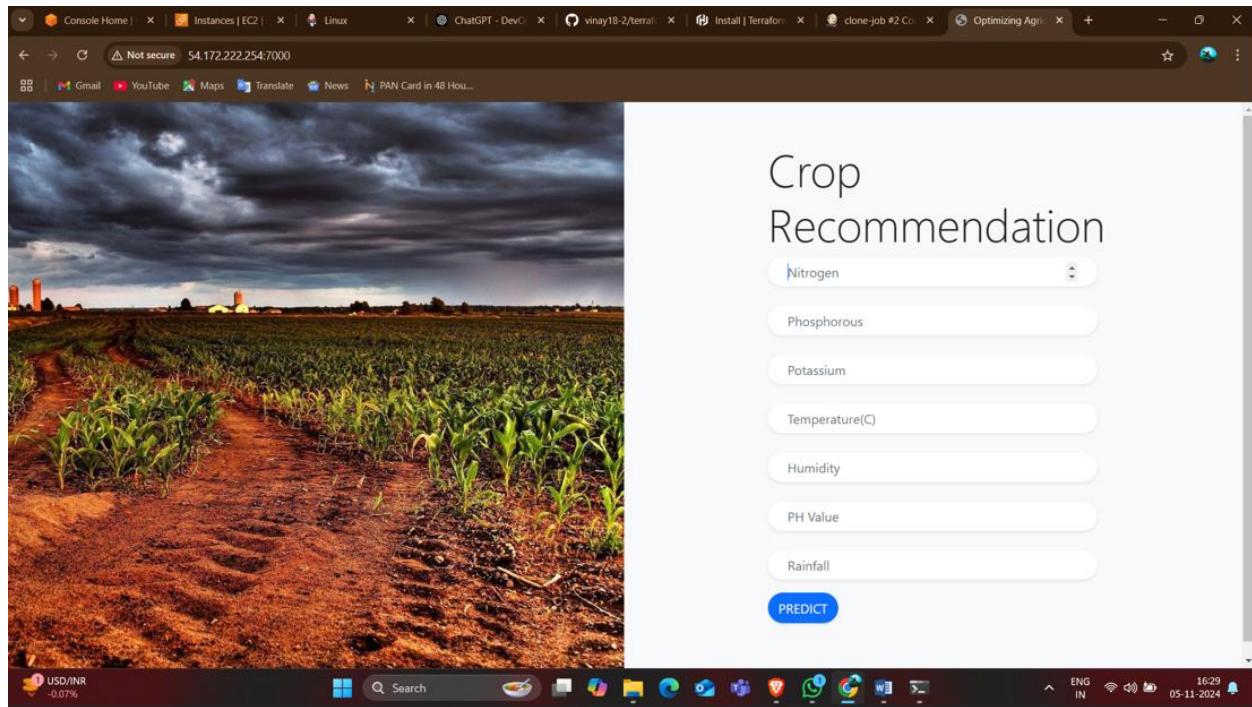
```
#!/bin/bash
# Install yum-utils and shadow-utils for package management
sudo yum install -y yum-utils shadow-utils
# Add Hashicorp's official repository
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
# Install Terraform
sudo yum -y install terraform
# Navigate to the Terraform configuration directory
cd /var/lib/jenkins/workspace/your-terraform-project-directory # Adjust to your Jenkins workspace directory
# Initialize Terraform
terraform init
# Format and validate Terraform files
terraform fmt -check
terraform validate
# Create a Terraform plan (saved as 'tfplan')
terraform plan -out=tfplan
# Apply the plan
terraform apply -auto-approve tfplan
```

- ❖ Build the job and observe the job is success or not.

The screenshot shows the Jenkins Dashboard. On the left sidebar, there are links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table with columns: S (Status), W (Work), Name (clone-job), Last Success (6 min 8 sec #2), Last Failure (12 min #1), and Last Duration (1 min 15 sec). Below the table, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). At the bottom right, it says 'REST API' and 'Jenkins 2.479.1'.

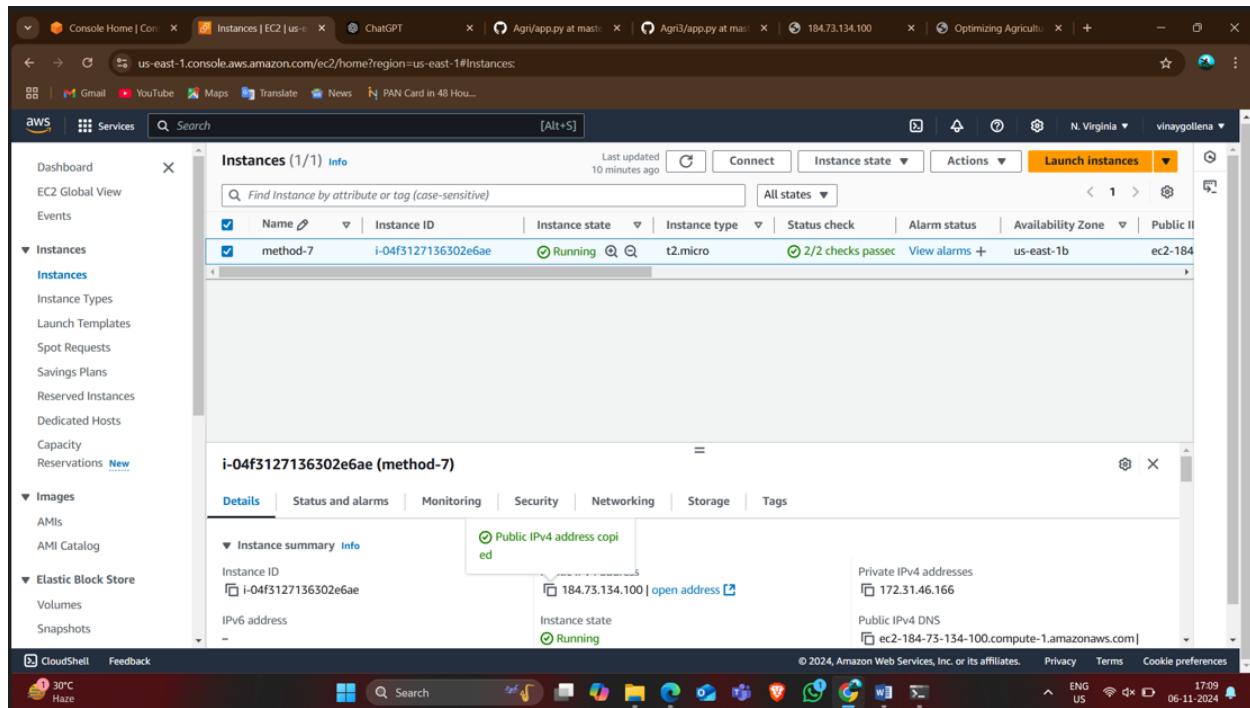
The screenshot shows the Jenkins job details page for 'clone-job'. The left sidebar has links for 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. The main content area shows the job name 'clone-job' with a green checkmark icon. Below it is a 'Permalinks' section with a bulleted list of build logs. The 'Builds' section shows a table with columns: Status, Build Number, and Last Result. It lists two builds: '#2 10:54 AM' (Success) and '#1 10:48 AM' (Failure). At the bottom right, it says 'REST API' and 'Jenkins 2.479.1'.

- ❖ Than browse the created instance ip adres in the google and observe the output.



# 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

- ❖ Launch an instance.



- ❖ Connect to the terminal and install docker.

