

# PROJECT-3



BY RAMAKRISHNA  
SANGANAMONI

# DEPLOYMENT OF PYTHON APPLICATIONS WITH DIFFERENT TYPES OF METHODS



- Python is a high-level, versatile programming language known for its simplicity and readability. It was created by Guido van Rossum and first released in 1991. Python emphasizes code readability with its clean and easy-to-understand syntax, which makes it an excellent language for beginners and professionals alike

**1. Simple and Easy to Learn:** Python has a straightforward syntax that emphasizes readability and reduces the cost of program maintenance.

**2. High-Level Language:** Python abstracts many low-level details, making it easier to focus on solving problems rather than dealing with complex system details.

**3. Interpreted:** Python code is executed line by line by an interpreter, which allows for rapid development and debugging.

**4. Dynamically Typed:** Python is dynamically typed, meaning you don't need to declare the type of a variable explicitly. This makes development faster, but it also means that type-related errors might not be caught until runtime.

**5. Extensive Standard Library:** Python comes with a large standard library that provides support for many common programming tasks, such as string processing, file I/O, network protocols, and more.

**6. Cross-Platform:** Python is available for various platforms, including Windows, macOS, and Linux, making it suitable for developing applications that can run on different operating systems.

**7. Object-Oriented:** Python supports both procedural and object-oriented programming paradigms. It allows you to define classes and create objects with properties and methods.

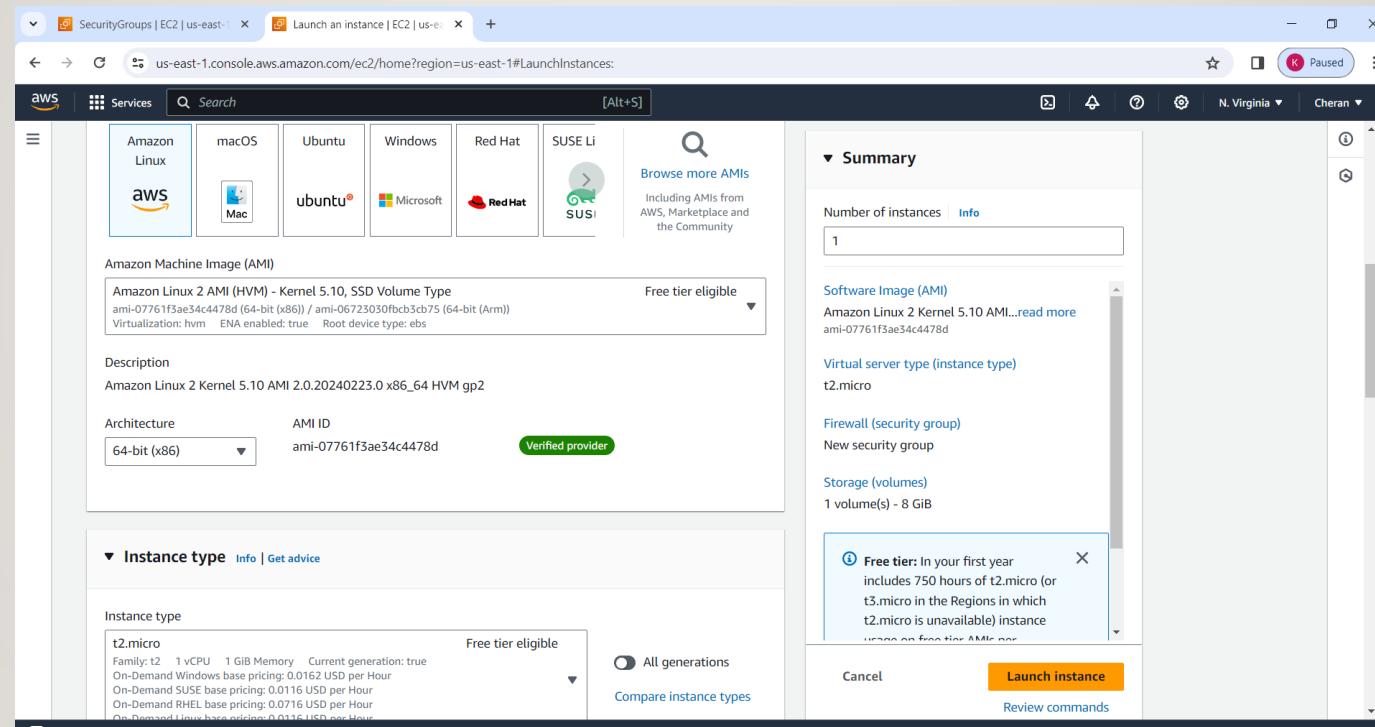
**8. Community and Ecosystem:** Python has a large and active community that contributes to libraries and frameworks, expanding its capabilities for various domains such as web development, data science, machine learning, artificial intelligence, scientific computing, and more.

## METHOD - I

BUILD AND DEPLOY PYTHON APPLICATIONS IN  
MANUAL PROCESS

---

- Open AWS console and login with aws account credentials
- Launch an ec2 instance type as **t2.micro** and port numbers **22** and **80** given in security group



After we launch an instance connect it terminal

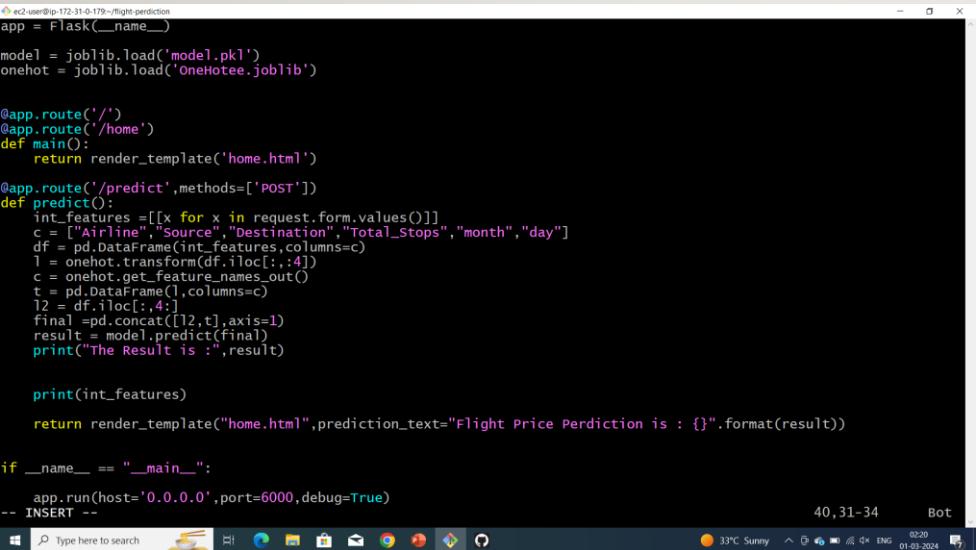
Next we are giving commands for **SUDO YUM UPDATE**

Next we need to install git **SUDO YUM INSTALL GIT -Y**

After installing

```
ec2-user@ip-172-31-0-179:~$ git clone https://github.com/Gouserabbani44/flight-perdiction.git
Cloning into 'flight-perdiction'...
remote: Enumerating objects: 10831, done.
remote: Counting objects: 100% (15/15), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 10831 (delta 6), reused 6 (delta 0), pack-reused 10816
Receiving objects: 100% (10831/10831), 84.41 MiB | 22.00 MiB/s, done.
Resolving deltas: 100% (877/877), done.
Updating files: 100% (10351/10351), done.
[ec2-user@ip-172-31-0-179 ~]$
```

- Now install python with commands “`sudo-amazon-linux-extras install python3.8`”
- Install python runtime in requirements.txt <`pip3 install -r requirements.txt`>
- Here we have seen the data and change host and port numbers



```

ec2-user@ip-172-31-0-179:~/flight-perdiction
app = Flask(__name__)

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

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

@app.route('/predict', methods=['POST'])
def predict():
    int_features = [[x for x in request.form.values()]]
    c = ['Airline', 'Source', 'Destination', 'Total_Stops', 'month', 'day']
    df = pd.DataFrame(int_features, columns=c)
    l = onehot.transform(df.iloc[:, :4])
    c = onehot.get_feature_names_out()
    t = pd.DataFrame(l, columns=c)
    final = pd.concat([t, df], axis=1)
    result = model.predict(final)
    print("The Result is :", result)

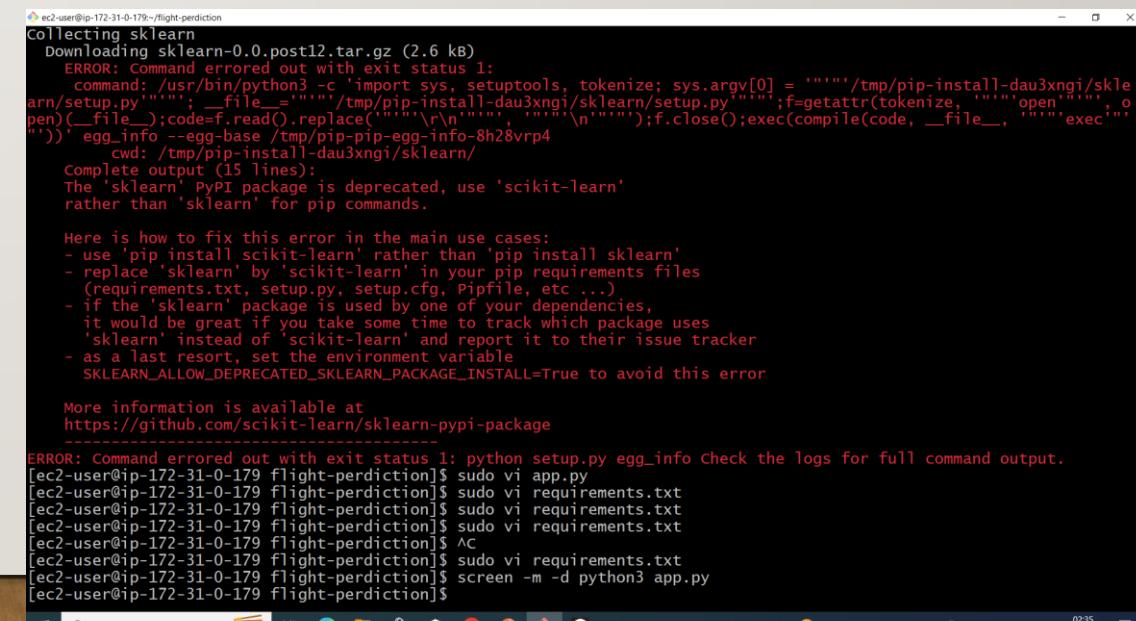
    print(int_features)

    return render_template("home.html", prediction_text="Flight Price Perdition is : {}".format(result))

if __name__ == "__main__":
    app.run(host='0.0.0.0', port=6000, debug=True)
-- INSERT --

```

37	mono	available	[ =5.x =stable ]
38	nginx1	available	[ =stable ]
40	mock	available	[ =stable ]
43	livepatch	available	[ =stable ]
44	python3.8	available	[ =stable ]
45	haproxy2	available	[ =stable ]
46	collectd	available	[ =stable ]
47	aws-nitro-enclaves-cli	available	[ =stable ]
48	R4	available	[ =stable ]



```

ec2-user@ip-172-31-0-179:~/flight-perdiction
Collecting sklearn
  Downloading sklearn-0.0.post12.tar.gz (2.6 kB)
    ERROR: Command errored out with exit status 1:
      command: /usr/bin/python3 -c 'import sys, setuptools, tokenize; sys.argv[0] = '../../../../../tmp/pip-install-dau3xngi/sklearn/setup.py'; f=getattr(tokenize, "'open'", open)(__file__);code=f.read().replace('"','\\n"');f.close();exec(compile(code, __file__, "'exec'"))
          cwd: /tmp/pip-install-dau3xngi/sklearn/
    Complete output (15 lines):
    The 'sklearn' PyPI package is deprecated, use 'scikit-learn' rather than 'sklearn' for pip commands.

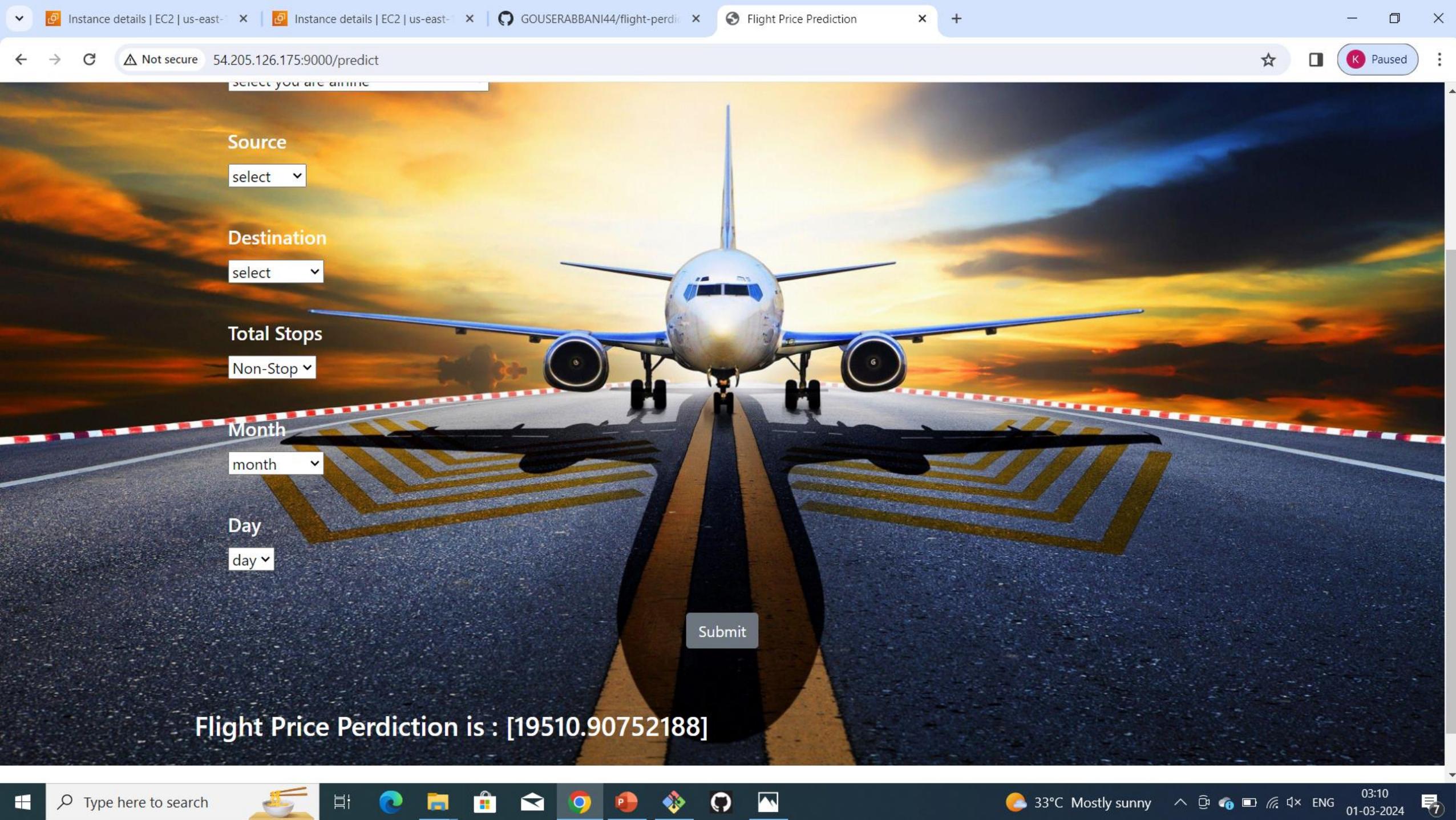
    Here is how to fix this error in the main use cases:
    - use 'pip install scikit-learn' rather than 'pip install sklearn'
    - replace 'sklearn' by 'scikit-learn' in your pip requirements files (requirements.txt, setup.py, setup.cfg, Pipfile, etc ...)
    - if the 'sklearn' package is used by one of your dependencies, it would be great if you take some time to track which package uses 'sklearn' instead of 'scikit-learn' and report it to their issue tracker
    - as a last resort, set the environment variable SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True to avoid this error

    More information is available at
    https://github.com/scikit-learn/sklearn-pypi-package
    -----
    ERROR: Command errored out with exit status 1: python setup.py egg_info Check the logs for full command output.
[ec2-user@ip-172-31-0-179 flight-perdiction]$ sudo vi app.py
[ec2-user@ip-172-31-0-179 flight-perdiction]$ sudo vi requirements.txt
[ec2-user@ip-172-31-0-179 flight-perdiction]$ sudo vi requirements.txt
[ec2-user@ip-172-31-0-179 flight-perdiction]$ sudo vi requirements.txt
[ec2-user@ip-172-31-0-179 flight-perdiction]$ ^C
[ec2-user@ip-172-31-0-179 flight-perdiction]$ sudo requirements.txt
[ec2-user@ip-172-31-0-179 flight-perdiction]$ screen -m -d python3 app.py
[ec2-user@ip-172-31-0-179 flight-perdiction]$
```