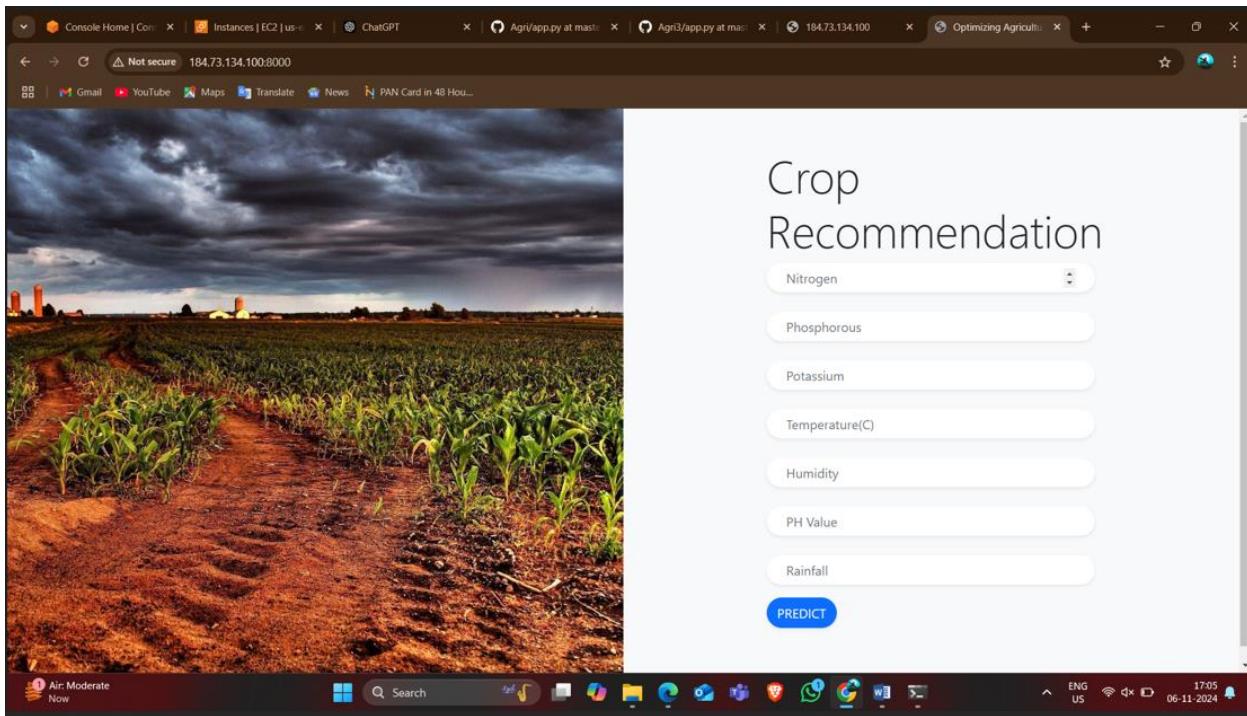
```
ec2-user@ip-172-31-46-166: ~ + ~
Amazon Linux 2023, GA and supported until 2028-03-15.
https://aws.amazon.com/linux/amazon-linux-2023/
[ec2-user@ip-172-31-46-166 ~]$ sudo yum install docker -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
Package docker-25.0.6-1.amzn2.0.2.x86_64 already installed and latest version
Nothing to do
[ec2-user@ip-172-31-46-166 ~]$ sudo systemctl start docker
sudo: systemctl: command not found
[ec2-user@ip-172-31-46-166 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-46-166 ~]$ sudo systemctl start docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-46-166 ~]$ sudo systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: disabled)
     Active: active (running) since Wed 2024-11-06 09:03:59 UTC; 30s ago
       Docs: https://docs.docker.com
 Main PID: 9737 (/usr/bin/dockerd)
   CGroup: /system.slice/docker.service
           └─ 9737 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock --default-ulimit nofile=32768:65536

Nov 06 09:03:58 ip-172-31-46-166.ec2.internal systemd[1]: Starting Docker Application Container Engine...
Nov 06 09:03:58 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:58.725494382Z" level=info msg="Starting up"
Nov 06 09:03:58 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:58.773184196Z" level=info msg="Loading containers: start."
Nov 06 09:03:58 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:58.773184196Z" level=info msg="Loading containers: end."
Nov 06 09:03:59 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:59.030548122Z" level=warning msg="WARNING: bridge-nf-call-iptc...sabled"
Nov 06 09:03:59 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:59.031117209Z" level=warning msg="WARNING: bridge-nf-call-ip6tc...sabled"
Nov 06 09:03:59 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:59.031592209Z" level=info msg="Docker daemon command line: c...-25.0.6"
Nov 06 09:03:59 ip-172-31-46-166.ec2.internal dockerd[9737]: time="2024-11-06T09:03:59.06924935H" level=info msg="API listen on /run/docker.sock"
Nov 06 09:03:59 ip-172-31-46-166.ec2.internal systemd[1]: Started Docker Application Container Engine.
Warning: No metrics were collected since the system was full.
[ec2-user@ip-172-31-46-166 ~]$ sudo chmod 666 /var/run/docker.sock
chmod: cannot access '/var/run/docker.sock': No such file or directory
[ec2-user@ip-172-31-46-166 ~]$ sudo usermod -aG docker ec2-user
usermod: usermod not found
[ec2-user@ip-172-31-46-166 ~]$ sudo usermod aG username
Usage: usermod [options] LOGIN
```

- ❖ Start and eabbel the docker and than provide the permissions to the docker-sock.
  - ❖ And write the docker name it as Dokcer.yaml  
“docker build –t image”

- ❖ Than build image and check the images than crate and run the container with the port assign method.  
“ docker run -dt -name conatianername -p 8000:7000 imagename

- ❖ Than copy the ip of instance and browse it.



- ❖ Than login to the docker and push that image to the docker hub.

```
ec2-user@ip-172-31-46-166:~$ 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: vinay182206
Password:
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

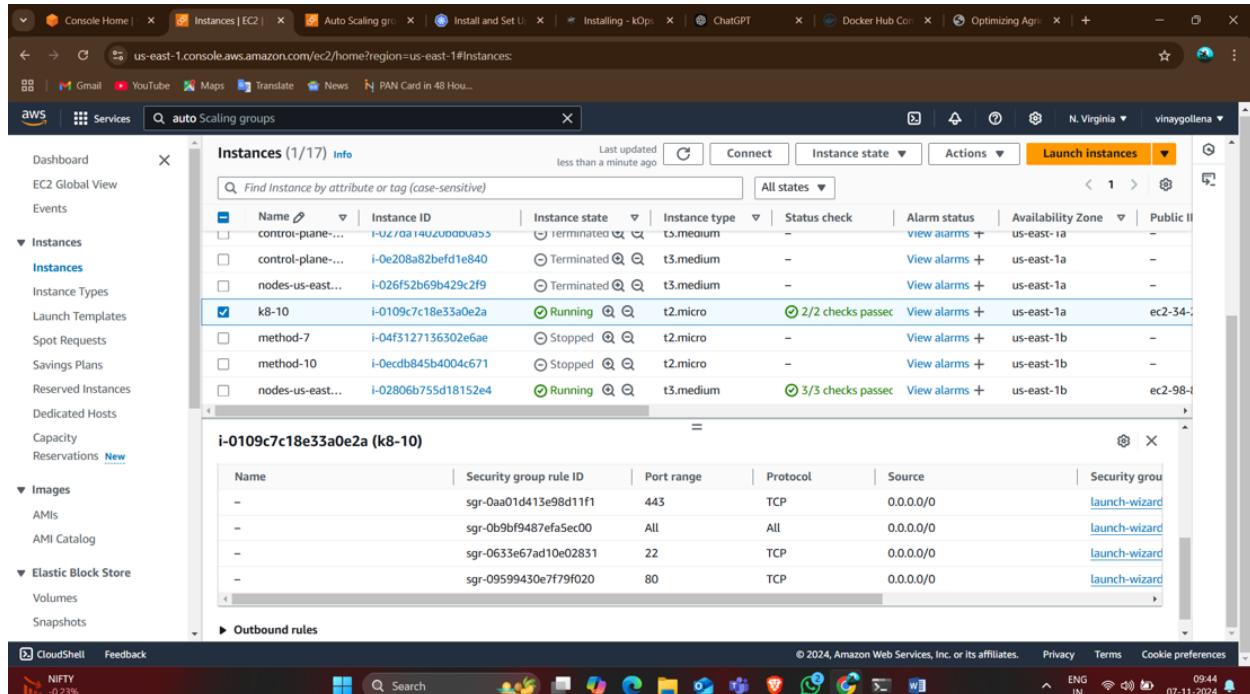
[ec2-user@ip-172-31-46-166:~]$ docker tag image vinay182206\Agri3:latest
Error parsing reference: "vinay182206Agri3:latest" is not a valid repository/tag: invalid reference format: repository name (library/vinay182206Agri3) must be lowercase
[ec2-user@ip-172-31-46-166:~]$ docker tag image vinay182206\agri3:latest
[ec2-user@ip-172-31-46-166:~]$ docker tag image vinay182206\agri3:latest
[ec2-user@ip-172-31-46-166:~]$ docker push vinay182206/agri3:latest:client_loop: send disconnect: Connection reset

C:\Users\user\Downloads>ssh -i "vinay.pem" ec2-user@ec2-184-73-134-100.compute-1.amazonaws.com
Last login: Wed Nov  6 11:40:24 2024 from 103.55.213.30
_
  _###_ Amazon Linux 2
  _\###\ AL2 End of Life is 2025-06-30.
   \##|
    \#/ -->
     / A newer version of Amazon Linux is available!
      / Amazon Linux 2023, GA and supported until 2028-03-15.
     /_m/ https://aws.amazon.com/linux/amazon-linux-2023/

[ec2-user@ip-172-31-46-166:~]$ docker push vinay182206\agri3:latest
The push refers to repository [docke.io/vinay182206/agri3]
e6b3b92d8110: Pushed
5f70bf18a086: Pushed
569af805faa8: Pushed
39cb5cddda92: Pushed
fcac1b22fe3d: Mounted from library/amazonlinux
latest: digest: sha256:131fee805e8bc77381a35f06d4e5d0c07460c7fd4feb29a4375daf8140cddc73 size: 1373
[ec2-user@ip-172-31-46-166:~]$
```

# METHOD-8 : Build and deploy python applications with Docker and k8's (EKS and KOPS) (use Declarative manifest method along with docker image)

- ❖ Launch the instance.