- Open the app.py file and change the versions of of packages
  - After that we execute the command “screen -m -d python3 app.py”
  - Copy the public ip along with what we have given the port number
- 

```
ec2-user@ip-172-31-0-37:~/Medical-Insurance
click==8.1.3
colorama==0.4.5
Flask==2.2.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
MarkupSafe==2.1.1
numpy==1.21.6
pandas==1.3.5
python-dateutil==2.8.2
pytz==2022.2.1
scikit-learn==0.24.2
scipy==1.7.3
six==1.16.0
sklearn==0.0
threadpoolctl==3.1.0
werkzeug==2.2.2
zipp==3.8.1
gunicorn==20.0.4
~
```

```
ec2-user@ip-172-31-0-179:~/flight-perdiction
Requirement already satisfied: Jinja2 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 6)) (3.1.3)
Requirement already satisfied: joblib in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 7)) (1.3.2)
Requirement already satisfied: MarkupSafe in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 8)) (2.1.5)
Requirement already satisfied: numpy in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 9)) (1.21.6)
Requirement already satisfied: pandas in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 10)) (1.3.5)
Requirement already satisfied: python-dateutil in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 11)) (2.9.0)
Requirement already satisfied: pytz in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 12)) (2024.1)
Requirement already satisfied: scikit-learn in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 13)) (1.0.2)
Requirement already satisfied: scipy in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 14)) (1.7.3)
Requirement already satisfied: six in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 15)) (1.16.0)
Requirement already satisfied: threadpoolctl in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 17)) (3.1.0)
Requirement already satisfied: werkzeug in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 18)) (2.2.3)
Requirement already satisfied: zipp in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 19)) (3.15.0)
Requirement already satisfied: gunicorn in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 20)) (21.2.0)
Requirement already satisfied: typing-extensions>=3.6.4; python_version < "3.8" in /home/ec2-user/.local/lib/python3.7/site-packages (from importlib-metadata->-r requirements.txt (line 4)) (4.7.1)
Requirement already satisfied: packaging in /home/ec2-user/.local/lib/python3.7/site-packages (from gunicorn->-r requirements.txt (line 20)) (23.2)
[ec2-user@ip-172-31-0-179 flight-perdiction]$ screen -m -d python3 app.py
[ec2-user@ip-172-31-0-179 flight-perdiction]$
```



# PENGUIN

---

- Here also we are doing the same process as usual before flight prediction
- First clone the repository of python applications and open the files which we will host file
- Next we execute the commands for changing versions of app.py and requirements.txt files
- With commands <`pip3 install -r requirements.txt`>
  - “`screen -m -d python3 app.py`”
- After execution this commands next copy the public ip and browse it in the browser

Here we can see penguin page will host

Not secure 3.93.169.27:5000/predict K Paused

# PENGUIN

island

Biscoe

sex

Female

culmen\_length\_mm

culmen\_depth\_mm

flipper\_length\_mm

Submit

Error occurred during prediction

The screenshot shows a web application interface for a penguin prediction model. At the top, the URL is 3.93.169.27:5000/predict. The main title is 'PENGUIN'. Below it, there are several input fields and dropdown menus:

- island:** A dropdown menu showing 'Biscoe'.
- sex:** A dropdown menu showing 'Female'.
- culmen\_length\_mm:** An empty input field.
- culmen\_depth\_mm:** An empty input field.
- flipper\_length\_mm:** An empty input field.

At the bottom center is a blue 'Submit' button. Below the button, the text 'Error occurred during prediction' is displayed in bold black font.

# Car prediction

- Clone the repository of python applications and open files
- Change the versions of files and data after that we execute the commands
- ~~<pip3 install -r requirements.txt >~~
- ~~<screen -m -d python3 app.py>~~

```
ec2-user@ip-172-31-0-37:~/Medical-Insurance
click==8.1.3
colorama==0.4.5
Flask==2.2.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
MarkupSafe==2.1.1
numpy==1.21.6
pandas==1.3.5
python-dateutil==2.8.2
pytz==2022.2.1
scikit-learn==0.24.2
scipy==1.7.3
six==1.16.0
sklearn==0.0
threadpoolctl==3.1.0
Werkzeug==2.2.2
zipp==3.8.1
gunicorn==20.0.4
~
```

## Second Hand Car Prediction

What Is the Fuel type?

Petrol ▾

Are you A Dealer or Individual

Dealer ▾

Transmission type

Manual C ▾

What is the Showroom Price?(In lakhs)

How Many Kilometers Drived?

How much owners previously had the car(0 or 1 or 3) ?



# IN NEWS

---

Launch an ec2 instance and connect with terminal

Clone the repository url for hosting in news using python application

In this process we need to install python

- Clone the repository of python applications and open files
- Change the versions of files and data after that we execute the commands
- `<pip3 install -r requirements.txt >`
- `<screen -m -d python3 app.py>`

- Here we are using the command `streamlit run app.py --server.port 9000`
  - After giving this command it will give the url's we copy the external url and browse it in the browser it host the in news
- 

Streamlit is a Python library used for creating web applications with simple Python scripts.

```
ec2-user@ip-172-31-0-37:~/InNews
Pillow
newspaper3k
urllib3>=1.25.4,<2.0
streamlit
beautifulsoup4
protobuf==3.20.0
jinja2>=3.0
~
~
```

```
ec2-user@ip-172-31-0-37:~/InNews
requirements.txt (line 4)) (4.0.11)
Requirement already satisfied: jsonschema>=3.0 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (4.17.3)
Requirement already satisfied: toolz in /home/ec2-user/.local/lib/python3.7/site-packages (from altair<6,>=0.12.1)
Requirement already satisfied: markdown-it-py>=2.2.0 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (2.17.2)
Requirement already satisfied: zipp>=0.5 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (3.15.0)
Requirement already satisfied: tzdata; python_version >= "3.6" in /home/ec2-user/.local/lib/python3.7/tzdata (from -r requirements.txt (line 4)) (2024.1)
Requirement already satisfied: smmap<6,>=3.0.1 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (5.0.1)
Requirement already satisfied: pyrsistent!=0.17.0,!>=0.17.1,!<0.17.2,>=0.14.0 in /home/ec2-user/.local/lib/python3.7/pyschema (from -r requirements.txt (line 4)) (0.19.3)
Requirement already satisfied: importlib-resources>=1.4.0; python_version < "3.9" in /home/ec2-user/.local/lib/python3.7/importlib/resources (from -r requirements.txt (line 4)) (5.12.0)
Requirement already satisfied: attrs>=17.4.0 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (23.2.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10; python_version < "3.9" in /home/ec2-user/.local/lib/python3.7/pkgutil (from -r requirements.txt (line 4)) (1.3.10)
Requirement already satisfied: mdurl~0.1 in /home/ec2-user/.local/lib/python3.7/site-packages (from -r requirements.txt (line 4)) (0.1.2)
[ec2-user@ip-172-31-0-37 InNews]$ streamlit run App.py --server.port 9000
Usage: streamlit run [OPTIONS] TARGET [ARGS]...
Try 'streamlit run --help' for help.

Error: No such option: --server.port 9000 (Possible options: --server.cookieSecret, --server.port)
[ec2-user@ip-172-31-0-37 InNews]$ streamlit run App.py --server.port 9000
Collecting usage statistics. To deactivate, set browser.gatherUsageStats to False.

You can now view your Streamlit app in your browser.

Network URL: http://172.31.0.37:9000
External URL: http://3.93.169.27:9000
```

# InNewsIN: A Summarised News



Select your Category

-Select-

Please select Type!!

Select your Category

Trending🔥 News

Here is the Trending🔥 news for you

Number of News:

5

5

25

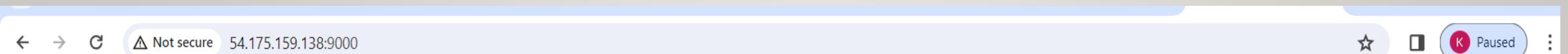
(1) Israel-Gaza war: Kamala Harris urges more aid for starving Gazans - BBC.com



# Know your Medical Insurance Charges

**Sex** male**Smoker** no**Region** Southwest**Age****BMI****Children****Submit**

# PORTFOLIO



Anudeep Reddy



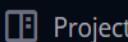
Home



About



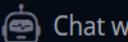
Resume



Projects



Contact



Chat with my bot

# Hey I'm Anudeep Reddy

I'm a passionate Data Scientist fro

# AGRI

```
ec2-user@ip-172-31-0-32:~/Agri
click==8.1.3
colorama==0.4.5
Flask==2.2.2
importlib-metadata==4.12.0
itsdangerous==2.1.2
Jinja2==3.1.2
joblib==1.1.0
MarkupSafe==2.1.1
numpy==1.21.6
pandas==1.3.5
python-dateutil==2.8.2
pytz==2022.2.1
scikit-learn==0.24.2
scipy==1.7.3
six==1.16.0
sklearn==0.0
threadpoolctl==3.1.0
werkzeug==2.2.2
zipp==3.8.1
gunicorn==20.0.4
~
```



# Crop Recommendation

Nitrogen

Phosphorous

Potassium

Temperature(C)

Humidity

PH Value

Rainfall

PREDICT

# FISH

Click to go back, hold to see history 1.213.129.186:8000



## FISH WEIGHT

**Species \***

Bream

**Vertical length in cm \***

23.2

**Diagonal length in cm \***

25.4

**Cross length in cm \***

30.0

**Height \***

11.5200

**Width \***

4.0200

**SUBMIT**

# LIVER PATIENT

Male

0.03

0.02



0.02

0.04

0.02

0.03

0.02

0.03

0.02

Submit!

# HEARING

A screenshot of a web browser window titled "Hearing Test". The browser has multiple tabs open, including "Connect to instance | EC2 | us-east-1", "flight-prediction/app.py at master · GitHub", and "ChatGPT". The main content area 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 button labeled "Predict Now!" is positioned below these fields.

Not secure 3.129.45.77:9000

Gmail YouTube Maps All Bookmarks

Predict SERVICES CONTACT

# Hearing Test

Age \* Acura

Physical Score \* ILX

Predict Now!

# FUEL CONSUMPTION

Not secure 54.175.159.138:9000 K Paused

## Fuel Consumption

Make  Engine Size

Model  Cylinders

Vechile Class  CO2 emission

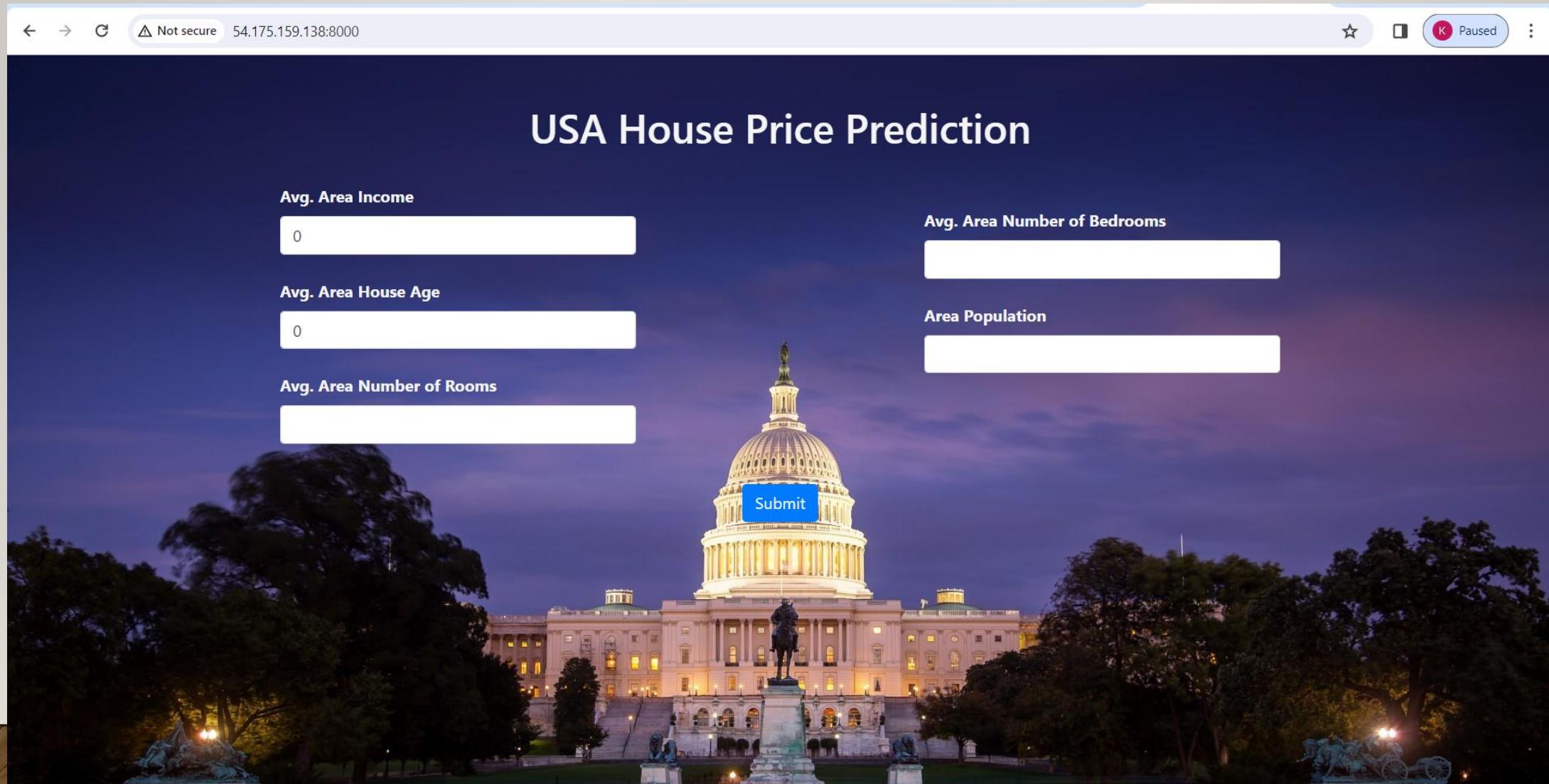
Transmission  CO2 Rating

Fuel Type  SmokeRate



Predict Now!

# USA- HOUSING



A screenshot of a web browser window showing a house price prediction form. The background is a photograph of the United States Capitol building at night, illuminated against a dark sky. The browser's address bar shows "Not secure 54.175.159.138:8000". The title of the page is "USA House Price Prediction". The form consists of five input fields: "Avg. Area Income" (value 0), "Avg. Area Number of Bedrooms" (empty), "Avg. Area House Age" (value 0), "Area Population" (empty), and "Avg. Area Number of Rooms" (empty). A blue "Submit" button is centered below the input fields.

△ Not secure 54.175.159.138:8000

K Paused

## USA House Price Prediction

Avg. Area Income

Avg. Area House Age

Avg. Area Number of Rooms

Avg. Area Number of Bedrooms

Area Population

Submit

# MY FUEL

Not secure 54.175.159.138:7000 Paused

Home Contact Service

## Fuel Consumption

Make

Acura

Model

Cylinders

Vechile Class

Compact

Transmission

AM8

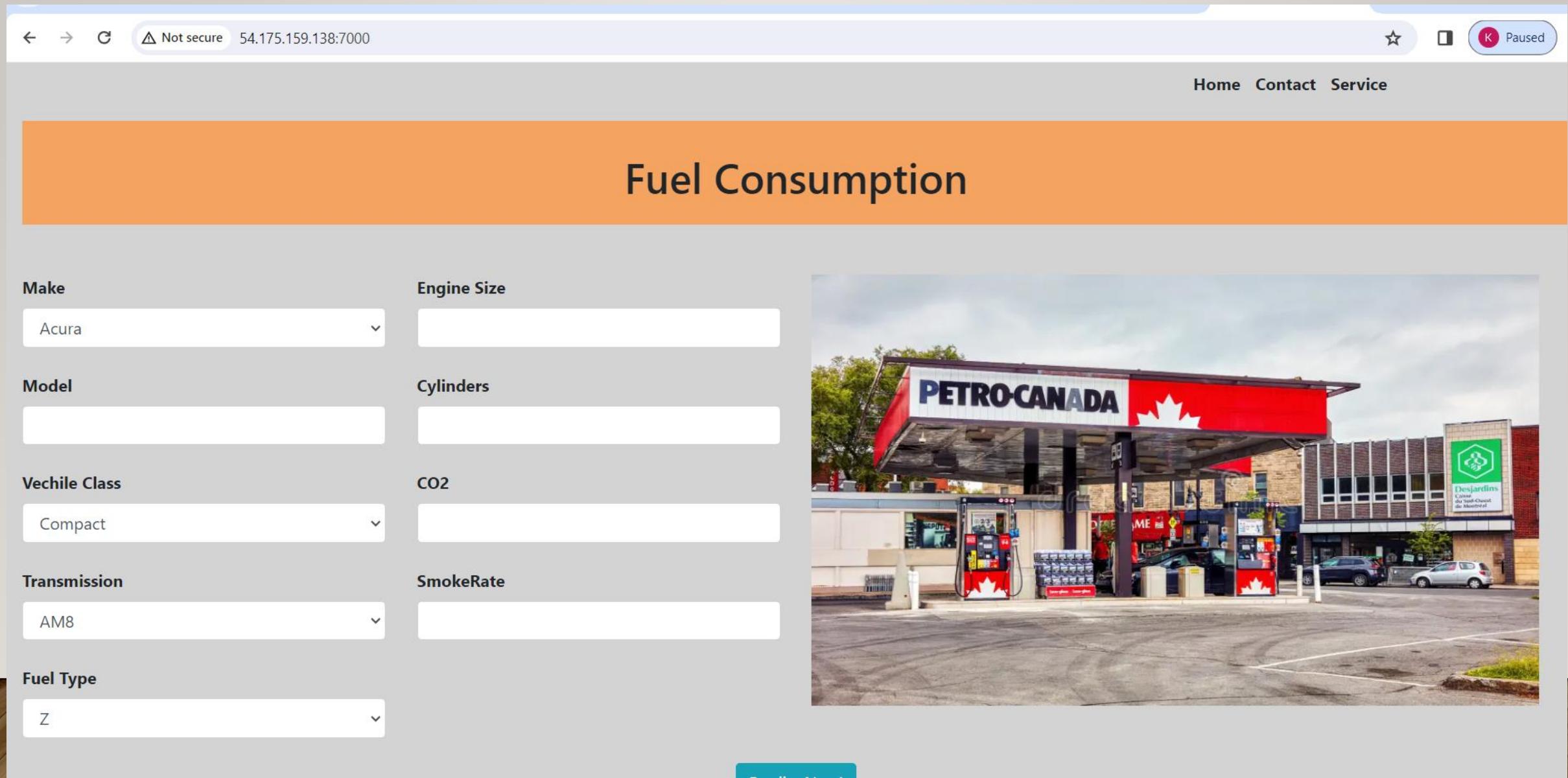
Fuel Type

Z

Engine Size

CO2

SmokeRate



The image shows a modern gas station with a red and white canopy featuring the 'PETRO-CANADA' logo and a large Canadian maple leaf. The station has several fuel pumps and is located in front of a brick building with a sign for 'Desjardins'.

## Method -2

---

BUILD AND DEPLOY PYTHON APPLICATION WITH EC2  
INSTANCE (USER DATA)

procedure

Launch an ec2 instance by giving instance name and select an AMI  
Select keypair , vpc, and subnets and security groups

The screenshot shows the AWS EC2 'Launch an instance' wizard. The top navigation bar includes the AWS logo, a 'Services' dropdown, a search bar, and keyboard shortcuts. The main breadcrumb path is 'EC2 > Instances > Launch an instance'. The left sidebar has a 'Summary' section with a collapsed arrow. The main content area starts with a 'Launch an instance' heading and a brief description of what EC2 allows. Below this is a 'Name and tags' section where the 'Name' field contains 'python -2'. A 'Software Image (AMI)' section lists 'Amazon Linux 2023 AMI 2023.3.2...' with a 'read more' link. The 'Virtual server type (instance type)' is set to 't2.micro'. Under 'Application and OS Images (Amazon Machine Image)', there's a note about AMIs being templates for software configuration, followed by a search bar. A 'Quick Start' section at the bottom features links for 'Amazon', 'macOS', 'Ubuntu', 'Windows', 'Red Hat', and 'SUSE'. On the right, a summary panel shows 'Number of instances' (1), 'Software Image (AMI)' (Amazon Linux 2023 AMI 2023.3.2...), 'Virtual server type (instance type)' (t2.micro), 'Firewall (security group)' (New security group), 'Storage (volumes)' (1 volume(s) - 8 GiB), and a 'Free tier' note. At the bottom right are 'Cancel', 'Launch instance', and 'Review commands' buttons.

EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags Info**

Name

python -2

Add additional tags

**Application and OS Images (Amazon Machine Image) Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon | macOS | Ubuntu | Windows | Red Hat | SUSE | i

**Summary**

Number of instances Info

1

Software Image (AMI)

Amazon Linux 2023 AMI 2023.3.2...read more

ami-0d7a109bf30624c99

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month

Cancel

Launch instance

Review commands

AWS Services Search [Alt+S] □ ⓘ

Amazon Machine Image (AMI)

Amazon Linux 2 AMI (HVM) - Kernel 5.10, SSD Volume Type  
ami-02d7fd1c2af6eed0 (64-bit (x86)) / ami-03c5a38d438cf3d3e (64-bit (Arm))  
Virtualization: hvm ENA enabled: true Root device type: ebs  
Free tier eligible

Description  
Amazon Linux 2 Kernel 5.10 AMI 2.0.20240306.2 x86\_64 HVM gp2

Architecture: 64-bit (x86) AMI ID: ami-02d7fd1c2af6eed0 Verified provider

Instance type

t2.micro Free tier eligible  
Family: t2 1 vCPU 1 GiB Memory Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
All generations

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more  
ami-02d7fd1c2af6eed0

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per year.

Cancel Launch instance Review commands

AWS Services Search [Alt+S] □ ⓘ

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: ram Create new key pair

Network settings Info

Network: vpc-00db2eefe23cdafe0 Subnet: subnet-0ccb5e8e7a506113 | python1 Auto-assign public IP: Info Enable Firewall (security groups): Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

create user data in additional settings

After that launch an ec2 instance

Check the version of repo and copy the public ip of an instance and browse it in the browser

Metadata response timeout [Alt+S]

Allow tags in metadata [Info](#)

User data - optional [Info](#)  
Upload a file with your user data or enter it in the field.

```
#!/bin/bash
sudo amazon-linux-extras install python3.8
sudo yum -y install git
git clone https://github.com/rama861/python-carperdition.git
cd python-carperdition
pip3 install -r requirements.txt
screen -m -d python3 app.py
```

User data has already been base64 encoded

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free-tier AMIs per month

[Cancel](#) [Launch instance](#) [Review commands](#)

EC2 Dashboard [X](#)

EC2 Global View

Events

Console-to-Code [Preview](#)

Instances [Instances](#)

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations [New](#)

Images [Images](#)

AMIs

AMI Catalog

Elastic Block Store [Elastic Block Store](#)

Volumes

Snapshots

EC2 > Instances > i-09decbe82d6188ae (python-2) [Info](#)

Updated less than a minute ago

Instance ID	Public IPv4 address	Private IPv4 addresses
i-09decbe82d6188ae (python-2)	107.21.71.92 <a href="#">Open address</a>	172.31.0.72
IPv6 address	Instance state	Public IPv4 DNS
-	Running	ec2-107-21-71-92.compute-1.amazonaws.com <a href="#">Open address</a>
Hostname type	Private IP DNS name (IPv4 only)	Elastic IP addresses
IP name: ip-172-31-0-72.ec2.internal	ip-172-31-0-72.ec2.internal	-
Answer private resource DNS name	Instance type	AWS Compute Optimizer finding
IPv4 (A)	t2.micro	<a href="#">Opt-in to AWS Compute Optimizer for recommendations</a> <a href="#">Learn more</a>
Auto-assigned IP address	VPC ID	Auto Scaling Group name
107.21.71.92 [Public IP]	vpc-00db2eefe23cdaf0 <a href="#">Open</a>	-
IAM Role	Subnet ID	
-	subnet-0cc8b5e8e7a506113 (python1) <a href="#">Open</a>	
IMDSv2		
Required		

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

## Second Hand Car Prediction

What Is the Fuel type?

Petrol ▾

Are you A Dealer or Individual

Dealer ▾

Transmission type

Manual C ▾

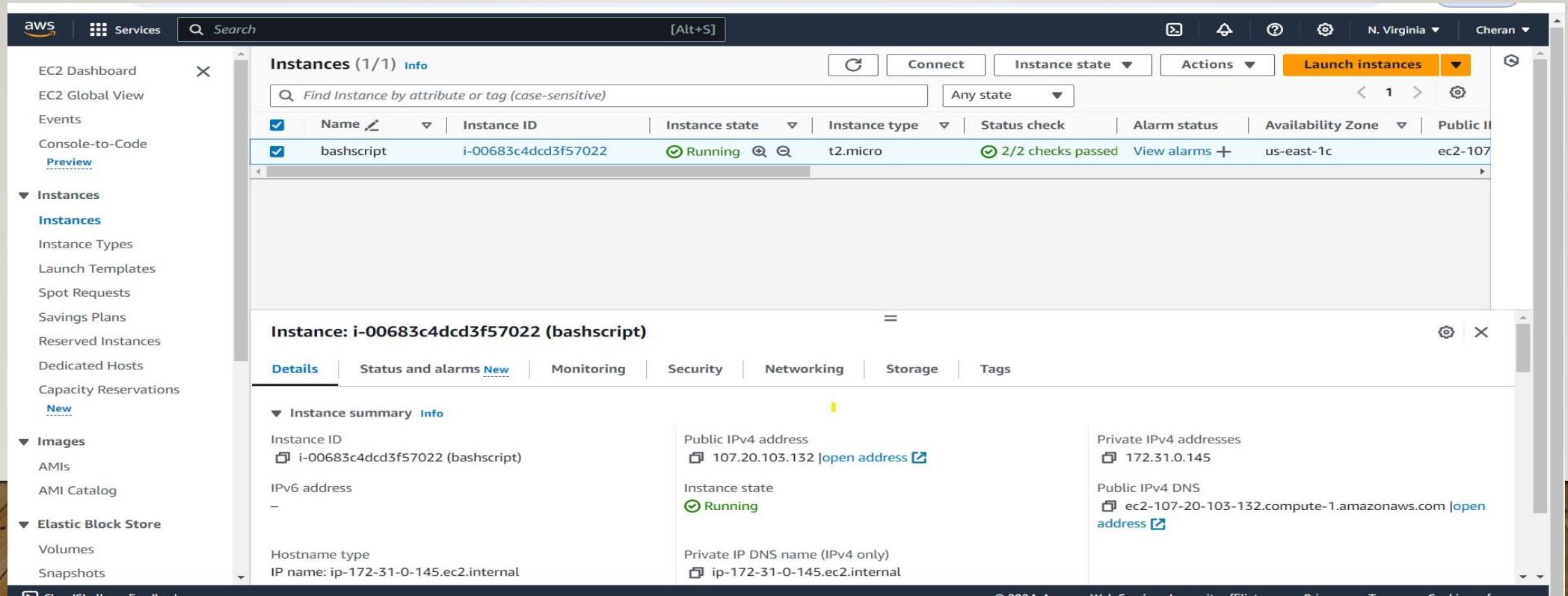
What is the Showroom Price?(In lakhs)

How Many Kilometers Drived?

How much owners previously had the car(0 or 1 or 3) ?

# METHOD- 3

## BUILD AND DEPLOY PYTHON APPLICATION USING BASHSCRIPT



Create a file with the name data.sh

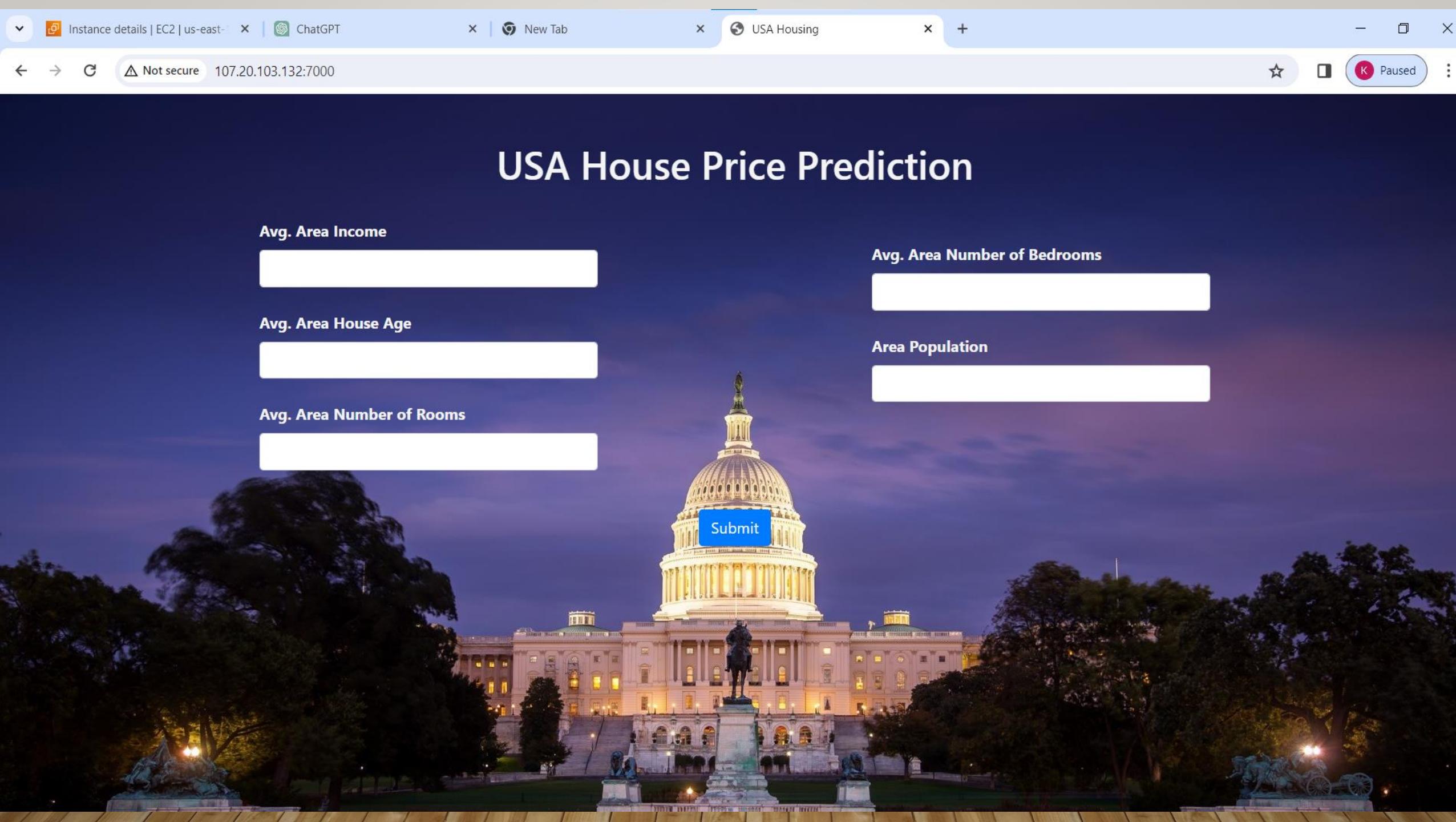
```
ec2-user@ip-172-31-0-145 ~]$ ls  
ata.sh USA-Housing-py  
ec2-user@ip-172-31-0-145 ~]$ cd USA-Housing-py/  
ec2-user@ip-172-31-0-145 USA-Housing-py]$ ls  
pp.py gitattributes Price Procfile regressor.pkl requirements.txt templates USA House.ipynb  
ec2-user@ip-172-31-0-145 USA-Housing-py]$ sudo vi app.py  
ec2-user@ip-172-31-0-145 USA-Housing-py]$ sudo vi app.py  
ec2-user@ip-172-31-0-145 USA-Housing-py]$ screen -m -d python3 app.py
```

After creating file in that file give the commands for install GITCLONE, and run commands

Below we seen the commands what I have given

```
#!/bin/bash  
sudo amazon-linux-extras install python3.8  
sudo yum -y install git  
git clone https://github.com/03Aziz02/USA-Housing-py.git  
cd USA-Housing-py  
pip3 install -r requirements.txt  
screen -m -d python3 app.py
```

Copy the public ip of an instance and browse it in the browser along with port number



# USA House Price Prediction

Avg. Area Income

Avg. Area House Age

Avg. Area Number of Rooms

Avg. Area Number of Bedrooms

Area Population

Submit

# METHOD -4

## BUILD AND DEPLOY PYTHON APPLICATIONS WITH GIT, GITHUB,AND JENKINS (EXECUTE SHELL)

---

### PROCEDURE

- Open AWS console management and login with aws account credentials
- In the search bar search for an ec2 instance and launch an instance with instance type as t2.micro and port numbers ssh 22 and all traffic

AWS Services Search [Alt+S] N. Virginia Cheran

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags [Info](#)

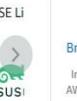
Name  Add additional tags

Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux  macOS  Ubuntu  Windows  Red Hat  SUSE Linux  Browse more AMIs [Info](#)  
Including AMIs from AWS, Marketplace and the Community

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free-tier AMIs [Learn more](#)

Cancel [Launch instance](#) Review commands

AWS Services Search [Alt+S] N. Virginia Cheran

Type [Info](#) Protocol [Info](#) Port range [Info](#)  
ssh TCP 22

Source type [Info](#) Source [Info](#) Description - optional [Info](#)  
Anywhere  0.0.0.0/0 [X](#)

Security group rule 2 (TCP, 80, 0.0.0.0/0) [Remove](#)

Type [Info](#) Protocol [Info](#) Port range [Info](#)  
HTTP TCP 80

Source type [Info](#) Source [Info](#) Description - optional [Info](#)  
Anywhere  0.0.0.0/0 [X](#)

Security group rule 3 (TCP, 8080, 0.0.0.0/0) [Remove](#)

Type [Info](#) Protocol [Info](#) Port range [Info](#)  
Custom TCP TCP 8080

Source type [Info](#) Source [Info](#) Description - optional [Info](#)  
Anywhere  0.0.0.0/0 [X](#)

Summary

Number of instances [Info](#) 1

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI... [read more](#)  
ami-07761f5ae34c4478d

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free-tier AMIs [Learn more](#)