- ❖ Connect to the terminal and update it

```
ec2-user@ip-172-31-16-226:~ + ~
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\user>cd downloads

C:\Users\user\Downloads>ssh -i "vinayam.pem" ec2-user@ec2-34-229-175-131.compute-1.amazonaws.com
The authenticity of host 'ec2-34-229-175-131.compute-1.amazonaws.com (34.229.175.131)' can't be established.
ED25519 key fingerprint is SHA256:G6N9VzFhqIf6EiiulsGRH8m72dTBEERQDZSqlLzujc.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-34-229-175-131.compute-1.amazonaws.com' (ED25519) to the list of known hosts.

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-16-226 ~]$ sudo yum update -y
Last metadata expiration check: 0:00:22 ago on Thu Nov  7 03:47:11 2024.
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-172-31-16-226 ~]$ curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64
chmod +x kops
sudo mv kops /usr/local/bin/kops
Total  % Received % Xferd  Average Speed   Time   Time   Time  Current
          Dload  Upload Total Spent   Left  Speed
 0    0    0    0    0    0    0:--:-- --:--:-- 0
100 238M 100 238M  0    0    66.3M  0:00:03 0:00:03 70.1M
[ec2-user@ip-172-31-16-226 ~]$ kops
Kops is Kubernetes Operations.

Kops is the easiest way to get a production grade Kubernetes cluster up and running. We like to think of it as kubectl for clusters.

Kops helps you create, destroy, upgrade and maintain production-grade, highly available, Kubernetes clusters from the command line. AWS (Amazon Web Service) is currently officially supported, with Digital Ocean and OpenStack in beta support.

NIFTY -0.23%
```

- ❖ Install the kubernetes and kubectland kops.

```

ec2-user@ip-172-31-16-226:~ + ~
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
[ec2-user@ip-172-31-16-226 ~]$ sudo yum install kubectl
Kubernetes
Dependencies resolved.
=====
| Package           | Architecture | Version   | Repository | Size |
|=====|=====|=====|=====|=====|
| Installing:     |             |           |           |       |
| kubectl          | x86_64      | 1.31.2-150500.1.1 | kubernetes | 11 M |
|=====|=====|=====|=====|=====|
Transaction Summary
=====
| Install 1 Package |
Total download size: 11 M
Installed size: 54 M
Is this ok [y/N]: y
Downloading Packages:
kubectl-1.31.2-150500.1.1.x86_64.rpm
=====
| Total          | 45 MB/s | 11 MB | 00:00 |
| Kubernetes    | 45 MB/s | 11 MB | 00:00 |
Importing GPG key 0x9A296436:
Userid : "isv:kubernetes OBS Project <isv:kubernetes@build.opensuse.org>"
Fingerprint: DE15 B144 86CD 377B 9E87 6E1A 2346 54DA 9A29 6436
From   : https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repodata/repomd.xml.key
Is this ok [y/N]: y
Key imported successfully
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : kubectl-1.31.2-150500.1.1.x86_64 1/1
Verifying  : kubectl-1.31.2-150500.1.1.x86_64 1/1
Installed:
  kubectl-1.31.2-150500.1.1.x86_64

```

- ❖ Than create a s3 bucket by using aws configure and with access key and secret key or else we have another option that create an I am user create an instance than provide admin access to that user.
- ❖ After that generate key “ssh-keygen”

```

ec2-user@ip-172-31-16-226:~ + ~
Usage:
  kubectl [flags] [options]
  Use "kubectl <command> --help" for more information about a given command.
  Use "kubectl options" for a list of global command-line options (applies to all commands).
[ec2-user@ip-172-31-16-226 ~]$ aws configure
AWS Access Key ID [None]: AKIA4MTNNV3F6FP0Z07
AWS Secret Access Key [None]: 5VhfFeiY8Mp12CARg1oHzGcgK5GAQhMQCVldVtfh
Default region name [None]:
Default output format [None]:
[ec2-user@ip-172-31-16-226 ~]$ aws s3 mb s3://virat-18
make_bucket: virat-18
[ec2-user@ip-172-31-16-226 ~]$ aws s3 ls
2024-06-16 13:37:03 cf-templates-4opwclb4wydv-us-east-1
2024-06-16 16:43:48 elasticbeanstalk-us-east-1-851725625974
2024-09-04 09:37:45 vinay-k8
2024-10-22 09:28:06 vinay-k8
2024-11-06 19:25:09 vinay-k9
2024-11-06 20:11:53 vinnu-k8
2024-11-07 03:57:57 virat-18
[ec2-user@ip-172-31-16-226 ~]$ export KOPS_STATE_STORE=s3://virat-18
[ec2-user@ip-172-31-16-226 ~]$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ec2-user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ec2-user/.ssh/id_rsa
Your public key has been saved in /home/ec2-user/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:MMTx9Qhj7aZZkGs8iI0YeyRu6vQMc6fim8m+jxepxw ec2-user@ip-172-31-16-226.ec2.internal
The key's randomart image is:
+---[RSA 3072]---+
| . .
| ...
| . . . o .
| * . o o .
| o + S o o
| . = o * .
| o o Eo. + =
| =+o+o+o . +
|
```

❖ Than create cluster with the using s3 bucket.

```
| . . = 0 * . |
| o .Eo. + =
|+=+o++o . +
|+@=+o + oo. +
+---[SHA256]---+
[ec2-user@ip-172-31-16-226 ~]$ kops create cluster --name .k8s.local --state s3://virat-18 --zones useast-1b,us-east-1a --node-count 2 --yes
W1107 04:00:11.424931 2861 new_cluster.go:1435] Gossip is deprecated, using None DNS instead
I1107 04:00:11.425092 2861 new_cluster.go:1454] Cloud Provider ID: "aws"
Error: error assigning default machine type for control plane: error finding default machine type: error checking if instance type "t3.medium" is supported
in regions: error reading operation error from DescribeInstanceOfferings, https response error StatusCode: 0, RequestID: , request send failed, Post "https://ec2.us-east-1.amazonaws.com/dial tcp lookup ec2.us-east-1.amazonaws.com on 172.31.0.2:53: no such host
[ec2-user@ip-172-31-16-226 ~]$ kops create cluster --name viratcluster.local --state s3://virat-18 --zones us-east-1b,us-east-1a --node-count 2 --yes
I1107 04:00:11.425154 2865 new_cluster.go:1454] Cloud Provider ID: "aws"
I1107 04:00:44.822296 2865 subnets.go:224] Assigned CIDR 172.20.0.0/17 to subnet us-east-1a
I1107 04:00:44.822334 2865 subnets.go:224] Assigned CIDR 172.20.128.0/17 to subnet us-east-1b
I1107 04:00:46.719285 2865 builder.go:312] asset "https://dl.k8s.io/release/v1.30.2/bin/linux/amd64/kubelet" is not well-known, downloading hash
I1107 04:00:46.814638 2865 builder.go:312] asset "https://dl.k8s.io/release/v1.30.2/bin/linux/amd64/kubectl" is not well-known, downloading hash
I1107 04:00:46.958392 2865 builder.go:312] asset "https://github.com/containerd/containerd/releases/download/v1.7.16/containerd-1.7.16-linux-amd64.tar.gz"
" is not well-known, downloading hash
I1107 04:00:47.256932 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/amd64/nodeup" is not well-known, downloading hash
I1107 04:00:47.386793 2865 builder.go:312] asset "https://dl.k8s.io/release/v1.30.2/bin/linux/arm64/kubelet" is not well-known, downloading hash
I1107 04:00:47.453510 2865 builder.go:312] asset "https://github.com/containerd/containerd/releases/download/v1.7.16-linux-arm64.tar.gz"
" is not well-known, downloading hash
I1107 04:00:47.599365 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/arm64/nodeup" is not well-known, downloading hash
I1107 04:00:47.898195 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/amd64/protokube" is not well-known, downloading hash
I1107 04:00:48.169813 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/arm64/protokube" is not well-known, downloading has
h
I1107 04:00:48.276128 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/arm64/protokube" is not well-known, downloading has
h
I1107 04:00:48.393289 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/amd64/channels" is not well-known, downloading hash
I1107 04:00:48.486194 2865 builder.go:312] asset "https://artifacts.k8s.io/binaries/kops/1.30.1/linux/arm64/channels" is not well-known, downloading hash
I1107 04:00:48.544526 2865 executor.go:113] Tasks: 0 done / 126 total; 43 can run
W1107 04:00:56.518431 2865 vfs_keystorereader.go:143] CA private key was not found
I1107 04:00:56.536699 2865 keypair.go:226] Issuing new certificate: "etcd-peers-ca-main"
I1107 04:00:56.548611 2865 keypair.go:226] Issuing new certificate: "etcd-peers-ca-events"
I1107 04:00:56.568165 2865 keypair.go:226] Issuing new certificate: "etcd-clients-ca"
I1107 04:00:56.580269 2865 keypair.go:226] Issuing new certificate: "etcd-manager-ca-events"
I1107 04:00:56.605322 2865 keypair.go:226] Issuing new certificate: "apiserver-aggregator-ca"
I1107 04:00:56.620921 2865 keypair.go:226] Issuing new certificate: "apiserver-peers-ca-main"
I1107 04:00:56.768476 2865 keypair.go:226] Issuing new certificate: "service-account"
I1107 04:00:56.788533 2865 vfs_keystorereader.go:143] CA private key was not found
W1107 04:00:56.788533 2865 vfs_keystorereader.go:143] CA private key was not found
```

❖ Than write the deployment file and service file.