Cancel [Launch instance](#) Review commands

```
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 -y

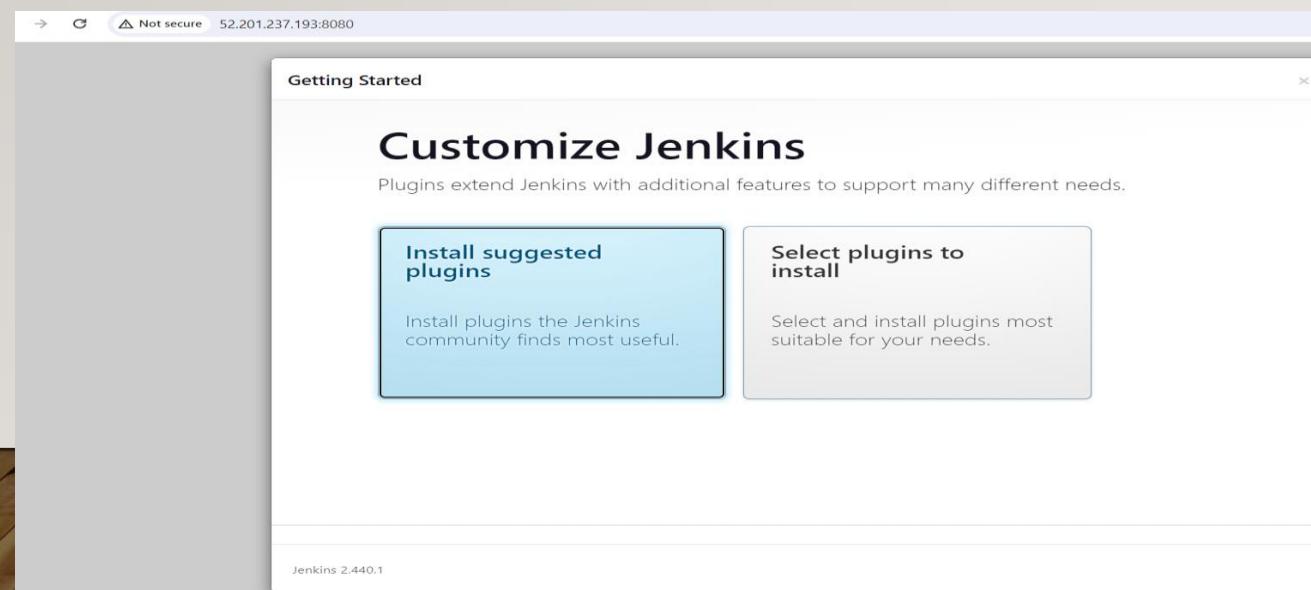
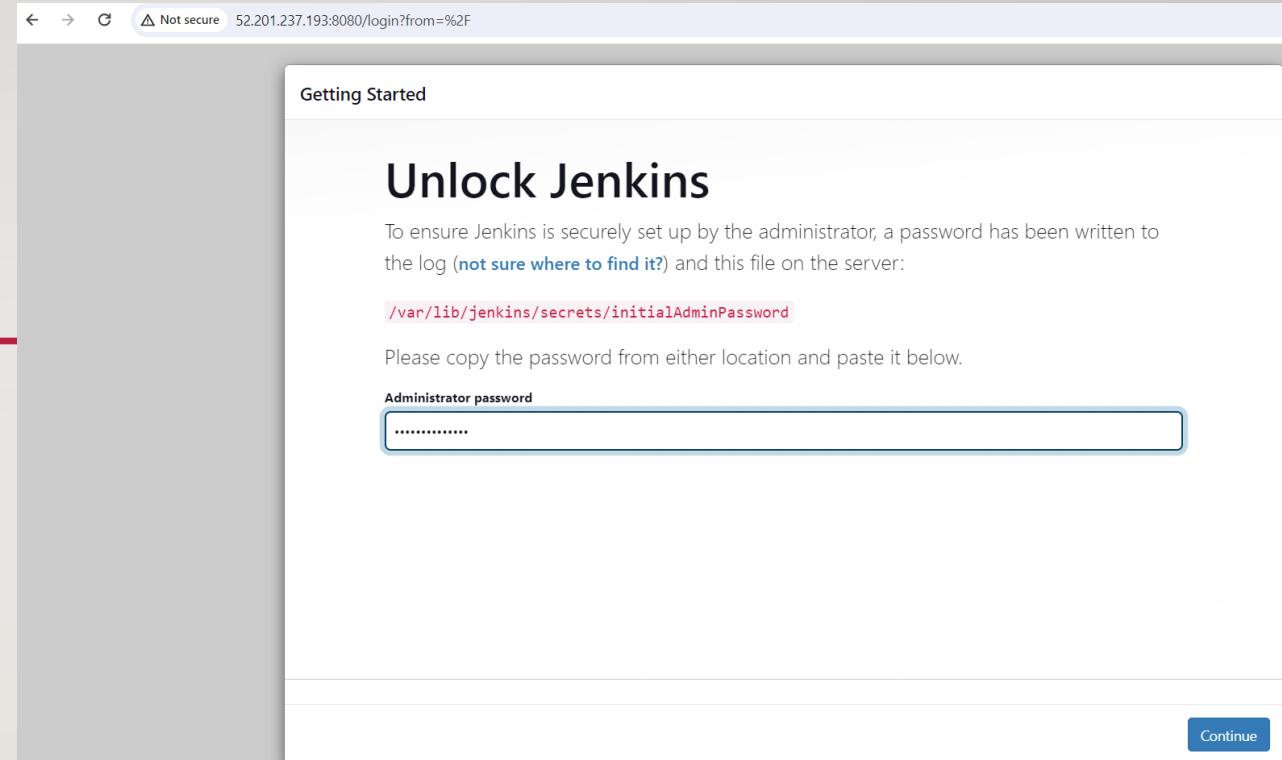

---


sudo yum install -y java-17*
sudo yum install -y Jenkins
sudo systemctl start jenkins
sudo systemctl enable jenkins
```

```
● jenkins.service - Jenkins Continuous Integration Server
   Loaded: loaded (/usr/lib/systemd/system/jenkins.service; disabled; vendor preset: disabled)
   Active: active (running) since Tue 2024-03-05 12:09:22 UTC; 35s ago
     Main PID: 3850 (java)
        CGroup: /system.slice/jenkins.service
                  └─3850 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Mar  5 12:08:50 ip-172-31-0-188.ec2.internal jenkins[3850]: 449a174c026c4ea2998bdb8006fb8fd4
Mar  5 12:08:50 ip-172-31-0-188.ec2.internal jenkins[3850]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar  5 12:08:50 ip-172-31-0-188.ec2.internal jenkins[3850]: ****
Mar  5 12:08:50 ip-172-31-0-188.ec2.internal jenkins[3850]: ****
Mar  5 12:08:50 ip-172-31-0-188.ec2.internal jenkins[3850]: ****
Mar  5 12:09:22 ip-172-31-0-188.ec2.internal jenkins[3850]: 2024-03-05 12:09:22.508+0000 [id=32]           INFO      jenkins.Initialization
Mar  5 12:09:22 ip-172-31-0-188.ec2.internal jenkins[3850]: 2024-03-05 12:09:22.546+0000 [id=25]           INFO      hudson.lifecycle
Mar  5 12:09:22 ip-172-31-0-188.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Mar  5 12:09:22 ip-172-31-0-188.ec2.internal jenkins[3850]: 2024-03-05 12:09:22.660+0000 [id=48]           INFO      h.m.Downloadaller
Mar  5 12:09:22 ip-172-31-0-188.ec2.internal jenkins[3850]: 2024-03-05 12:09:22.660+0000 [id=48]           INFO      hudson.util.Stream #1
Hint: Some lines were ellipsized, use -l to show in full.
```

- Here you can see the unlock Jenkins page you have given administrative password
- Select suggested pluggins



Now its getting started and asks some credentials for login after given the credentials it will give the [Jenkins url](#) next save and fish it the [official page](#) of Jenkins open

Getting Started

## Create First Admin User

Username: ram

Password: \*\*\*\*\*

Confirm password: \*\*\*\*\*

Full name: ramakrishna

E-mail address: ramakrishna2020@gmail.com

Jenkins 2.440.1

Skip and continue as admin Save and Continue

Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	✓ Gradle
✓ Pipeline	✓ GitHub Branch Source	✓ Pipeline: GitHub Groovy Libraries	✓ Pipeline: Stage View
✓ Git	✓ SSH Build Agents	✓ Matrix Authorization Strategy	✓ PAM Authentication
✓ LDAP	✓ Email Extension	✓ Mailer	✓ Dark Theme

Jenkins 2.440.1

\*\* Pipeline: Input Step  
\*\* Pipeline: Declarative Pipeline  
\*\* Java JSON Web Token (JWT)  
\*\* OkHttpClient  
\*\* GitHub API  
Git  
\*\* GitHub  
GitHub Branch Source  
Pipeline: GitHub Groovy Libraries  
\*\* Pipeline Graph Analysis  
\*\* Pipeline: REST API  
Pipeline: Stage View  
Git  
SSH Build Agents  
Matrix Authorization Strategy  
PAM Authentication  
LDAP  
Email Extension  
Mailer  
\*\* Theme Manager  
Dark Theme  
\*\* - required dependency

Getting Started

## Instance Configuration

Jenkins URL: http://52.201.237.193:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.440.1

After Jenkins page open next create a new job and give an item name and select project style as free style

Jenkins

Dashboard > All >

Enter an item name

clone-job  
» Required field

**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 folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

OK Branch Pipeline

For each commit of Pipeline projects according to defined branches in one SCM repository

Not secure 52.201.237.193:8080

Dashboard >

+ New Item

People

Build History

Manage Jenkins

My Views

Welcome to Jenkins!

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

Start building your software project

Create a job +

Build Queue

No builds in the queue.

Build Executor Status

1 Idle

2 Idle

Set up a distributed build

Set up an agent

Configure a cloud

Learn more about distributed builds ?

- In the source code management select git and give the repository url of python application files
- In build steps select an execute shell and give commands for execution

Configure

Source Code Management

None

Git ?

Repositories ?

Repository URL ?  
https://github.com/PavanGitPro/villa.git

Credentials ?  
- none -

+ Add ▾

Advanced ▾

Add Repository

Save Apply

Configure

General

Source Code Management

Build Triggers

Build Environment

Build Steps

Post-build Actions

Inspect build log for published build scans

Terminate a build if it's stuck

With Ant ?

Build Steps

Execute shell ?

Command

See [the list of available environment variables](#)

```
cd /var/lib/jenkins/workspace/clone-job
pip3 install -r requirements.txt
screen -m -d python3 app.py
```

Advanced ▾

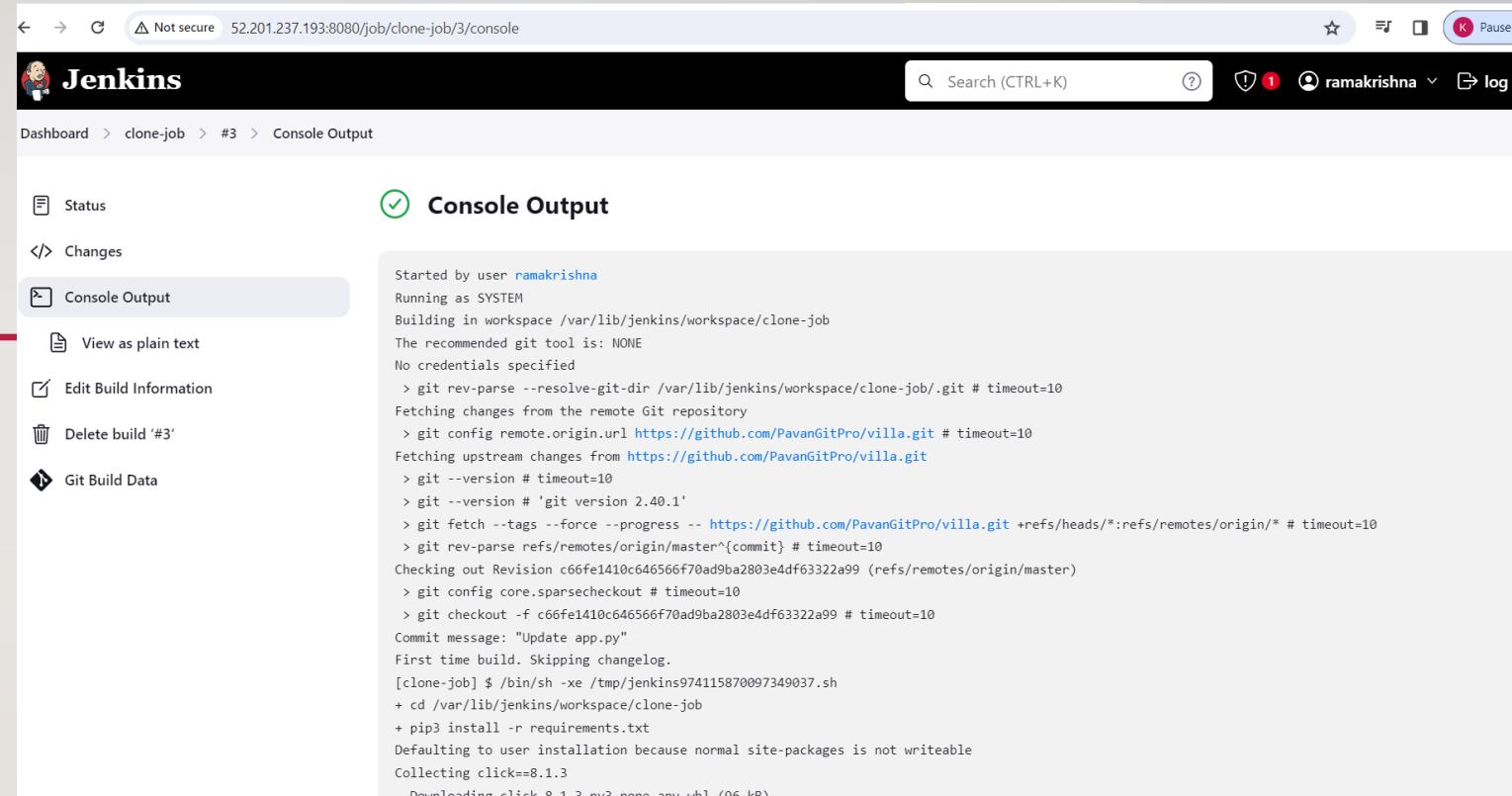
Add build step ▾

Save Apply

- ✓ Click on build now it builds and give the console output success

---

- ✓ Copy the public ip of an instance and browse it in the browser along with portnumber the page will be host



The screenshot shows a Jenkins job named "clone-job" with build #3. The "Console Output" tab is selected. The output log is displayed, starting with the message "Started by user ramakrishna". It details the git fetch and checkout process, followed by the execution of a shell script, pip3 install of requirements.txt, and finally the download of click-8.1.3-py3-none-any.whl. The build status is marked as "Success".

```

Started by user ramakrishna
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/PavanGitPro/villa.git # timeout=10
Fetching upstream changes from https://github.com/PavanGitPro/villa.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/PavanGitPro/villa.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^{commit} # timeout=10
Checking out Revision c66fe1410c646566f70ad9ba2803e4df63322a99 (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
> git checkout -f c66fe1410c646566f70ad9ba2803e4df63322a99 # timeout=10
Commit message: "Update app.py"
First time build. Skipping changelog.
[clone-job] $ /bin/sh -xe /tmp/jenkins974115870097349037.sh
+ cd /var/lib/jenkins/workspace/clone-job
+ pip3 install -r requirements.txt
Defaulting to user installation because normal site-packages is not writeable
Collecting click==8.1.3
  Downloading click-8.1.3-py3-none-any.whl (96 kB)

```

Consider adding this directory to PATH or, if you prefer to sup  
Successfully installed Flask-2.2.2 Jinja2-3.1.2 MarkupSafe-2.1.1  
itsdangerous-2.1.2 joblib-1.1.0 numpy-1.21.6 pandas-1.3.5 python-  
threadpoolctl-3.1.0 typing-extensions-4.7.1 zipp-3.8.1  
+ screen -m -d python3 app.py  
Finished: SUCCESS

# USA House Price Prediction

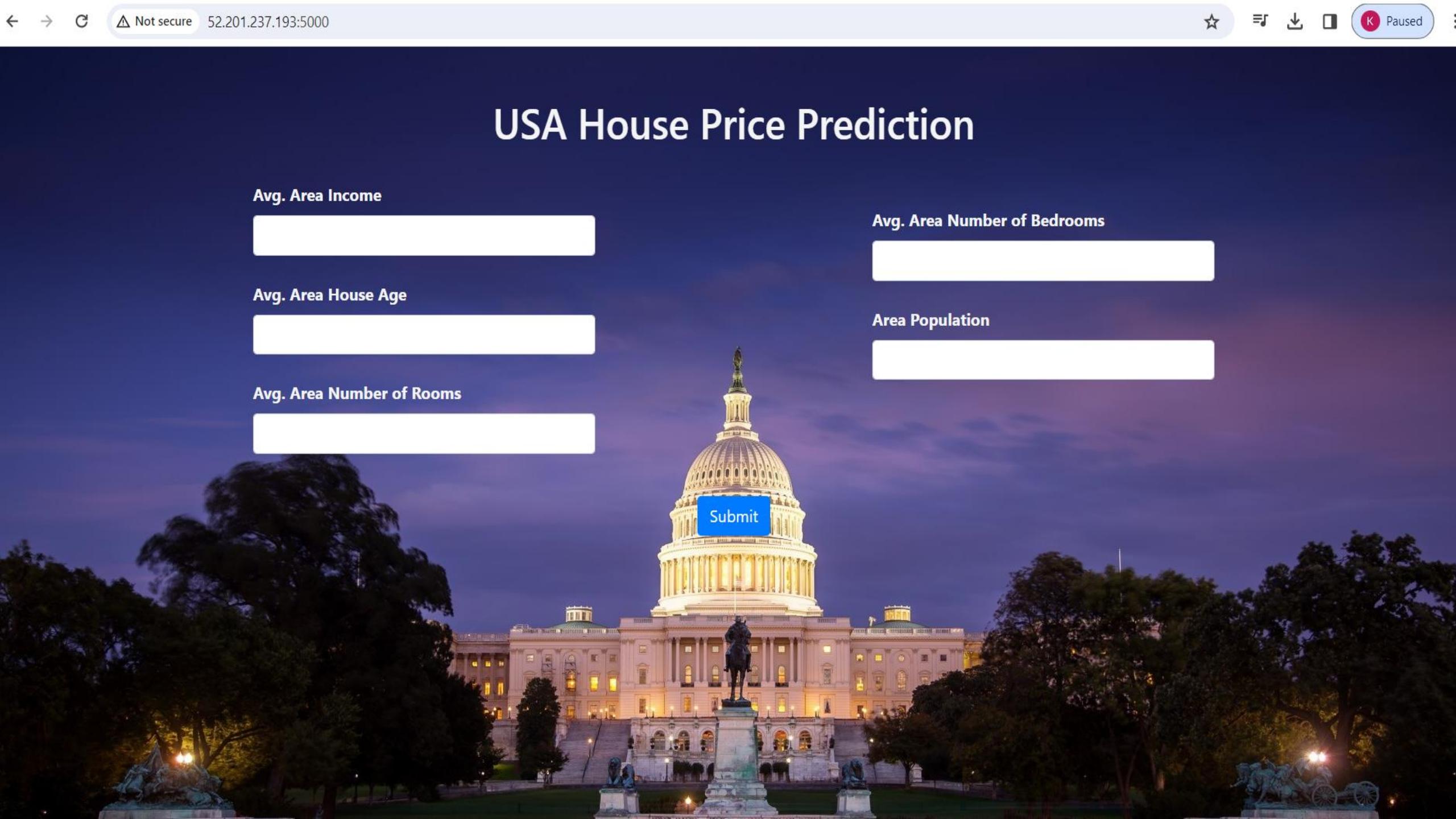
Avg. Area Income

Avg. Area House Age

Avg. Area Number of Rooms

Avg. Area Number of Bedrooms

Area Population



Submit

# METHOD -5

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

---

### procedure

- Open an AWS console and login with aws account credentials
- In the console search for EC2 service and open it launch an instance

**EC2 Dashboard**

**Instances (1/1) [Info](#)**

Find Instance by attribute or tag (case-sensitive)

Instance state = running

Clear filters

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
python-5	i-0faebabf2a8f5d68f	Running	t2.micro	2/2 checks passed	<a href="#">View alarms</a>	us-east-1c	ec2-18-20...

Unselect instance: python-5

**Instance: i-0faebabf2a8f5d68f (python-5)**

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

Instance summary

Instance details

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-07761f3ae34c4478d	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	amzn2-ami-kernel-5.10-hvm-2.0.20240223.0-x86_64-gp2	Disabled
Stop protection	Launch time	AMI location

EC2 Instances Launch an instance

## Launch an instance [Info](#)

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

### Name and tags [Info](#)

Name  Add additional tags

### Application and OS Images (Amazon Machine Image) [Info](#)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Software Image (AMI)  
Amazon Linux 2 Kernel 5.10 AMI... [read more](#)

Virtual server type (instance type)  
t2.micro

Firewall (security group)  
New security group

Storage (volumes)  
1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on frontier AMIs per year

**Launch instance**

AWS Services Search [Alt+S] N. Virginia Cheran

EC2 Dashboard Instances (1/1) Info

Find Instance by attribute or tag (case-sensitive)

Instance state = running Any state

Clear filters

Name Instance ID Instance state Instance type Status check Alarm status Availability Zone Public IPv4

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
python-5	i-0faebabf2a8f5d68f	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c	ec2-18-205-11-111

Unselect instance: python-5

Instance: i-0faebabf2a8f5d68f (python-5)

Details Status and alarms New Monitoring Security Networking Storage Tags

Instance summary Info

Instance details Info

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-07761f3ae34c4478d	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	amzn2-ami-kernel-5.10-hvm-2.0.20240223.0-x86_64-gp2	Disabled
Launch time	AMI location	

AMIs AMI Catalog

Elastic Block Store Volumes Snapshots

Stop protection

The screenshot shows the AWS EC2 Instances page with a single instance named "python-5" listed. The instance is running on an Amazon Linux AMI (Inferred) and has a t2.micro instance type. It has 2/2 checks passed and is located in the us-east-1c availability zone. The modal window for this instance shows detailed information like the AMI ID (ami-07761f3ae34c4478d), platform (Amazon Linux), and monitoring status (disabled). The modal also displays the AMI name (amzn2-ami-kernel-5.10-hvm-2.0.20240223.0-x86\_64-gp2), platform details (Linux/UNIX), and termination protection (Disabled).

- 
- ✓ sudo yum install -y yum-utils
  - ✓ sudo yum-config-manager --add-repo  
<https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo>
  - ✓ sudo yum -y install terraform

```
Installed:
  terraform.x86_64 0:1.7.4-1

Dependency Installed:
  git.x86_64 0:2.40.1-1.amzn2.0.1
  git-core-doc.noarch 0:2.40.1-1.amzn2.0.1
  perl-Git.noarch 0:2.40.1-1.amzn2.0.1
  git-core.x86_64 0:2.40.1-1.amzn2.0.1
  perl-Error.noarch 1:0.17020-2.amzn2
  perl-TermReadKey.x86_64 0:2.30-20.amzn2.0.2

Complete!
[ec2-user@ip-172-31-0-131 ~]$ aws configure
AWS Access Key ID [None]: AKIAQ3EGWRWVEQSSHI2V
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-172-31-0-131 ~]$
```

## DATA.SH

```
#!/bin/bash
sudo yum update -y
sudo yum install git -y
git clone https://github.com/rama861/python-fish.git
cd fish
pip3 install -r requirements.txt
screen -m -d python3 app.py
~
```

## Vpc.tf

```
#creating vpc
resource "aws_vpc" "demovpc" {
  cidr_block = "10.0.0.0/16"
  tags = {
    Name = "vpc"
  }
}
```

## Sg.tf

```
resource "aws_security_group" "sg1" {
  vpc_id = aws_vpc.demovpc.id
  ingress {
    from_port   = 8000
    to_port     = 8000
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  ingress {
    from_port   = 0
    to_port     = 0
    protocol    = "tcp"
    cidr_blocks = ["0.0.0.0/0"]
  }

  egress {
    from_port   = 0
    to_port     = 0
    protocol    = "-1"
    cidr_blocks = ["0.0.0.0/0"]
  }

  tags = {
    Name = "pysg"
  }
}

~
~
~
~
"sg.tf" 27L, 446B
```

## Instance.tf

```
resource "aws_instance" "python" {  
    ami           = "ami-07761f3ae34c4478d"  
    instance_type = "t2.micro"  
  
    key_name      = "shiva"  
    vpc_security_group_ids = [aws_security_group.sg1.id]  
    subnet_id     = aws_subnet.publicsubnet1.id  
    associate_public_ip_address = true  
    user_data     = file("data.sh")  
    tags = {  
        Name = "python-5"  
    }  
}
```

## Subnets.tf

```
resource "aws_subnet" "publicsubnet1" {  
    vpc_id          = aws_vpc демovpc.id  
    cidr_block     = "10.0.1.0/24"  
    map_public_ip_on_launch = true  
    availability_zone = "us-east-1a"  
  
    tags = {  
        Name = "subnet"  
    }  
}
```

## Rtb.tf

```
resource "aws_route_table" "route1" {  
    vpc_id = aws_vpc демовpc.id  
  
    route {  
        cidr_block = "0.0.0.0/0"  
        gateway_id = aws_internet_gateway.igpy.id  
    }  
  
    tags = {  
        Name = "routetable"  
    }  
}
```

```
aws_vpc.demovpc: Creating...
aws_vpc.demovpc: Creation complete after 1s [id=vpc-06fb797f359e5e814]
aws_subnet.publicsubnet1: Creating...
aws_security_group.sg1: Creating...
aws_route_table.route1: Creating...
aws_internet_gateway.igpy: Creating...
aws_internet_gateway.igpy: Creation complete after 1s [id=igw-0917b30a787ae9131]
aws_route_table.route1: Creation complete after 1s [id=rtb-0cf1fcfd3b69b022a4]
aws_security_group.sg1: Creation complete after 2s [id=sg-0c5a3ab354c920385]
aws_subnet.publicsubnet1: Still creating... [10s elapsed]
aws_subnet.publicsubnet1: Creation complete after 11s [id=subnet-09b05e792a118ac66]
aws_instance.python: Creating...
aws_route_table_association.route1: Creating...
aws_route_table_association.route1: Creation complete after 0s [id=rtbassoc-07943a2b5]
aws_instance.python: Still creating... [10s elapsed]
aws_instance.python: Still creating... [20s elapsed]
aws_instance.python: Still creating... [30s elapsed]
aws_instance.python: Creation complete after 32s [id=i-04ce0195873d4598f]
```

Apply complete! Resources: 7 added, 0 changed, 0 destroyed.

[ec2-user@ip-172-31-0-131 ~]\$ |

AWS Services Search [Alt+S] N. Virginia Cheran

EC2 Dashboard EC2 Global View Events Console-to-Code Preview Instances Instances Instance Types Launch Templates Spot Requests Savings Plans Reserved Instances Dedicated Hosts Capacity Reservations New Images AMIs AMI Catalog Elastic Block Store Volumes Snapshots

Instances (1/2) Info

Find Instance by attribute or tag (case-sensitive)

Instance state: Any state

Launch instances

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
python-5	i-0faebabf2a8f5d68f	Running	t2.micro	2/2 checks passed	View alarms	us-east-1c	ec2-18-209-55-10
<input checked="" type="checkbox"/> python-5	i-04ce0195873d4598f	Running	t2.micro	2/2 checks passed	View alarms	us-east-1a	-

Unselect instance: python-5

Instance: i-04ce0195873d4598f (python-5)

Details Status and alarms New Monitoring Security Networking Storage Tags

Instance summary Info

Instance details Info

Platform	AMI ID	Monitoring
Amazon Linux (Inferred)	ami-07761f3ae34c4478d	disabled
Platform details	AMI name	Termination protection
Linux/UNIX	amzn2-ami-kernel-5.10-hvm-2.0.20240223.0-x86_64-gp2	Disabled

AWS Services Search [Alt+S] N. Virginia Cheran

VPC dashboard EC2 Global View Filter by VPC: Select a VPC

Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
vpc	vpc-030d9ef01e54304e0	Available	10.0.0.0/16	-	dopt-
-	vpc-00db2eefe23cdafe0	Available	172.31.0.0/16	-	dopt-

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists Endpoints Endpoint services NAT gateways Peering connections Security Network ACLs

Select a VPC above

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Services
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only internet gateways
Carrier gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections

aws Services Search [Alt+S]

### Security Groups (1/5) Info

Name	Security group ID	Security group name	VPC ID	Description
-	sg-09c861c522663df70	launch-wizard-6	vpc-00db2eefe23cdaf0	launch-wizard
<input checked="" type="checkbox"/> pysg	sg-09b19a348372c5a24	terraform-20240306130337389800...	vpc-030d9ef01e54304e0	Managed by Terraform
-	sg-044bb8921f74544eb	launch-wizard-3	vpc-00db2eefe23cdaf0	launch-wizard
-	sg-02786c2d77d989d4c	default	vpc-030d9ef01e54304e0	default VPC
-	sg-07520cf9272b05dfc	default	vpc-00db2eefe23cdaf0	default VPC

aws Services Search [Alt+S]

### Subnets (1/2) Info

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-06e080e9bb0f7507c	Available	vpc-030d9ef01e54304e0   vpc	10.0.1.0/24
<input checked="" type="checkbox"/> python1	subnet-0cc8b5e8e7a506113	Available	vpc-00db2eefe23cdaf0	172.31.0.0/24

aws Services Search [Alt+S]

### Internet gateways (1/2) Info

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-045d5a8b9d96e509d	Attached	vpc-030d9ef01e54304e0   vpc	05826455...
<input checked="" type="checkbox"/>	igw-0d6e98e7a387966d2	Attached	vpc-00db2eefe23cdaf0	05826455...

aws Services Search [Alt+S]

### Route tables (1/3) Info

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0797070512abe5f39	subnet-0cc8b5e8e7a506...	-	Yes	vpc-00db2eefe23cdaf0
<input checked="" type="checkbox"/> routetable	rtb-057d183720a3e9d94	subnet-06e080e9bb0f75...	-	No	vpc-030d9ef01e54304e0
-	rtb-07384a1c91afaa6c6	-	-	Yes	vpc-030d9ef01e54304e0



# FISH WEIGHT

**Species \***

Bream

**Vertical length in cm \***

23.2

**Diagonal length in cm \***

25.4

**Cross length in cm \***

30.0

**Height \***

11.5200

**Width \***

4.0200

**SUBMIT**

---

## METHOD-6

BUILD AND DEPLOY PYTHON APPLICATIONS WITH GIT  
,GITHUB,JENKINS AND TERRAFORM

AWS Console Home > EC2 > Instances > Launch an instance

## Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

**Name and tags Info**

Name: python-6 Add additional tags

**Application and OS Images (Amazon Machine Image) Info**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon | macOS | Ubuntu | Windows | Red Hat | SUSE | ...

**Summary**

Number of instances Info: 1

Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI... read more ami-07761f5ae54c4478d

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance Learn more about free tier AMI usage

**Launch instance** Cancel Review commands

EC2 Dashboard X

EC2 Global View

Events

Console-to-Code Preview

Instances Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Services Search [Alt+S]

## Instances (1) Info

Find Instance by attribute or tag (case-sensitive)

Instance ID = i-02a914d85cf204e85 X Clear filters

<input type="checkbox"/>	Name <small>edit</small>	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
<input type="checkbox"/>	python-6	i-02a914d85cf204e85	<span>Running</span> <small>View details</small> <small>Logs</small>	t2.micro	<span>Initializing</span> <small>View details</small>	<small>View alarms</small> +	us-east-1c	ec2-18-23-

Select an instance

we need to install jenkins

```
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 -y
```

```
sudo yum install -y java-17*
```

```
sudo yum install -y Jenkins
```

```
sudo systemctl start jenkins
```

```
sudo systemctl enable jenkins
```

After we install Jenkins we check status of Jenkins `sudo systemctl status Jenkins`

```
complete!
[ec2-user@ip-172-31-0-211 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-0-211 ~]$ 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-172-31-0-211 ~]$ 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 Wed 2024-03-06 15:10:16 UTC; 2min 57s ago
      Main PID: 6733 (java)
         CGrou.../system.slice/jenkins.service
              └─ 6733 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: 194e4e6646c4435b8f5f5a0e742f8f4f
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: ****
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: ****
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: ****
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: 2024-03-06 15:10:16.705+0000 [id=32]           INFO  jenkins.InitReactorRun...
Mar 06 15:09:46 ip-172-31-0-211.ec2.internal jenkins[6733]: 2024-03-06 15:10:16.734+0000 [id=25]           INFO  hudson.lifecycle.Life...
Mar 06 15:10:16 ip-172-31-0-211.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Mar 06 15:10:16 ip-172-31-0-211.ec2.internal jenkins[6733]: 2024-03-06 15:10:16.839+0000 [id=48]           INFO  h.m.DownloadService$...
Mar 06 15:10:16 ip-172-31-0-211.ec2.internal jenkins[6733]: 2024-03-06 15:10:16.840+0000 [id=48]           INFO  hudson.util.Retrie...
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-0-211 ~]$
```

Getting Started

## Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

`/var/lib/jenkins/secrets/initialAdminPassword`

Please copy the password from either location and paste it below.

Administrator password

.....

Continue

## Getting Started

## Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

**Install suggested plugins**

Install plugins the Jenkins community finds most useful.

**Select plugins to install**

Select and install plugins most suitable for your needs.

Jenkins 2.440.1

## Getting Started

# Getting Started

✓ Folders	✓ OWASP Markup Formatter	✓ Build Timeout	✓ Credentials Binding
✓ Timestamper	✓ Workspace Cleanup	✓ Ant	⌚ Gradle
⌚ Pipeline	⌚ GitHub Branch Source	⌚ Pipeline: GitHub Groovy Libraries	⌚ Pipeline: Stage View
⌚ Git	⌚ SSH Build Agents	⌚ Matrix Authorization Strategy	⌚ PAM Authentication
⌚ LDAP	⌚ Email Extension	⌚ Mailer	⌚ Dark Theme

```
Timestamper
** Caffeine API
** Script Security
** JavaBeans Activation Framework (JAF) API
** JAXB
** SnakeYAML API
** Jackson 2 API
** commons-text API
** Pipeline: Supporting APIs
** Plugin Utilities API
** Font Awesome API
** Bootstrap 5 API
** JQuery3 API
** ECharts API
** Display URL API
** Checks API
** JUnit
** Matrix Project
** Resource Disposer
Workspace Cleanup
Ant
** JavaMail API
** Durable Task
** - required dependency
```

## Getting Started

# Jenkins is ready!

Your Jenkins setup is complete.

[Start using Jenkins](#)

Not secure 54.242.49.131:8080

## Getting Started

# Create First Admin User

Username

rama

Password

\*\*\*\*\*

Confirm password

\*\*\*\*\*

Full name

ramakrishna

E-mail address

shivaram3028@gmail.com

Skin and continue as admin

Save and Continue

Not secure 54.242.49.131:8080

## Getting Started

# Instance Configuration

Jenkins URL:

http://54.242.49.131:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.440.1

Not now

Save and Finish

Enter an item name

**Required field**

**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 folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

**OK** **Search Pipeline**  
Create a new Pipeline project according to detected branches in your SCM repository.

Dashboard > Add description

- + New Item
- People
- Build History
- Manage Jenkins
- My Views

**Welcome to Jenkins!**

This page is where your Jenkins jobs will be displayed. To get started, you can set up distributed builds or start building a software project.