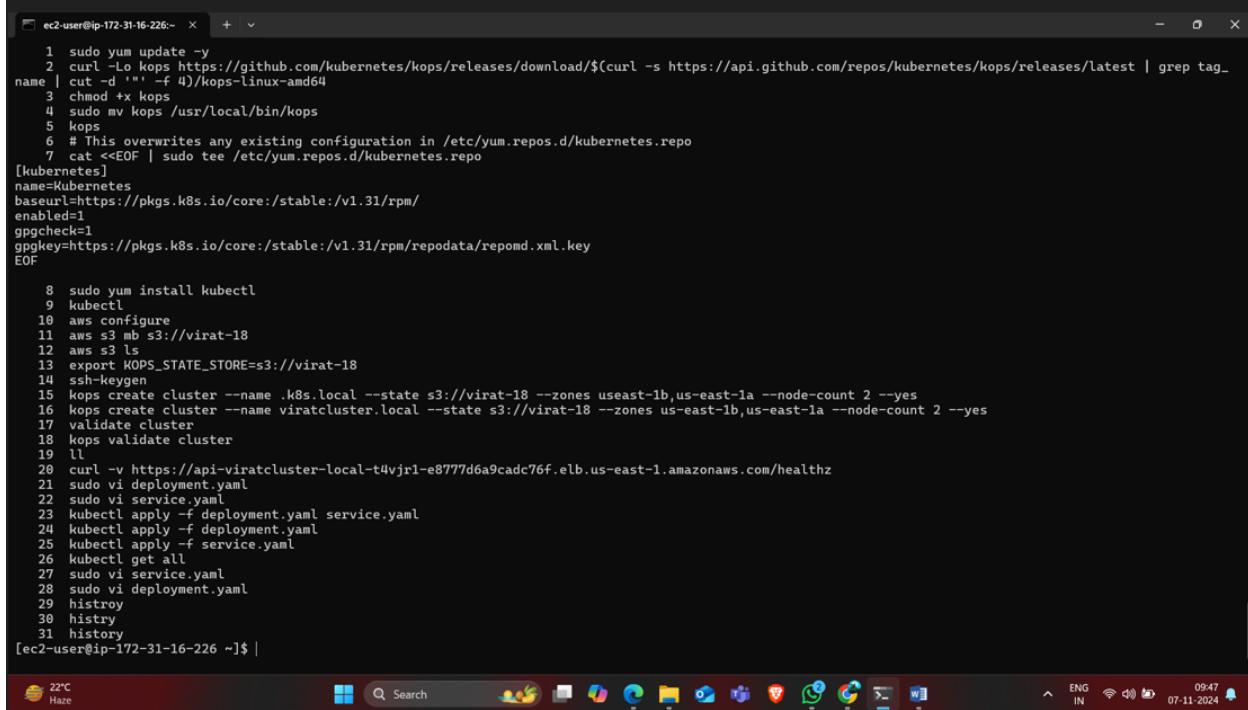
```
# deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: python-app
spec:
  replicas: 3
  selector:
    matchLabels:
      app: python-app
  template:
    metadata:
      labels:
        app: python-app
    spec:
      containers:
        - name: agri3
          image: vinay182206/agri3
          ports:
            - containerPort: 7000
```

"deployment.yaml" 21L, 346B

```
# service.yaml
apiVersion: v1
kind: Service
metadata:
  name: python-app-service
spec:
  selector:
    app: python-app
  ports:
    protocol: TCP
    port: 7000
    targetPort: 7000
  type: LoadBalancer
```

"service.yaml" 14L, 210B

❖ Here the used command history.



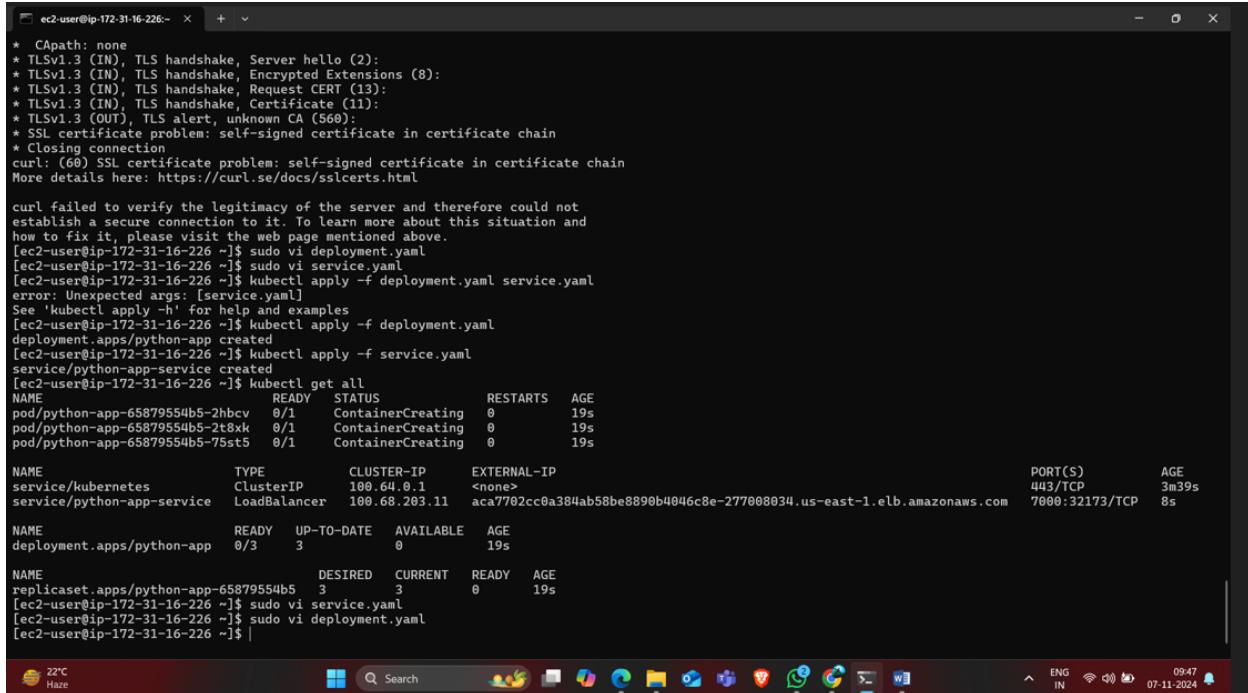
```

1 sudo yum update -y
2 curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_ name | cut -d '"' -f 4)/kops-linux-amd64
3 chmod +x kops
4 sudo mv kops /usr/local/bin/kops
5 kops
6 # This overwrites any existing configuration in /etc/yum.repos.d/kubernetes.repo
7 cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/
enabled=1
gpgcheck=1
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.31/rpm/repo/repodata/repomd.xml.key
EOF

8 sudo yum install kubectl
9 kubectl
10 aws configure
11 aws s3 mb s3://virat-18
12 aws s3 ls
13 export KOPS_STATE_STORE=s3://virat-18
14 ssh-keygen
15 kops create cluster --name .k8s.local --state s3://virat-18 --zones us-east-1b,us-east-1a --node-count 2 --yes
16 kops create cluster --name viratcluster.local --state s3://virat-18 --zones us-east-1b,us-east-1a --node-count 2 --yes
17 validate cluster
18 kops validate cluster
19 ll
20 curl -v https://api-viratcluster-local-t4vr1-e8777d6a9cadc76f.elb.us-east-1.amazonaws.com/healthz
21 sudo vi deployment.yaml
22 sudo vi service.yaml
23 kubectl apply -f deployment.yaml service.yaml
24 kubectl apply -f deployment.yaml
25 kubectl apply -f service.yaml
26 kubectl get all
27 sudo vi service.yaml
28 sudo vi deployment.yaml
29 histroy
30 histroy
31 histroy
[ec2-user@ip-172-31-16-226 ~]$ |