**Start building your software project**

**Create a job**

**Set up a distributed build**

**Set up an agent**

**Configure a cloud**

**Learn more about distributed builds**

**Jenkins** Search (CTRL+K)

Dashboard > clone-job > Configuration

### Configure

- General
- Source Code Management
- Build Triggers
- Build Environment
- Build Steps
- Post-build Actions

### General

Description

for cloning-job

Plain text [Preview](#)

Discard old builds

GitHub project

This project is parameterized

Throttle builds

Execute concurrent builds if necessary

**Advanced**

**Save** **Apply**

## Configure

- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

## Build Steps

### Execute shell

#### Command

See [the list of available environment variables](#)

```
sudo yum -y install git  
git clone https://github.com/rama861/python-terraform.git
```

Save

Apply

## Enter an item name

build-job

» Required field



### 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 for other Jenkins jobs. A folder creates a single URL for all its children.



Jenkins

Dashboard > build-job > Configuration

OK

Cancel

Creates a

separate n

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

o

</

```
sudo yum install -y yum-utils
```

```
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
```

```
sudo yum -y install terraform
```

```
cd /var/lib/jenkins/workspace/clone-job/terra
```

```
terraform init
```

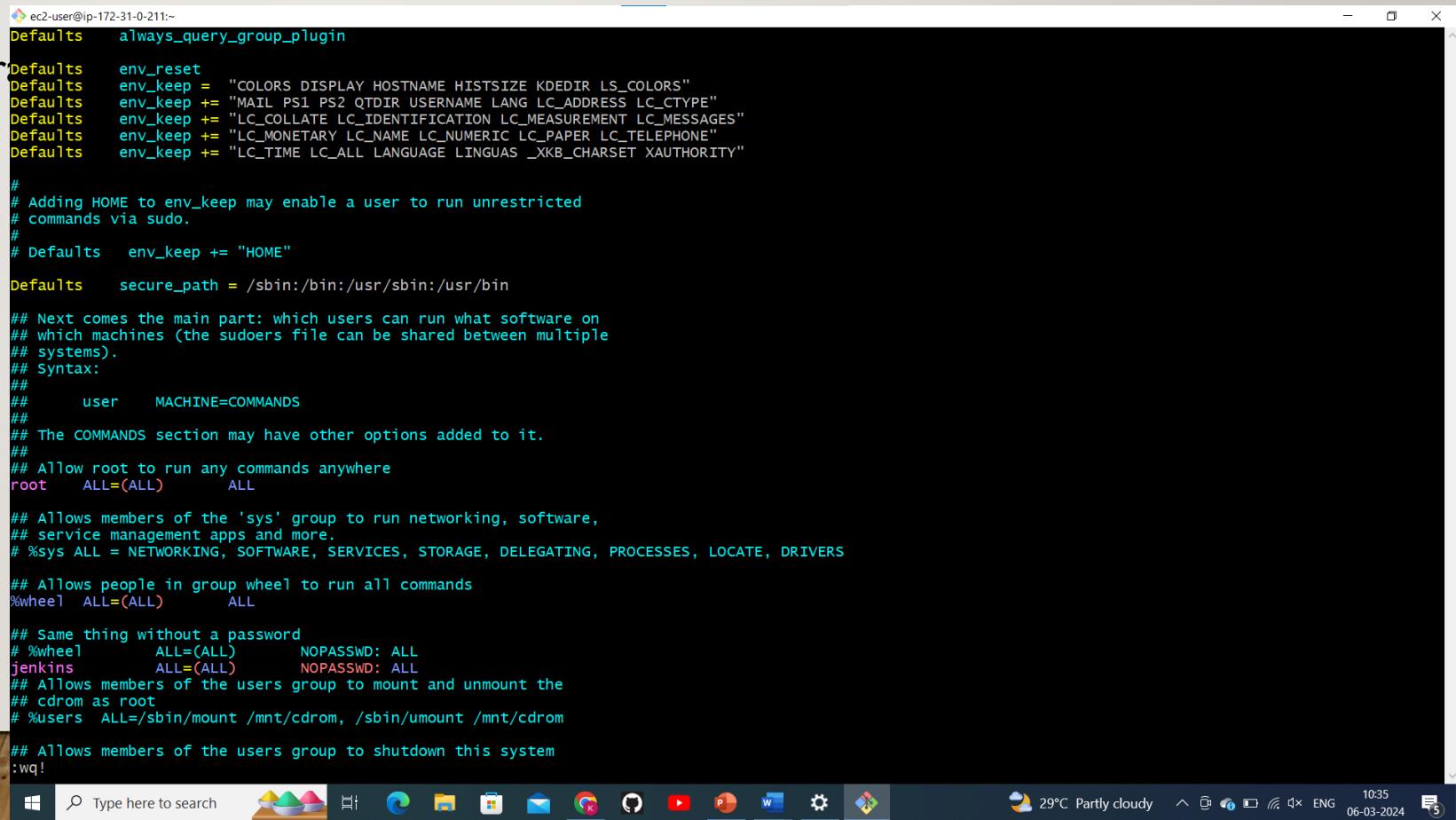
```
terraform fmt
```

```
terraform plan
```

```
terraform apply
```

```
terraform --auto-approve
```

## Sudo visudo



```
ec2-user@ip-172-31-0-211:~
```

```
Defaults always_query_group_plugin
Defaults env_reset
Defaults env_keep = "COLORS DISPLAY HOSTNAME HISTSIZE KDEDIR LS_COLORS"
Defaults env_keep += "MAIL PS1 PS2 QTDIR USERNAME LANG LC_ADDRESS LC_CTYPE"
Defaults env_keep += "LC_COLLATE LC_IDENTIFICATION LC_MEASUREMENT LC_MESSAGES"
Defaults env_keep += "LC_MONETARY LC_NAME LC_NUMERIC LC_PAPER LC_TELEPHONE"
Defaults env_keep += "LC_TIME LC_ALL LANGUAGE LINGUAS _XKB_CHARSET XAUTHORITY"

#
# 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 unmount the
## cdrom as root
# %users   ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

## Allows members of the users group to shutdown this system
:wq!
```

The screenshot shows a Windows desktop environment. A terminal window is open, displaying the contents of the /etc/sudoers file. The file contains standard sudo configuration, including entries for root, the wheel group, and the jenkins user. The terminal window has a dark background with white text. Below the terminal is a taskbar with several pinned icons, including File Explorer, Microsoft Edge, Mail, and others. At the bottom right of the screen, there is a system tray with icons for battery status, network, volume, and system notifications. The date and time (06-03-2024, 10:35) are also visible in the bottom right corner.



Dashboard > clone-job > #1 > Console Output

Status

</> Changes

Console Output

View as plain text

Edit Build Information

Delete build '#1'

Git Build Data

## Console Output

Started by user ramakrishna

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/clone-job

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository <https://github.com/rama861/python-terraform.git>

> git init /var/lib/jenkins/workspace/clone-job # timeout=10

Fetching upstream changes from <https://github.com/rama861/python-terraform.git>

> git --version # timeout=10

> git --version # 'git version 2.40.1'

Dashboard > build-job > Configuration

## Configure

### Source Code Management

None

Git

### General

### Source Code Management

### Build Triggers

### Build Environment

### Build Steps

### Post-build Actions

### Build Triggers

Trigger builds remotely (e.g., from scripts) ?

Build after other projects are built ?

#### Projects to watch

clone-job.

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Always trigger, even if the build is aborted

The screenshot shows two side-by-side views of a Jenkins job configuration page. The top view is a wider shot, and the bottom view is a zoomed-in detail of the 'Build Steps' section.

**Top View (Full Configuration Page):**

- Header:** Not secure 54.242.49.131:8080/job/build-job/configure
- Breadcrumbs:** Dashboard > build-job > Configuration
- Title:** Configure
- Left Sidebar:** General, Source Code Management (selected), Build Triggers, Build Environment, Build Steps (selected), Post-build Actions.
- Right Panel:**
  - Source Code Management:** None (radio button selected), Git (radio button unselected).
  - Build Triggers:** Trigger builds remotely (e.g., from scripts) (checkbox unselected), Build after other projects are built (checkbox selected).
  - Projects to watch:** clone-job,
  - Trigger options:** Trigger only if build is stable (radio button selected), Trigger even if the build is unstable (radio button unselected), Trigger even if the build fails (radio button unselected), Always trigger, even if the build is aborted (radio button unselected).

**Bottom View (Build Steps Detail):**

- Header:** Not secure 54.242.49.131:8080/job/build-job/configure
- Breadcrumbs:** Dashboard > build-job > Configuration
- Title:** Configure
- Left Sidebar:** General, Source Code Management, Build Triggers, Build Environment, Build Steps (selected), Post-build Actions.
- Content Area:**
  - Build Steps:** Execute shell
  - Command:** See [the list of available environment variables](#)
  - ```
sudo yum install -y yum-utils
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
sudo yum -y install terraform
cd /var/lib/jenkins/workspace/clone-job/python-terraform
terraform init
terraform validate
terraform apply --auto-approve
```

Copy the public ip of newly created instance  
and browse it in the browser along with port number



The image shows a close-up of a betta fish with a large, flowing, multi-layered tail and dorsal fin. The fins are primarily blue with distinct red and white stripes. The body of the fish is a pale yellow or cream color. It is positioned on the left side of a web browser window.

Not secure 3.219.231.152:8000

# FISH WEIGHT

**Species \***  
Bream

**Vertical length in cm \***  
23.2

**Diagonal length in cm \***  
25.4

**Cross length in cm \***  
30.0

**Height \***  
11.5200

**Width \***  
4.0200

**SUBMIT**

# We can see here the resources created

**VPC dashboard**

**Your VPCs (1/3) Info**

| Name                 | VPC ID                       | State            | IPv4 CIDR          | IPv6 CIDR | DHC         |
|----------------------|------------------------------|------------------|--------------------|-----------|-------------|
| vpc-00db2eef23cdafe0 | Available                    | 172.31.0.0/16    | -                  | -         | dopt        |
| <b>python</b>        | <b>vpc-0010f90409c8c228c</b> | <b>Available</b> | <b>10.0.0.0/16</b> | <b>-</b>  | <b>dopt</b> |
| vpc                  | vpc-0c6d2b6d91fb51fd5        | Available        | 10.0.0.0/16        | -         | dopt        |

**vpc-0010f90409c8c228c / vpc**

**Details**

|                                 |                                                       |                                           |                                           |
|---------------------------------|-------------------------------------------------------|-------------------------------------------|-------------------------------------------|
| VPC ID<br>vpc-0010f90409c8c228c | State<br><span style="color: green;">Available</span> | DNS hostnames<br>Disabled                 | DNS resolution<br>Enabled                 |
| Tenancy<br>Default              | DHCP option set<br>dopt-04622e84dafddd19              | Main route table<br>rtb-03ac78a677a1d3693 | Main network ACL<br>acl-08543399b05e73b22 |

**EC2 Dashboard**

**Instances (1/2) Info**

| Name          | Instance ID         | Instance state                             | Instance type | Status check                                         | Alarm status                                     | Availability Zone | Public IPv4 |
|---------------|---------------------|--------------------------------------------|---------------|------------------------------------------------------|--------------------------------------------------|-------------------|-------------|
| <b>python</b> | i-038f6bfd23ae52e4b | <span style="color: green;">Running</span> | t2.micro      | <span style="color: green;">2/2 checks passed</span> | <span style="color: green;">View alarms +</span> | us-east-1a        | -           |
| python-6      | i-04f3cc9f22f85e907 | <span style="color: green;">Running</span> | t2.micro      | <span style="color: green;">2/2 checks passed</span> | <span style="color: green;">View alarms +</span> | us-east-1c        | ec2-54-242- |

**Instance: i-038f6bfd23ae52e4b (python)**

**Details**

|                                                     |                                                                     |                                     |
|-----------------------------------------------------|---------------------------------------------------------------------|-------------------------------------|
| Instance ID<br>i-038f6bfd23ae52e4b (python)         | Public IPv4 address<br>3.219.231.152 <a href="#">[open address]</a> | Private IPv4 addresses<br>10.0.1.68 |
| IPv6 address<br>-                                   | Instance state<br><span style="color: green;">Running</span>        | Public IPv4 DNS<br>-                |
| Hostname type<br>IP name: ip-10-0-1-68.ec2.internal | Private IP DNS name (IPv4 only)<br>ip-10-0-1-68.ec2.internal        |                                     |

**VPC dashboard**

**Subnets (3) Info**

| Name    | Subnet ID                | State                                        | VPC                   | IPv4 CIDR     |
|---------|--------------------------|----------------------------------------------|-----------------------|---------------|
| subnet  | subnet-0544ab79cd892b58  | <span style="color: green;">Available</span> | vpc-0010f90409c8c228c | 10.0.1.0/24   |
| subnet  | subnet-08c174330026e25dd | <span style="color: green;">Available</span> | vpc-0c6d2b6d91fb51fd5 | 10.0.1.0/24   |
| python1 | subnet-0cc8b5e8e7a506113 | <span style="color: green;">Available</span> | vpc-00db2eef23cdafe0  | 172.31.0.0/24 |

AWS CloudWatch Metrics

aws Services Search [Alt+S]

VPC dashboard EC2 Global View Filter by VPC: Select a VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways DHCP option sets Elastic IPs Managed prefix lists

Internet gateways (3) Info

| Name | Internet gateway ID   | State    | VPC ID                      | Owner        |
|------|-----------------------|----------|-----------------------------|--------------|
| -    | igw-070d7bb3df5cdf084 | Attached | vpc-0c6d2b6d91fb51fd5   vpc | 058264554922 |
| -    | igw-0868fa27f5f1d2b77 | Attached | vpc-0010f90409c8c228c   vpc | 058264554922 |
| -    | igw-0d6e98e7a387966d2 | Attached | vpc-00db2eefe23cdaf0        | 058264554922 |

Select an internet gateway above

Actions Create internet gateway

N. Virginia Cheran

This screenshot shows the AWS CloudWatch Metrics interface for the VPC service. The left sidebar navigation includes 'VPC dashboard', 'EC2 Global View', 'Filter by VPC' (set to 'Select a VPC'), and a 'Virtual private cloud' section with links for 'Your VPCs', 'Subnets', 'Route tables', 'Internet gateways' (which is the current page), 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', and 'Managed prefix lists'. The main content area is titled 'Internet gateways (3) Info' and displays a table with three rows. The columns are 'Name' (empty), 'Internet gateway ID' (igw-070d7bb3df5cdf084, igw-0868fa27f5f1d2b77, igw-0d6e98e7a387966d2), 'State' (Attached), 'VPC ID' (vpc-0c6d2b6d91fb51fd5 | vpc, vpc-0010f90409c8c228c | vpc, vpc-00db2eefe23cdaf0), and 'Owner' (058264554922, 058264554922, 058264554922). Below the table is a message 'Select an internet gateway above'. The top navigation bar includes the AWS logo, 'Services', a search bar, and account information for 'N. Virginia' and 'Cheran'.

AWS CloudWatch Metrics

aws Services Search [Alt+S]

VPC dashboard EC2 Global View Filter by VPC: Select a VPC

Virtual private cloud Your VPCs Subnets Route tables Internet gateways Egress-only internet gateways Carrier gateways

Route tables (5) Info

| Name       | Route table ID        | Explicit subnet associ... | Edge associations | Main | VPC                   |
|------------|-----------------------|---------------------------|-------------------|------|-----------------------|
| -          | rtb-032987b571d3a5014 | -                         | -                 | Yes  | vpc-0c6d2b6d91fb51fd  |
| routetable | rtb-0ab6646d2594e5aa1 | subnet-0544ab79cdc892...  | -                 | No   | vpc-0010f90409c8c228  |
| -          | rtb-03ac78a677a1d3693 | -                         | -                 | Yes  | vpc-0010f90409c8c228  |
| routetable | rtb-0f938837679df6dc0 | subnet-08c174330026e2...  | -                 | No   | vpc-0c6d2b6d91fb51fd  |
| -          | rtb-0797070512abe5f39 | subnet-0cc8b5e8e7a506...  | -                 | Yes  | vpc-00db2eefe23cdafec |

Actions Create route table

N. Virginia Cheran

This screenshot shows the AWS CloudWatch Metrics interface for the VPC service. The left sidebar navigation includes 'VPC dashboard', 'EC2 Global View', 'Filter by VPC' (set to 'Select a VPC'), and a 'Virtual private cloud' section with links for 'Your VPCs', 'Subnets', 'Route tables' (which is the current page), 'Internet gateways', 'Egress-only internet gateways', and 'Carrier gateways'. The main content area is titled 'Route tables (5) Info' and displays a table with five rows. The columns are 'Name' (empty, routetable, empty, routetable, empty), 'Route table ID' (rtb-032987b571d3a5014, rtb-0ab6646d2594e5aa1, rtb-03ac78a677a1d3693, rtb-0f938837679df6dc0, rtb-0797070512abe5f39), 'Explicit subnet associ...' (empty, subnet-0544ab79cdc892..., empty, subnet-08c174330026e2..., empty, subnet-0cc8b5e8e7a506...), 'Edge associations' (empty, -, empty, -, empty, -), 'Main' (Yes, No, Yes, No, Yes), and 'VPC' (vpc-0c6d2b6d91fb51fd, vpc-0010f90409c8c228, vpc-0010f90409c8c228, vpc-0c6d2b6d91fb51fd, vpc-00db2eefe23cdafec). The top navigation bar includes the AWS logo, 'Services', a search bar, and account information for 'N. Virginia' and 'Cheran'.

## METHOD-9

BUILD AND DEPLOY PYTHON APPLICATIONS  
WITH GIT,GITHUB,JENKINS AND TERRAFORM  
(WITH BUILD PERIODICALLY POLLSCM AAND  
WEBHOOKS)

Launch an instance with name and select AMI version

Select an keypair existing or newly created keypair

Next give the vpc,subnets and security groups with port numbers

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. The 'Summary' step is displayed, showing the following configuration:

- Name and tags:** Name is set to "python-9".
- Software Image (AMI):** Amazon Linux 2023 AMI 2023.3.2... (ami-0f403e3180720dd7e)
- Virtual server type (instance type):** t2.micro
- Firewall (security group):** New security group
- Storage (volumes):** 1 volume(s) - 8 GiB

A tooltip for the t2.micro instance type indicates it is "Free tier eligible". At the bottom, there are "Cancel", "Launch instance", and "Review commands" buttons.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console, displaying the 'Instance type' and 'Key pair (login)' steps.

**Instance type:** t2.micro (Free tier eligible)  
Family: t2 - 1 vCPU - 1 GiB Memory - Current generation: true  
On-Demand Windows base pricing: 0.0162 USD per Hour  
On-Demand SUSE base pricing: 0.0116 USD per Hour  
On-Demand RHEL base pricing: 0.0116 USD per Hour  
On-Demand Linux base pricing: 0.0116 USD per Hour

**Key pair (login):** Key pair name is required. A dropdown menu shows "Select" and a "Create new key pair" button.

**Network settings:** Network (Info) - vpc-09db20ef637cd4f50

At the bottom, there are "Cancel", "Launch instance", and "Review commands" buttons.

Next we need to install Jenkins and java  
Start the Jenkins and enable jenkins

```
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 -y
sudo yum install -y java-17*
sudo yum install -y Jenkins
sudo systemctl start Jenkins
sudo systemctl enable jenkins
```

```
cve-2021-44228-hotpatch.service.
Installing : 1:java-17-amazon-corretto-headless-17.0.10+8-1.amzn2.1.x86_64 27/3
Installing : 1:java-17-amazon-corretto-devel-17.0.10+8-1.amzn2.1.x86_64 28/3
Installing : 1:java-17-amazon-corretto-jmod-17.0.10+8-1.amzn2.1.x86_64 29/3
Installing : 1:java-17-amazon-corretto-17.0.10+8-1.amzn2.1.x86_64 30/3
Installing : 1:java-17-amazon-corretto-javadoc-17.0.10+8-1.amzn2.1.x86_64 31/3
Verifying : 1:java-17-amazon-corretto-jmod-17.0.10+8-1.amzn2.1.x86_64 1/3
Verifying : 1:java-17-amazon-corretto-devel-17.0.10+8-1.amzn2.1.x86_64 2/3
Verifying : dejavu-serif-fonts-2.33-6.amzn2.noarch 3/3
Verifying : 1:java-17-amazon-corretto-17.0.10+8-1.amzn2.1.x86_64 4/3
Verifying : fontpackages-filesystem-1.44-8.amzn2.noarch 5/3
Verifying : libxcb-1.12-1.amzn2.0.2.x86_64 6/3
Verifying : libxrandr-1.5.1-2.amzn2.0.3.x86_64 7/3
Verifying : libxext-1.3.3-3.amzn2.0.2.x86_64 8/3
Verifying : libxi1-1.6.7-3.amzn2.0.5.x86_64 9/3
Verifying : log4j-cve-2021-44228-hotpatch-1.3-7.amzn2.noarch 10/3
Verifying : libxi-1.7.9-1.amzn2.0.2.x86_64 11/3
Verifying : dejavu-fonts-common-2.33-6.amzn2.noarch 12/3
Verifying : 1:java-17-amazon-corretto-javadoc-17.0.10+8-1.amzn2.1.x86_64 13/3
Verifying : libxau-1.0.8-2.1.amzn2.0.2.x86_64 14/3
Verifying : libSM-1.2.2-2.amzn2.0.2.x86_64 15/3
Verifying : python-lxml-3.2.1-4.amzn2.0.5.x86_64 16/3
Verifying : libXrender-0.9.10-1.amzn2.0.2.x86_64 17/3
Verifying : libxi1-common-1.6.7-3.amzn2.0.5.noarch 18/3
Verifying : dejavu-sans-fonts-2.33-6.amzn2.noarch 19/3
Verifying : fontconfig-2.13.0-4.3.amzn2.x86_64 20/3
Verifying : libxt-1.1.5-3.amzn2.0.2.x86_64 21/3
Verifying : giflib-4.1.6-9.amzn2.0.2.x86_64 22/3
Verifying : libxinerama-1.1.3-2.1.amzn2.0.2.x86_64 23/3
Verifying : dejavu-sans-mono-fonts-2.33-6.amzn2.noarch 24/3
Verifying : libxsllt-1.1.28-6.amzn2.x86_64 25/3
Verifying : python-javapackages-3.4.1-11.amzn2.noarch 26/3
Verifying : libxtst-1.2.3-1.amzn2.0.2.x86_64 27/3
Verifying : alsalib-1.1.4.1-2.amzn2.x86_64 28/3
Verifying : 1:java-17-amazon-corretto-headless-17.0.10+8-1.amzn2.1.x86_64 29/3
Verifying : libICE-1.0.9-9.amzn2.0.2.x86_64 30/3
Verifying : javapackages-tools-3.4.1-11.amzn2.noarch 31/3

nstalled:
java-17-amazon-corretto.x86_64 1:17.0.10+8-1.amzn2.1 java-17-amazon-corretto-devel.x86_64 1:17.0.10+8-1.amzn2.1
java-17-amazon-corretto-headless.x86_64 1:17.0.10+8-1.amzn2.1 java-17-amazon-corretto-javadoc.x86_64 1:17.0.10+8-1.amzn2.1
java-17-amazon-corretto-jmod.x86_64 1:17.0.10+8-1.amzn2.1
```

```
dependency Installed:  
alsa-lib.x86_64 0:1.4.1-2.amzn2  
dejavu-sans-fonts.noarch 0:2.33-6.amzn2  
dejavu-sans-fonts.noarch 0:2.33-6.amzn2  
Fontpackages-filesystem.noarch 0:1.44-8.amzn2  
javapackages-tools.noarch 0:3.4.11-amzn2  
libSM.x86_64 0:1.2.2-2.amzn2.0.2  
libX11-common.noarch 0:1.6.7-3.amzn2.0.5  
libXext.x86_64 0:1.3.3-3.amzn2.0.2  
  
dependency Required-By:  
dejavu-fonts-common.noarch 0:2.33-6.amzn2  
dejavu-sans-mono-fonts.noarch 0:2.33-6.amzn2  
fontconfig.x86_64 0:2.13.0-4.amzn2  
giffib.x86_64 0:1.4.1-6-9.amzn2.0.2  
libICE.x86_64 0:1.0.9-9.amzn2.0.2  
libX11.x86_64 0:1.6.7-3.amzn2.0.5  
libXau.x86_64 0:1.0.8-2-1.amzn2.0.2  
libXi.x86_64 0:1.7.9-1.amzn2.0.2
```

Next we check the status of Jenkins

Here we can see active running of status Jenkins

```
Complete!
[ec2-user@ip-172-31-0-206 ~]$ sudo systemctl start jenkins

[ec2-user@ip-172-31-0-206 ~]$ 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-172-31-0-206 ~]$ sudo systemctl enable jenkins
[ec2-user@ip-172-31-0-206 ~]$ 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 Fri 2024-03-08 04:20:08 UTC; 23s ago
     Main PID: 6741 (java)
       CGroup: /system.slice/jenkins.service
               └─6741 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPo...

Mar 08 04:19:35 ip-172-31-0-206.ec2.internal jenkins[6741]: 745bae20f73b4664a3f1f32ee677b0ce
Mar 08 04:19:35 ip-172-31-0-206.ec2.internal jenkins[6741]: This may also be found at: /var/lib/jenkins/secrets/initial...word
Mar 08 04:19:35 ip-172-31-0-206.ec2.internal jenkins[6741]: ****
Mar 08 04:19:35 ip-172-31-0-206.ec2.internal jenkins[6741]: ****
Mar 08 04:19:35 ip-172-31-0-206.ec2.internal jenkins[6741]: ****
Mar 08 04:20:07 ip-172-31-0-206.ec2.internal jenkins[6741]: 2024-03-08 04:20:07.994+0000 [id=33]           INFO      jen...tion
Mar 08 04:20:08 ip-172-31-0-206.ec2.internal jenkins[6741]: 2024-03-08 04:20:08.020+0000 [id=25]           INFO      hud...ning
Mar 08 04:20:08 ip-172-31-0-206.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Mar 08 04:20:08 ip-172-31-0-206.ec2.internal jenkins[6741]: 2024-03-08 04:20:08.161+0000 [id=48]           INFO      h.m...ller
Mar 08 04:20:08 ip-172-31-0-206.ec2.internal jenkins[6741]: 2024-03-08 04:20:08.162+0000 [id=48]           INFO      hud...t #1
```

The image shows three screenshots of the Jenkins Setup Wizard interface, illustrating the progression from initial setup to a fully functional Jenkins instance.

**Unlock Jenkins**

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/var/lib/jenkins/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

**Administrator password**  
.....

**Customize Jenkins**

Plugins extend Jenkins with additional features to support many different needs.

**Install suggested plugins**  
Install plugins the Jenkins community finds most useful

**Select plugins to install**  
Select and install plugins most suitable for your needs

**Getting Started**

Jenkins 2.440.1

(3) WhatsApp PavanGitPro/terraform GitHub Instance details | EC2 Linux ChatGPT Setup Wizard [Jenkin

Not secure 54.163.27.133:8080

## Getting Started

### Create First Admin User

Username: krishna

Password: .....  
Confirm password: .....

Full name: ramakrishna

E-mail address: shivaram3028@gmail.com

Skip and continue as admin Save and Continue

## Getting Started

# Instance Configuration

Jenkins URL: http://54.163.27.133:8080/

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Jenkins 2.440.1 Not now Save and Finish

Not secure 54.163.27.133:8080

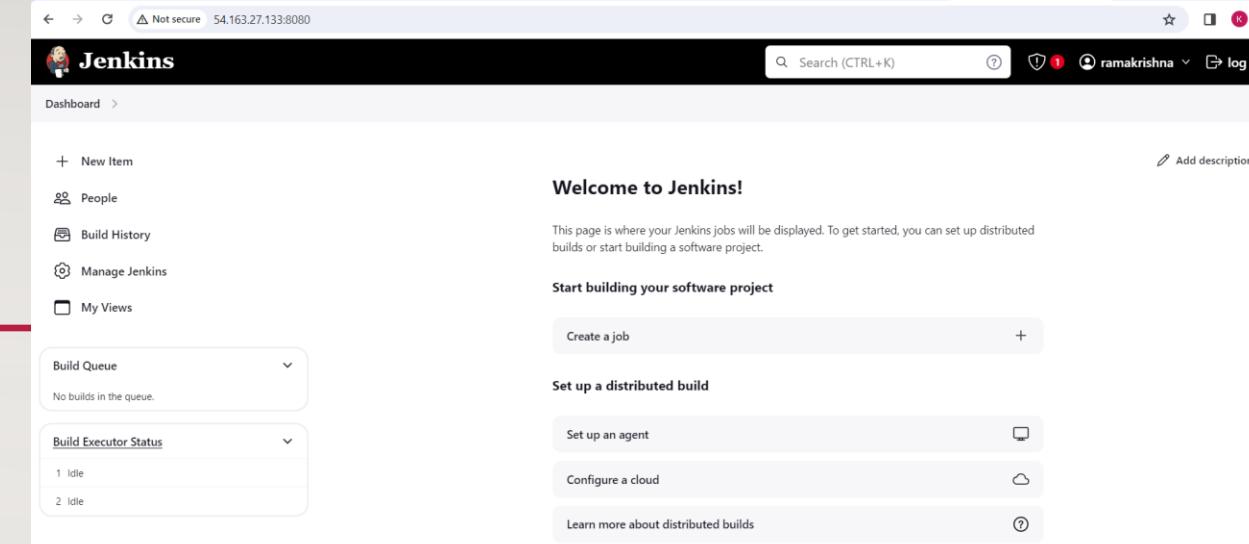
## Getting Started

# Jenkins is ready!

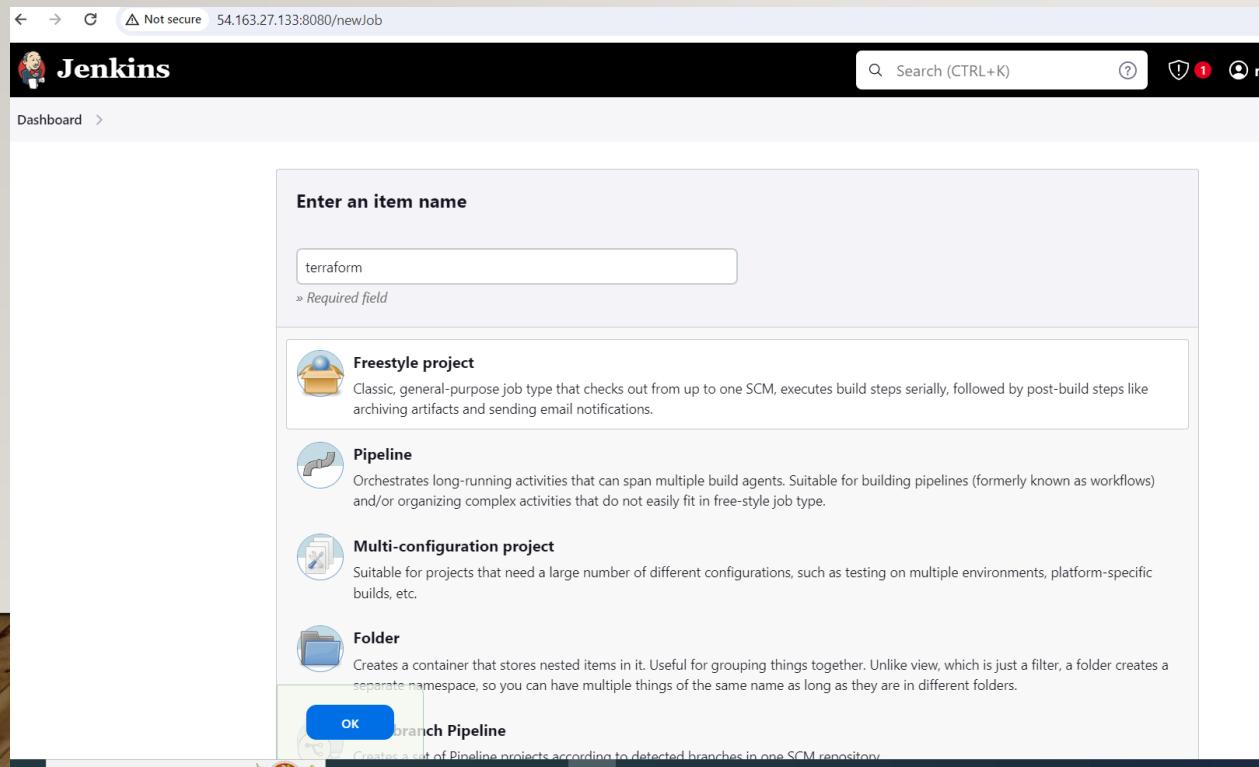
Your Jenkins setup is complete.

Start using Jenkins

After launching Jenkins create a job  
Give an item name and select project which we will work on it



The screenshot shows the Jenkins dashboard at the URL <http://54.163.27.133:8080>. The title bar says "Not secure". The main header is "Jenkins". The left sidebar includes links for "New Item", "People", "Build History", "Manage Jenkins", and "My Views". The right side features a "Welcome to Jenkins!" message, a "Start building your software project" section with a "Create a job" button, and a "Set up a distributed build" section with links for "Set up an agent", "Configure a cloud", and "Learn more about distributed builds".



The screenshot shows the "newJob" creation dialog box at the URL <http://54.163.27.133:8080/newJob>. The title bar says "Not secure". The main header is "Jenkins". The search bar contains "Search (CTRL+K)". The user has entered "terraform" into the "Enter an item name" input field. Below the input field, there is an error message: "» Required field". The dialog lists four project types: "Freestyle project", "Pipeline", "Multi-configuration project", and "Folder". Each type has a description and a corresponding icon. At the bottom of the dialog, there is a blue "OK" button and a link "Branch Pipeline".

# BUILD PERIODICALLY

- In the source code management select an GIT and give git repository URL
- In build periodical section give 5\* for build it every minute it builds

The screenshot shows a Jenkins configuration page for a job named 'terraform'. The top navigation bar indicates the URL is 'Not secure 54.163.27.133:8080/job/terraform/configure'. The page title is 'Configuration'. On the left, there's a sidebar with options: General, Source Code Management (which is selected), Build Triggers, Build Environment, Build Steps, and Post-build Actions. The main content area is titled 'Configure' and shows the 'Source Code Management' section. It has two radio button options: 'None' and 'Git' (which is selected). Below this, there's a 'Repositories' section with a 'Repository URL' input field containing 'https://github.com/rama861/python-terraform.git'. There's also a 'Credentials' section with a dropdown menu showing '- none -' and a '+ Add' button. A 'Branches to build' section at the bottom has a 'Branch Specifier (blank for 'any')' input field. At the bottom right, there are 'Save' and 'Apply' buttons.