```

❖ After the applying the deployment and service file with kubectl than we get this output and copy the load balancer url form terminal or load balancer created in the aws consol.



```

* CPath: none
* TLSv1.3 (IN), TLS handshake, Server hello (2):
* TLSv1.3 (IN), TLS handshake, Encrypted Extensions (8):
* TLSv1.3 (IN), TLS handshake, Request CERT (13):
* TLSv1.3 (IN), TLS handshake, Certificate (11):
* TLSv1.3 (OUT), TLS alert, unknown CA (560):
* SSL certificate problem: self-signed certificate in certificate chain
* Closing connection
curl: (60) SSL certificate problem: self-signed certificate in certificate chain
More details here: https://curl.se/docs/sslcerts.html

curl failed to verify the legitimacy of the server and therefore could not
establish a secure connection to it. To learn more about this situation and
how to fix it, please visit the web page mentioned above.
[ec2-user@ip-172-31-16-226 ~]$ sudo vi deployment.yaml
[ec2-user@ip-172-31-16-226 ~]$ sudo vi service.yaml
[ec2-user@ip-172-31-16-226 ~]$ kubectl apply -f deployment.yaml service.yaml
error: Unexpected args: [service.yaml]
See 'kubectl apply -h' for help and examples
[ec2-user@ip-172-31-16-226 ~]$ kubectl apply -f deployment.yaml
deployment.apps/python-app created
[ec2-user@ip-172-31-16-226 ~]$ kubectl apply -f service.yaml
service/python-app-service created
[ec2-user@ip-172-31-16-226 ~]$ kubectl get all
NAME READY STATUS RESTARTS AGE
pod/python-app-65879554b5-2hbcv 0/1 ContainerCreating 0 19s
pod/python-app-65879554b5-2t8xk 0/1 ContainerCreating 0 19s
pod/python-app-65879554b5-75st5 0/1 ContainerCreating 0 19s

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
service/kubernetes ClusterIP 100.64.0.1 <none> 443/TCP 3m39s
service/python-app-service LoadBalancer 100.68.203.11 aca7702cc0a384ab58be8890b4046c8e-277008034.us-east-1.elb.amazonaws.com 7000:32173/TCP 8s

NAME READY UP-TO-DATE AVAILABLE AGE
deployment.apps/python-app 0/3 3 0 19s

NAME DESIRED CURRENT READY AGE
replicaset.apps/python-app-65879554b5 3 3 0 19s
[ec2-user@ip-172-31-16-226 ~]$ sudo vi service.yaml
[ec2-user@ip-172-31-16-226 ~]$ sudo vi deployment.yaml
[ec2-user@ip-172-31-16-226 ~]$ |

```

- ❖ Copy the created loadbalancer end point address and search in the browser with the given port.

The screenshot shows the AWS Management Console with the URL [us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers](https://us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers). The sidebar on the left is expanded, showing sections like Elastic Block Store, Network & Security, Load Balancing (selected), Auto Scaling, and CloudShell. The main content area is titled "Load balancers (2)". It displays two entries:

Name	DNS name	State	VPC ID	Availability Zones	Type
aca7702cc0a384ab58...	aca7702cc0a384ab58be8...	-	vpc-09166cce1093170...	2 Availability Zones	classic
api-viratcluster-local-t...	api-viratcluster-local-t4vjr...	Active	vpc-09166cce1093170...	2 Availability Zones	network

Below the table, a message says "0 load balancers selected".

- ❖ This is the hosted application .

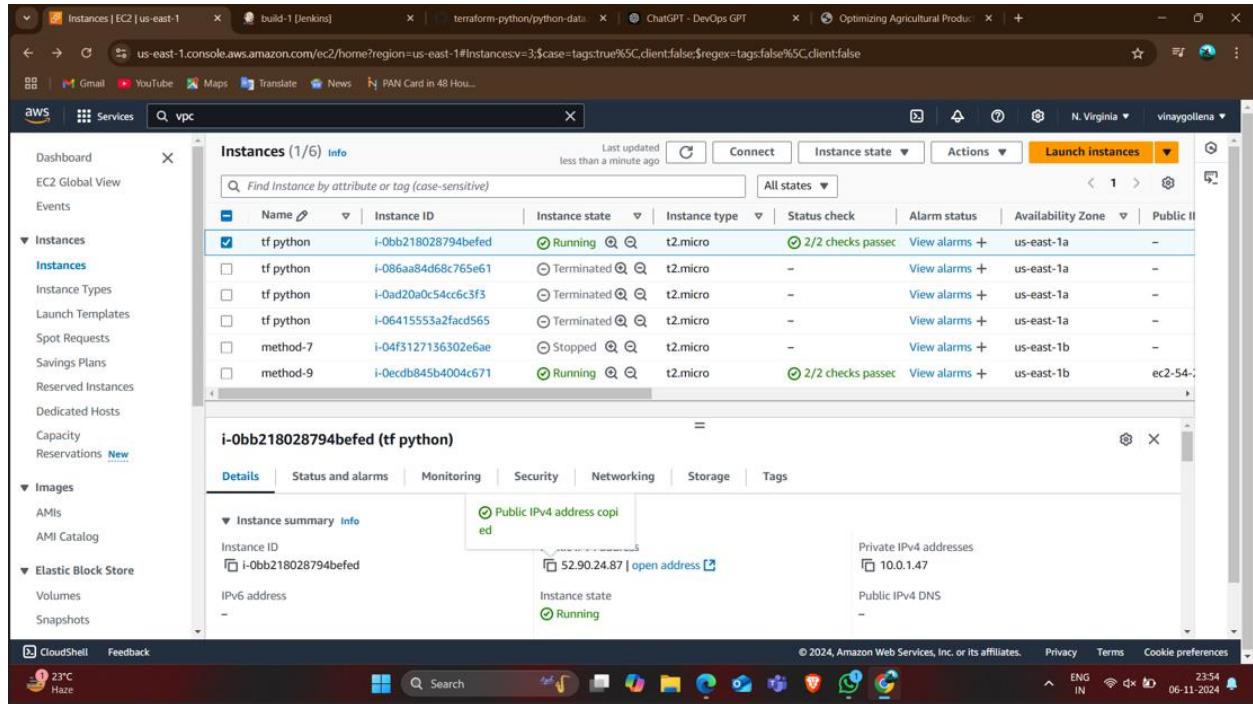
The screenshot shows a web browser window with the URL [aca7702cc0a384ab58be8890b4046c8e-277008034.us-east-1.elb.amazonaws.com:7000](http://aca7702cc0a384ab58be8890b4046c8e-277008034.us-east-1.elb.amazonaws.com:7000). The page title is "Not secure". The main content is a photograph of a field of young corn plants under a dramatic, cloudy sky. To the right of the image, there is a form titled "Crop Recommendation" with the following input fields:

- Nitrogen
- Phosphorous
- Potassium
- Temperature(C)
- Humidity
- PH Value
- Rainfall

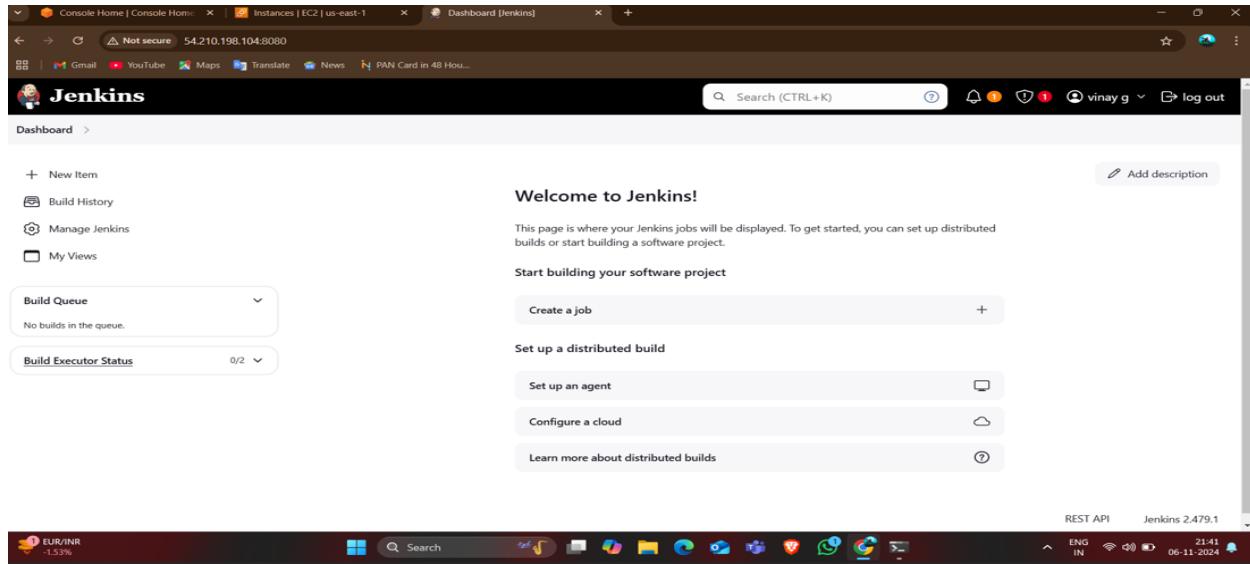
Below these fields is a blue "PREDICT" button.

# METHOD-9 : Build and deploy python applications with the Git, github, Jenkins and Terraform (with Build periodically, pollscm and webhooks)

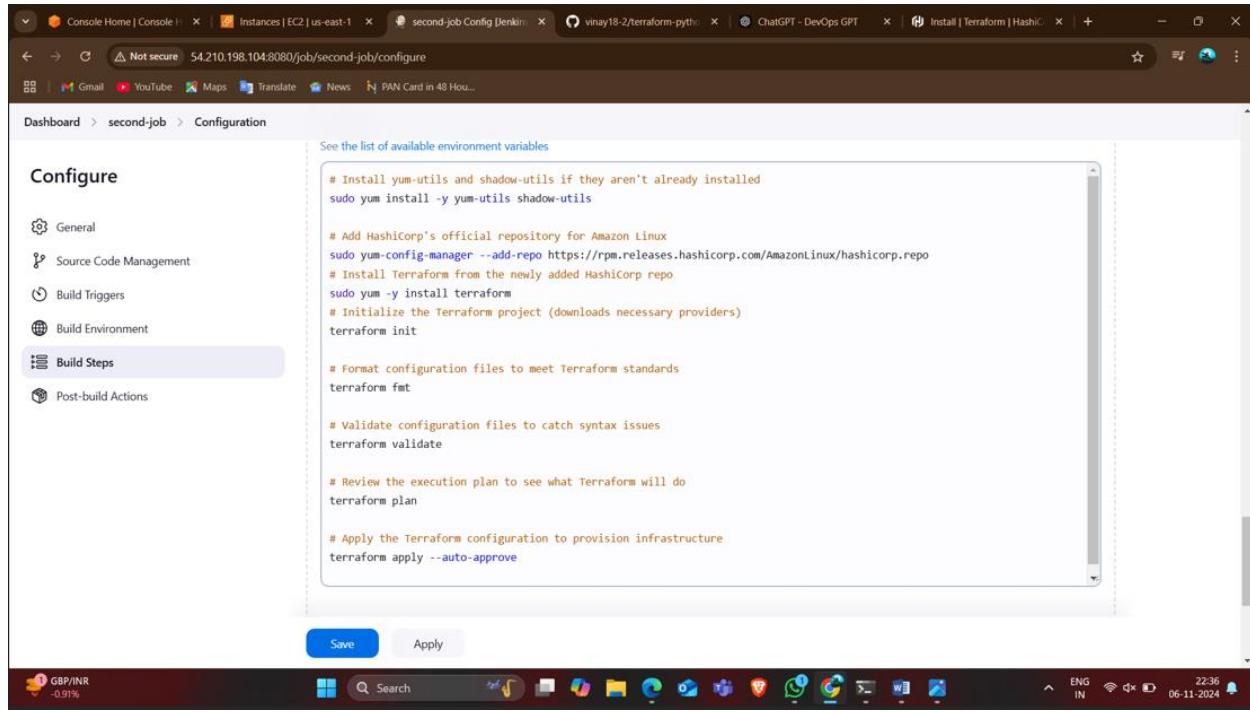
- ❖ Launch the instance and connect to the terminal.



- ❖ After connecting to the terminal than update it and install Jenkins.
- ❖ Than start and enable the Jenkins.
- ❖ Than login to the Jenkins page.
- ❖ Than create a clone job for cloning the git repo
- ❖ And save it build it.
- ❖ Than create the another job to build and add aws credentials to it by the help of installing aws credentials.
- ❖ And write the commands to install and apply the terraform for the resources.



## ❖ Terraform installation and execute commands.



- ❖ After building the we can get like this.

The screenshot shows the Jenkins Dashboard. On the left, there's a sidebar with links for 'New Item', 'Build History', 'Project Relationship', 'Check File Fingerprint', 'Manage Jenkins', and 'My Views'. The main area displays a table of builds:

S	W	Name	Last Success	Last Failure	Last Duration
		build-1	5 min 31 sec #1	N/A	1 min 22 sec
		clone-1	11 min #2	N/A	0.13 sec

Below the table, there are sections for 'Build Queue' (No builds in the queue) and 'Build Executor Status' (0/2). At the bottom right, there are links for 'REST API' and 'Jenkins 2.479.1'.

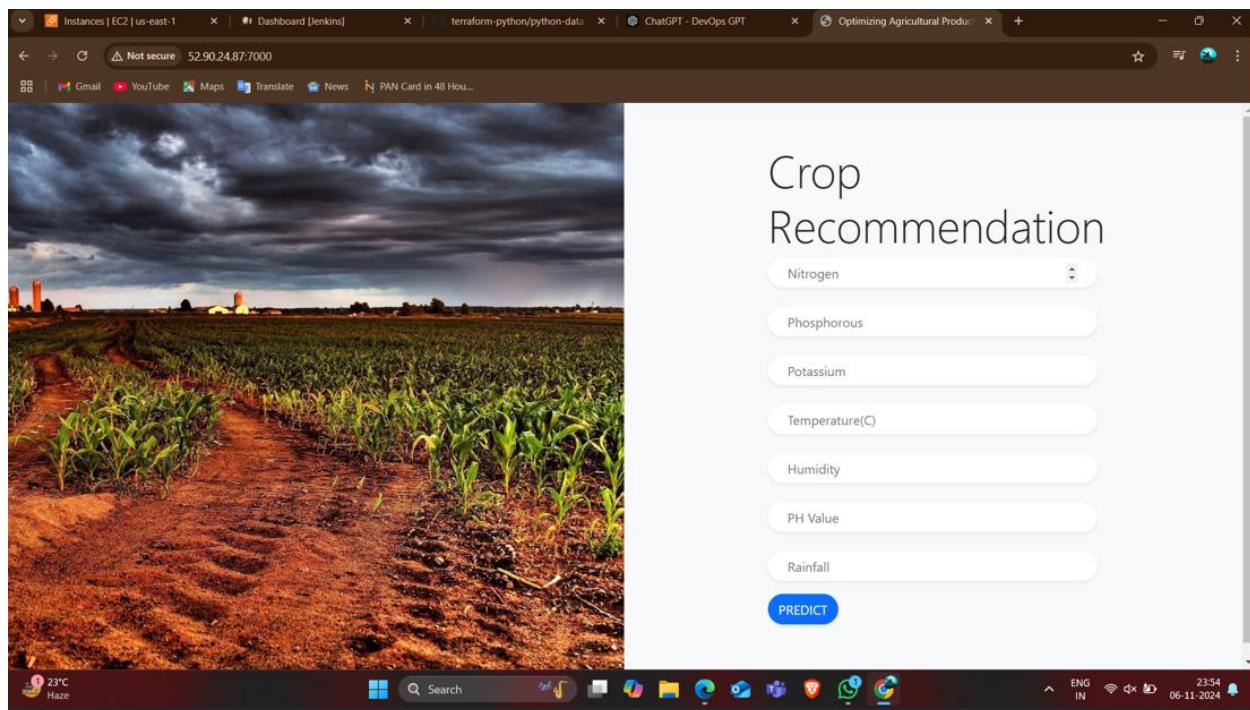
- ❖ Than create webhooks for that repo in the github.

The screenshot shows the GitHub repository settings for 'vinay18-2/terraform-python'. The left sidebar includes 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'. The 'Settings' tab is selected. Under 'Webhooks', it says 'Okay, that hook was successfully created. We sent a ping payload to test it out! Read more about it at <https://docs.github.com/webhooks/ping-event>'. A table lists a single webhook entry:

Action	URL	Edit	Delete
push	<a href="http://3.85.139.132:8080/github-webhook/">http://3.85.139.132:8080/github-webhook/</a>	Edit	Delete

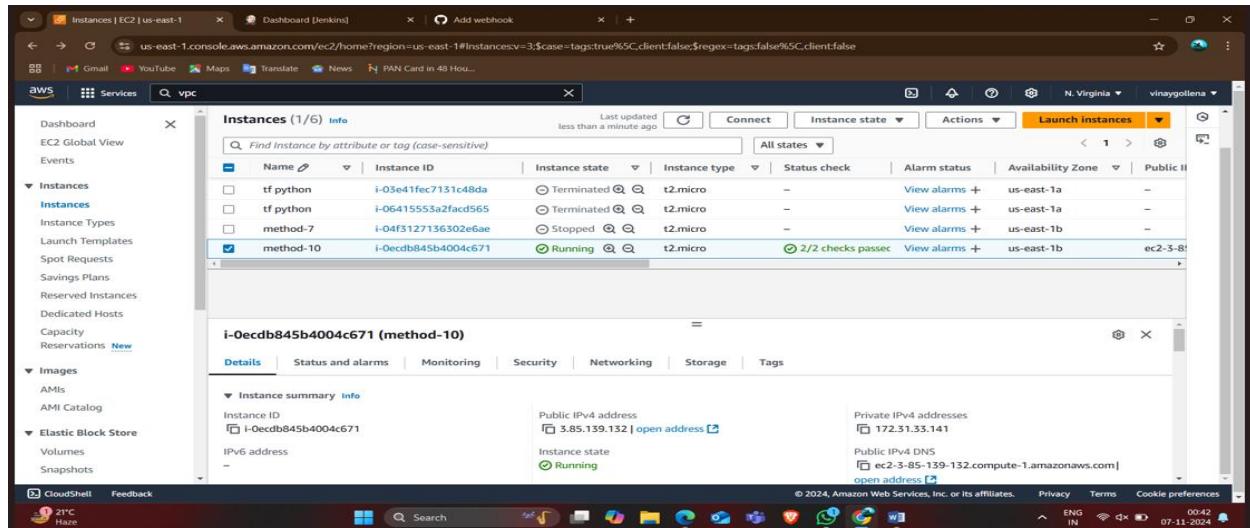
At the bottom, there's a note: 'Last delivery was not successful. Invalid HTTP Response: 302.'

- ❖ Than provide the build periodically,pollscm and webhooks.
- ❖ Browse the id address and observe .
- ❖ Here the buildperiodically,pollscm and webhooks helps to get the results when the developer changes the code.
- ❖ Here the webhooks will gets the developer changes and get the spontaneous output within less time.

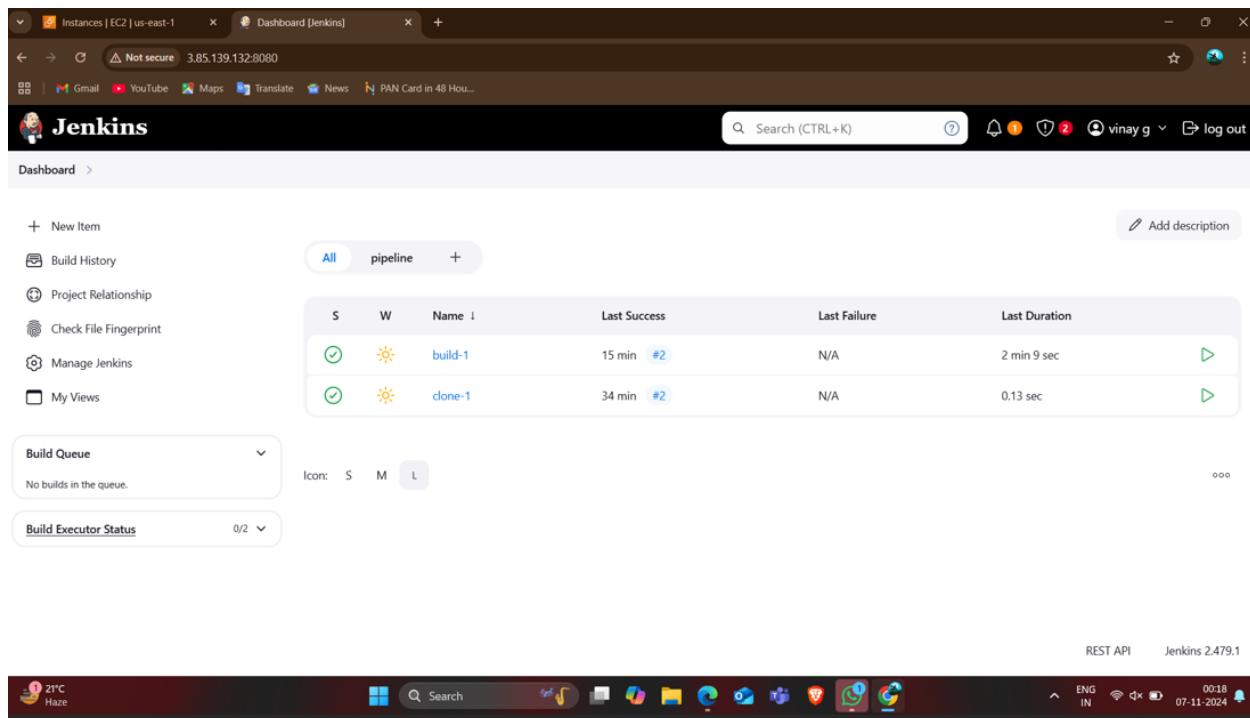


# Method-10 : Build and deploy python applications with pipeline method (create clone job and Build job) with Build periodically, pollscm and webhooks

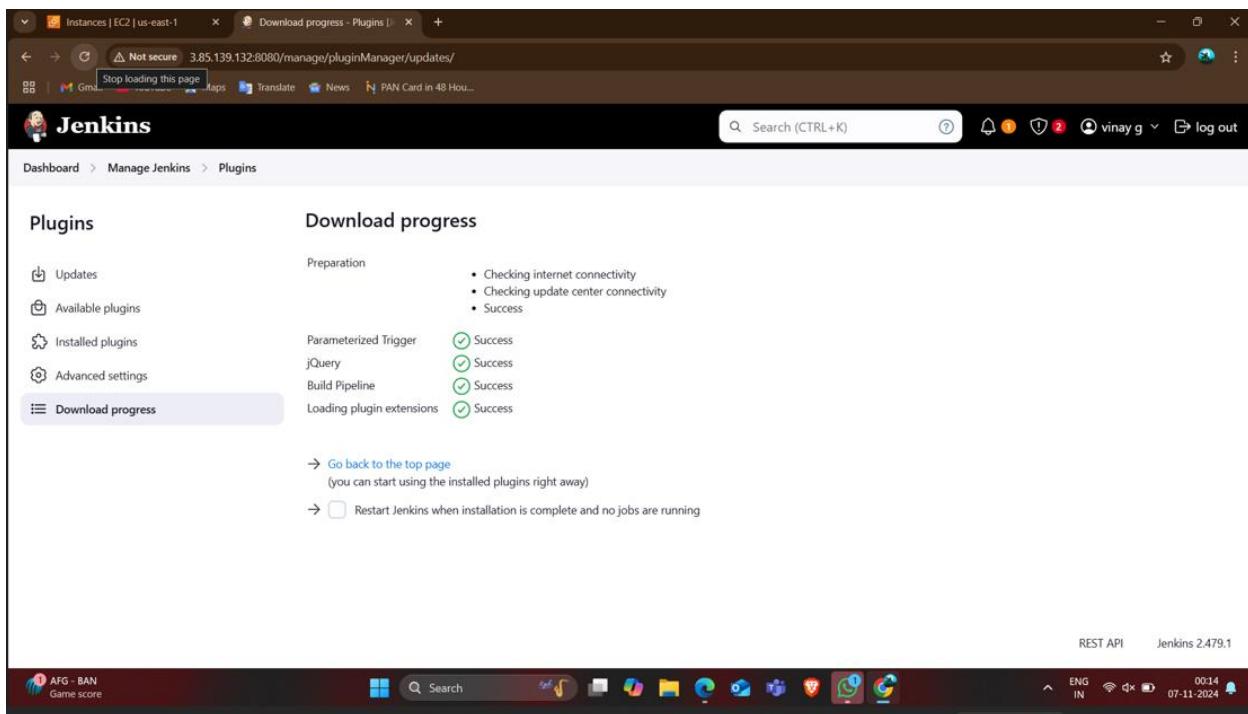
- ❖ Launch an instance.



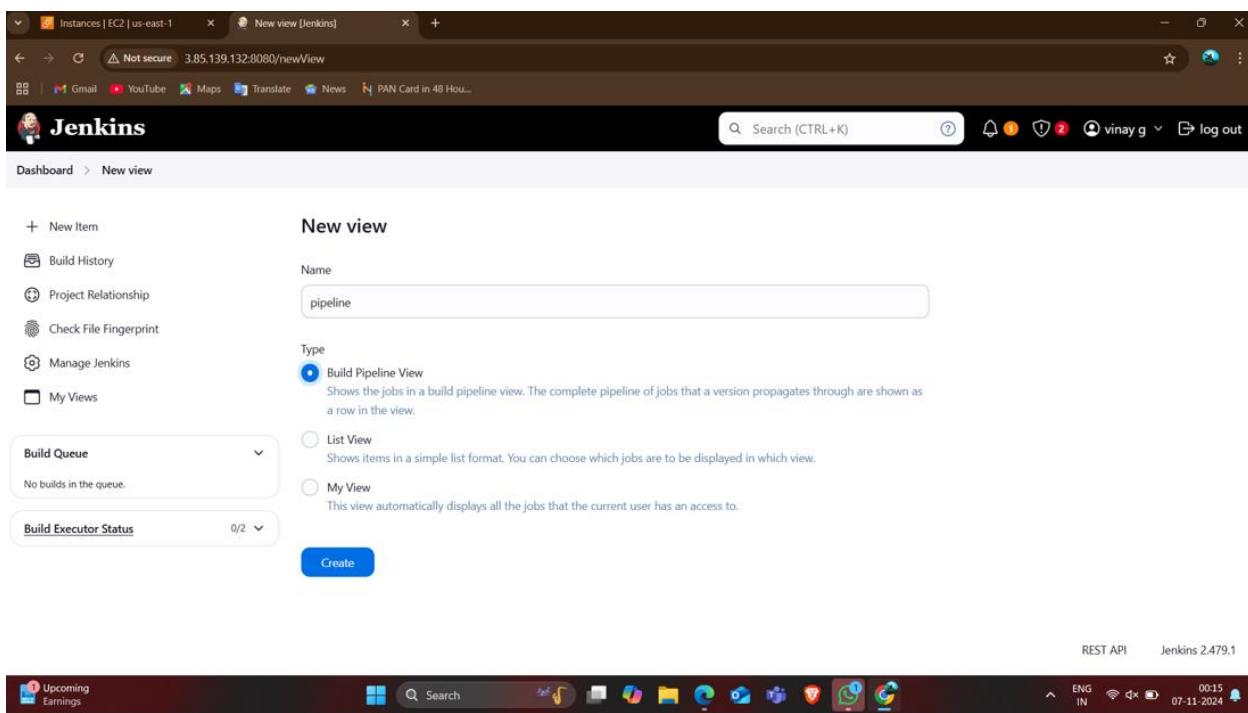
- ❖ Do the same process like thy method-9.



❖ Install the build pipeline plugin.



❖ Than create the pipeline.



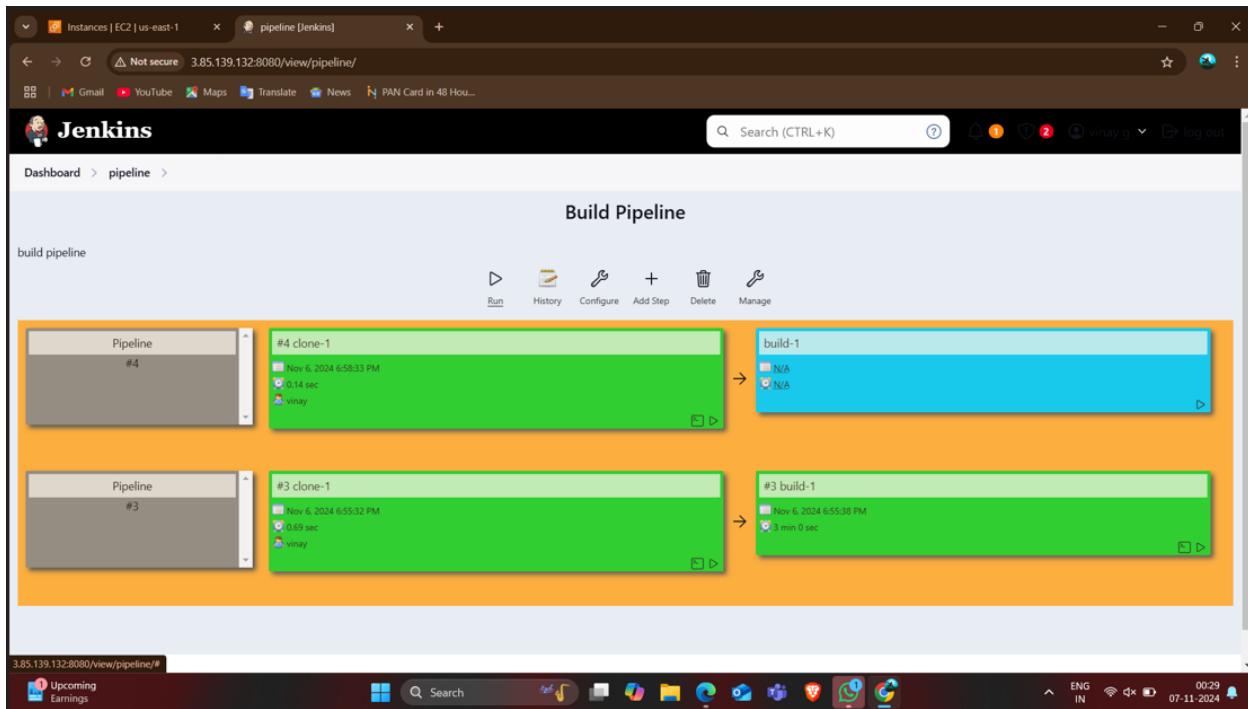
- ❖ After creating the pipeline this is the output of pipeline.

The screenshot shows the Jenkins Build Pipeline interface. At the top, there's a navigation bar with links for Dashboard, pipeline, and a search bar. Below it, the title "Build Pipeline" is centered. A toolbar with icons for Run, History, Configure, Add Step, Delete, and Manage is visible. The main area displays two completed pipeline runs. The first run, "#2 clone-1", is highlighted in green and was completed on Nov 6, 2024 at 6:13:31 PM, taking 0.13 sec by user vinay. The second run, "#1 clone-1", is also highlighted in green and was completed on Nov 6, 2024 at 6:13:29 PM, taking 0.16 sec by user vinay. On the left, there's a sidebar titled "Pipeline" with sections for #2 and #1. The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.

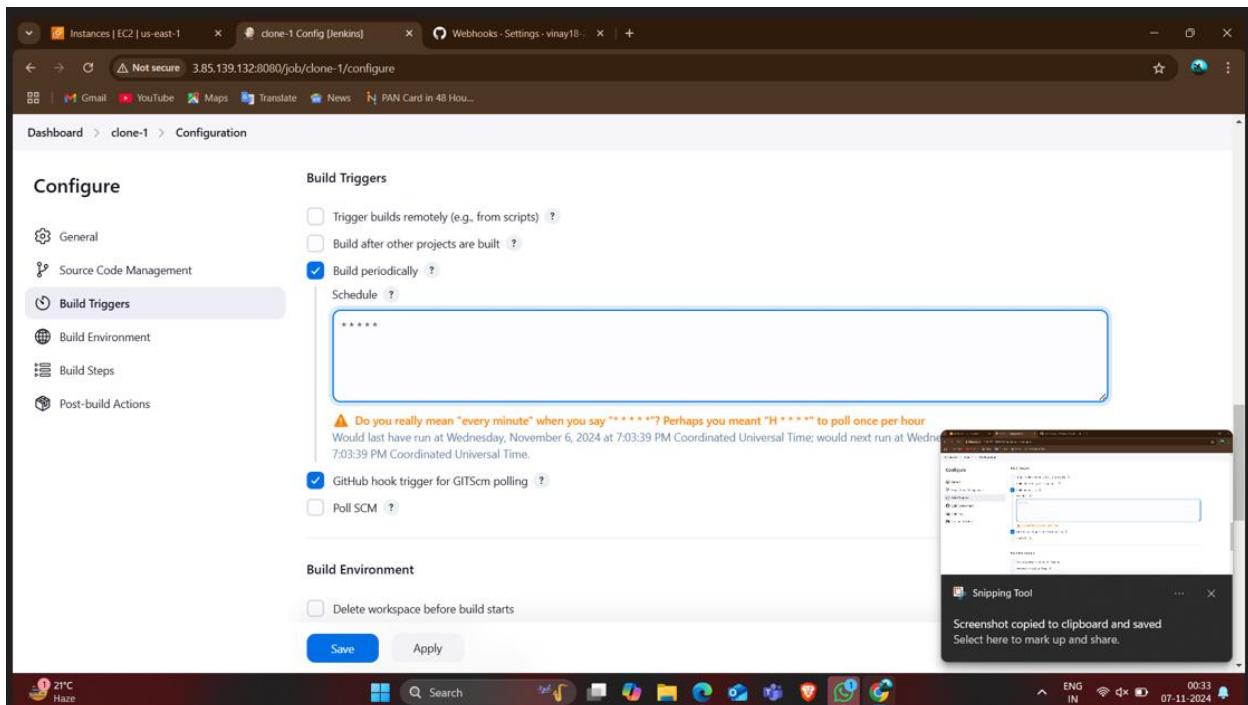
- ❖ Now set the buildperiodically ,webhooks and poll scm.

The screenshot shows the Jenkins job configuration page for "clone-1". The left sidebar lists sections: General, Source Code Management, Build Triggers (which is selected and highlighted in blue), Build Environment, Build Steps, and Post-build Actions. The main content area is titled "Build Triggers". Under "Configure", there are several options with checkboxes: Trigger builds remotely, Build after other projects are built, Build periodically (which is checked), and GitHub hook trigger for GITScm polling. Below these is a section for "Poll SCM" which is also checked. A "Schedule" field contains the cron expression "\* \* \* \* \*". A warning message below the schedule says: "⚠️ Do you really mean "every minute" when you say "\* \* \* \* \*"? Perhaps you meant "H \* \* \* \* \*" to poll once per hour. Would last have run at Wednesday, November 6, 2024 at 6:53:29 PM Coordinated Universal Time; would next run at Wednesday, November 6, 2024 at 6:53:29 PM Coordinated Universal Time." At the bottom of the page are "Save" and "Apply" buttons. The bottom of the screen shows a Windows taskbar with various pinned icons and system status indicators.

❖ Output of pollscm.



❖ Set the buildperiodically and webhooks.



## ❖ Output.