Not secure 54.163.27.133:8080/job/terraform/configure

Dashboard > terraform > Configuration

## Configure

### Build Triggers

Trigger builds remotely (e.g., from scripts) ?  
 Build after other projects are built ?  
 Build periodically ?

Schedule ?  
\*/5 \* \* \*

**⚠️** Do you really mean "every minute" when you say "\*/5 \* \* \*"? Perhaps you meant "H \*/5 \* \* \*" to poll once per hour  
Would last have run at Thursday, March 7, 2024 at 10:59:23 AM Coordinated Universal Time; would next run at Thursday, March 7, 2024 at 3:00:23 PM Coordinated Universal Time.

GitHub hook trigger for GITScm polling ?  
 Poll SCM ?

### Build Environment

Delete workspace before build starts

**Save** **Apply**

### Build Steps

Execute shell ?

Command

See the list of available environment variables

```
sudo yum install -y yum-utils  
sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo  
sudo yum -y install terraform  
terraform init  
terraform fmt  
terraform validate  
terraform apply -auto-approve
```

Advanced ▾

**Save** **Apply**



Status

Changes

Console Output

View as plain text

Edit Build Information

Delete build '#7'

Git Build Data

← Previous Build

## Console Output

```
Started by user ramakrishna
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/terraform
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/terraform/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/rama861/python-terraform.git # timeout=10
Fetching upstream changes from https://github.com/rama861/python-terraform.git
> git --version # timeout=10
> git --version # 'git version 2.40.1'
> git fetch --tags --force --progress -- https://github.com/rama861/python-terraform.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/main^{commit} # timeout=10
Checking out Revision 38ddc4ee0455ff24dd09a3761e19021cc4f11da (refs/remotes/origin/main)
> git config core.sparsecheckout # timeout=10
> git checkout -f 38ddc4ee0455ff24dd09a3761e19021cc4f11da # timeout=10
Commit message: "Update instance.tf"
> git rev-list --no-walk 38ddc4ee0455ff24dd09a3761e19021cc4f11da # timeout=10
[terraform] $ /bin/sh -xe /tmp/jenkins1017690740021573058.sh
+ sudo yum install -y yum-utils
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Package yum-utils-1.1.31-46.amzn2.0.1.noarch already installed and latest version
Nothing to do
+ sudo yum-config-manager --add-repo https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

Here we can see console output success and it gives the public ip copy the public ip and browse it in the browser along with port number what you have given

```
instance.tf
routetable.tf
+ terraform validate
@[32m@[1mSuccess!@0m The configuration is valid.
@[0m
+ terraform apply --auto-approve
@[0m@[1maws_vpc.demovpc: Refreshing state... [id=vpc-06ab441b09fb5f8a]@[0m
@[0m@[1maws_subnet.publicsubnet1: Refreshing state... [id=subnet-0d1c323539af34da4]@[0m
@[0m@[1maws_security_group.sg1: Refreshing state... [id=sg-09e70f5d0c1b40182]@[0m
@[0m@[1maws_internet_gateway.igpy: Refreshing state... [id=igw-021d0b94090495836]@[0m
@[0m@[1maws_instance.python: Refreshing state... [id=i-0f4bccb25806c5467]@[0m
@[0m@[1maws_route_table.route1: Refreshing state... [id=rtb-09301bc8323402513]@[0m
@[0m@[1maws_route_table_association.route1: Refreshing state... [id=rtbassoc-012440af87f66ed9b]@[0m

@[0m@[1m@[32mNo changes.@[0m@[1m Your infrastructure matches the configuration.@[0m
@[0mTerraform has compared your real infrastructure against your configuration
and found no differences, so no changes are needed.
@[0m@[1m@[32m
Apply complete! Resources: 0 added, 0 changed, 0 destroyed.
@[0m@[0m@[1m@[32m
Outputs:

@[0mpublic_ip_address_1 = "3.234.183.236"
Finished: SUCCESS
```



# FISH WEIGHT

**Species \***

Bream

**Vertical length in cm \***

23.2

**Diagonal length in cm \***

25.4

**Cross length in cm \***

30.0

**Height \***

11.5200

**Width \***

4.0200

**SUBMIT**

# Here the resources will be created

The screenshot displays five tabs from the AWS VPC console:

- EC2 Instances (1/2) Info**: Shows two instances: "python" (Running, t2.micro, 2/2 checks passed) and "python-jk" (Running, t2.micro, 2/2 checks passed).

| Name      | Instance ID         | Instance state | Instance type | Status check      | Alarm status                | Availability Zone |
|-----------|---------------------|----------------|---------------|-------------------|-----------------------------|-------------------|
| python    | i-0f4bccb25806c5467 | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1a        |
| python-jk | i-0c2d5ba54c87fd82a | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1c        |
- Your VPCs (1/2) Info**: Shows one VPC named "vpc".

| Name | VPC ID                | State     | IPv4 CIDR   | IPv6 CIDR |
|------|-----------------------|-----------|-------------|-----------|
| vpc  | vpc-06ab441b09bfb5f8a | Available | 10.0.0.0/16 | -         |
- Subnets (1/2) Info**: Shows two subnets: "subnet" and "python1".

| Name    | Subnet ID                | State     | VPC                         | IPv4 CIDR     |
|---------|--------------------------|-----------|-----------------------------|---------------|
| subnet  | subnet-0d1c323539af34da4 | Available | vpc-06ab441b09bfb5f8a   vpc | 10.0.1.0/24   |
| python1 | subnet-0cc8b5e8e7a506113 | Available | vpc-00db2eefe23cdaf0        | 172.31.0.0/24 |
- Route tables (1/3) Info**: Shows three route tables: "rtb-02d650d913e8de2df", "rtb-09301bc8323402513", and "rtb-0797070512abe5f39".

| Name       | Route table ID        | Explicit subnet assoc... | Edge associations | Main |
|------------|-----------------------|--------------------------|-------------------|------|
| -          | rtb-02d650d913e8de2df | -                        | -                 | Yes  |
| routetable | rtb-09301bc8323402513 | subnet-0d1c323539af34... | -                 | No   |
| -          | rtb-0797070512abe5f39 | subnet-0cc8b5e8e7a506... | -                 | Yes  |
- Internet Gateways**: Shows two internet gateways.

| Name | Internet gateway ID   | Status   | Attachment            | ABID         | Owner |
|------|-----------------------|----------|-----------------------|--------------|-------|
| -    | igw-00db2eefe23cdaf0  | Attached | vpc-06ab441b09bfb5f8a | 0282e4224655 | -     |
| -    | igw-00d1c323539af34da | Attached | vpc-0d1c323539af34da  | 0282e4224655 | -     |

# POLL SCM

In build trigger section select an poll scm and scheduled it for five minutes

The screenshot shows the Jenkins configuration page for a 'terraform' job. The left sidebar has 'Configure' selected. Under 'Source Code Management', 'Git' is chosen, and the 'Repository URL' is set to 'https://github.com/rama861/python-agri.git'. The 'Build Triggers' section is collapsed.

The screenshot shows the Jenkins configuration page for a 'build-job' job. The left sidebar has 'Configure' selected. Under 'Build Triggers', 'Poll SCM' is checked, and the schedule is set to 'H \* \* \* \*'. A warning message is displayed: '⚠ Do you really mean "every minute" when you say "H \* \* \* \*"? Perhaps you meant "H \* \* \* \*" to poll once every minute.' The 'Build Environment' section is partially visible at the bottom.

Next you may have some changes in github create some text files automatically it builds after changes

The screenshot shows a GitHub repository named "python-terraform". The commit history lists several changes made by user "rama861" to files like "data.sh", "dumy.txt", "igw.tf", etc. The commit message for "dumy.txt" is "Create dumy.txt". The URL for the commit is <https://github.com/rama861/python-terraform/blob/main/dumy.txt>.

| Name           | Last commit message  |
|----------------|----------------------|
| data.sh        | Update data.sh       |
| dumy.txt       | Create dumy.txt      |
| igw.tf         | Update igw.tf        |
| instance.tf    | Update instance.tf   |
| output.tf      | Update output.tf     |
| provider.tf    | Update provider.tf   |
| routeassoci.tf | gg                   |
| routetable.tf  | Update routetable.tf |
| sg.tf          | Update sg.tf         |
| subnets.tf     | Update subnets.tf    |
| vpc.tf         | Update vpc.tf        |

The screenshot shows a Jenkins build job named "build-job" with build number #19. The "Changes" tab is selected, showing a single change: "Create dumy.txt" by "noreply". The "Summary" section indicates the build was successful. The Jenkins logo is visible at the top.

**Changes**

Summary

1. Create dumy.txt ([details](#))

Commit [eee17ca4137698a5175f838bc1f8763ec08300dd](#) by noreply  
Create dumy.txt

+ dumy.txt

Status

</> Changes

Console Output

Edit Build Information

Delete build '#19'

Git Build Data

← Previous Build

The screenshot shows the same Jenkins build job #19. The "Changes" tab is selected, showing the commit details: "Create dumy.txt" by "noreply" with revision [eee17ca4137698a5175f838bc1f8763ec08300dd](#). The Jenkins logo is visible at the top.

**#19 (Mar 8, 2024, 7:23:39 AM)**

Status

</> Changes

Console Output

Edit Build Information

Delete build '#19'

Git Build Data

← Previous Build

Changes

1. Create dumy.txt ([details](#) / [githubweb](#))

Started by user ramakrishna

Revision: [eee17ca4137698a5175f838bc1f8763ec08300dd](#)  
Repository: <https://github.com/rama861/python-terraform.git>

- refs/remotes/origin/main

# WEBHOOKS

- In build triggers section select github hook
- Next go to github and select repository on the left side we have seen webhooks click and add webhooks
- In the payload URL enter Jenkins ip and content type JSON

Dashboard > build-job > Configuration

**Configure**

**Build Triggers**

- Trigger builds remotely (e.g., from scripts) ?
- Build after other projects are built ?
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

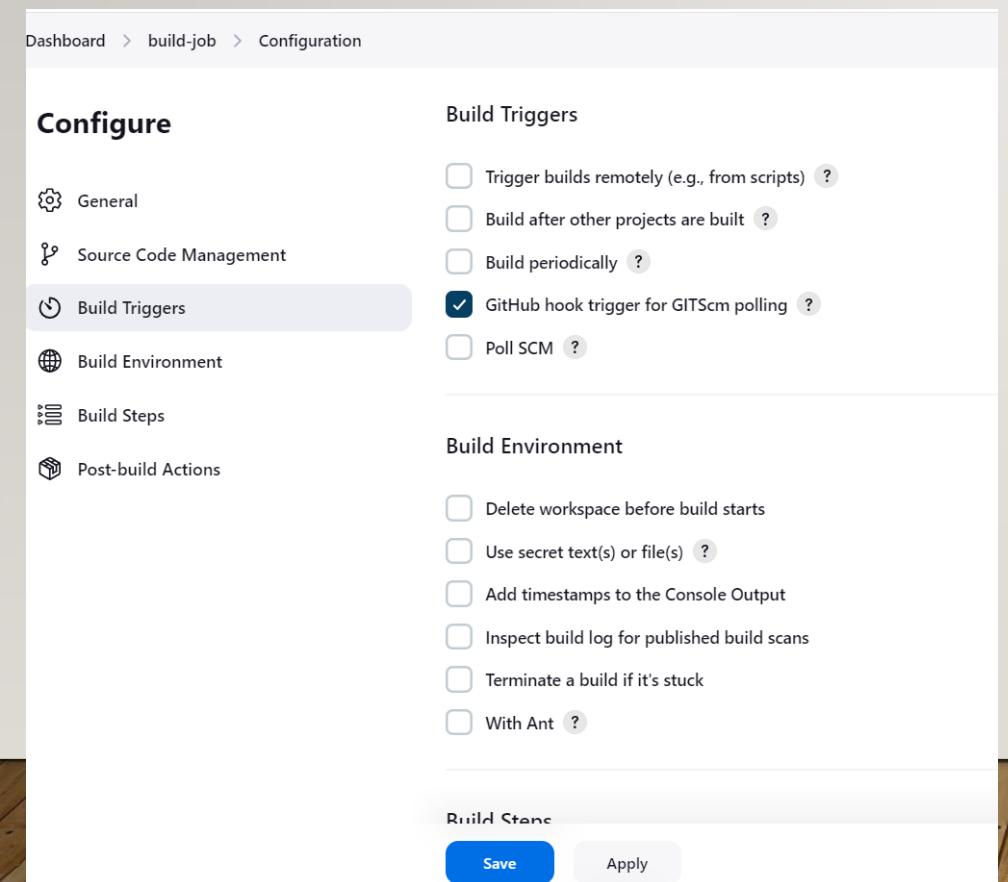
**Build Environment**

- Delete workspace before build starts
- Use secret text(s) or file(s) ?
- Add timestamps to the Console Output
- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant ?

**Build Steps**

**Post-build Actions**

**Save** **Apply**



rama861 / python-terraform

Type ⌘ to search | > | + | ○ | □

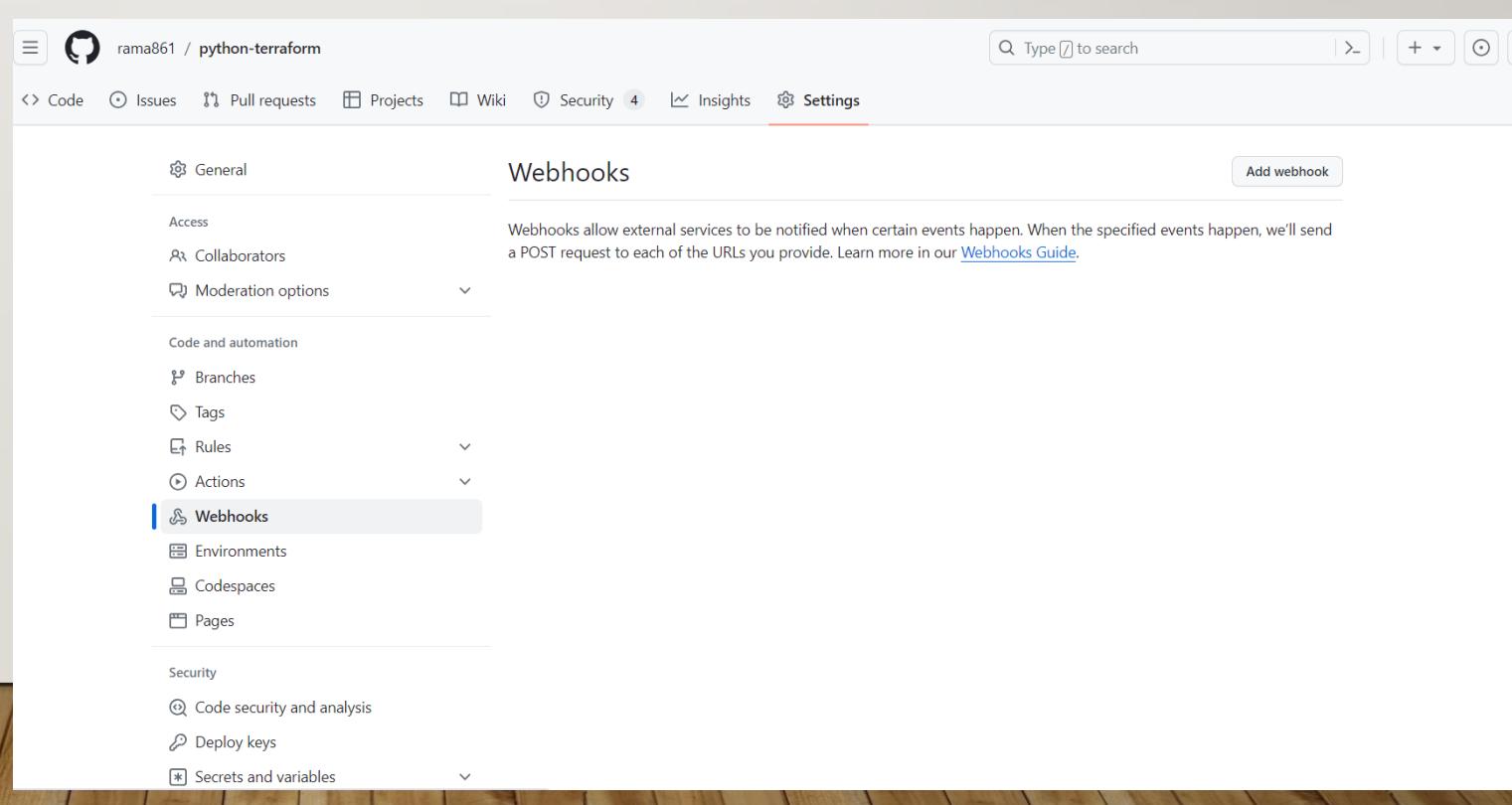
**General** **Webhooks** **Add webhook**

Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions

**Webhooks** Environments Codespaces Pages

Security Code security and analysis Deploy keys Secrets and variables



rama861 / python-terraform

Type  to search

Code Issues Pull requests Projects Wiki Security 4 Insights Settings

**General**

Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions

**Webhooks**

Add webhook

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ <http://52.91.37.161:8080/github-we...> (all events)

Edit Delete

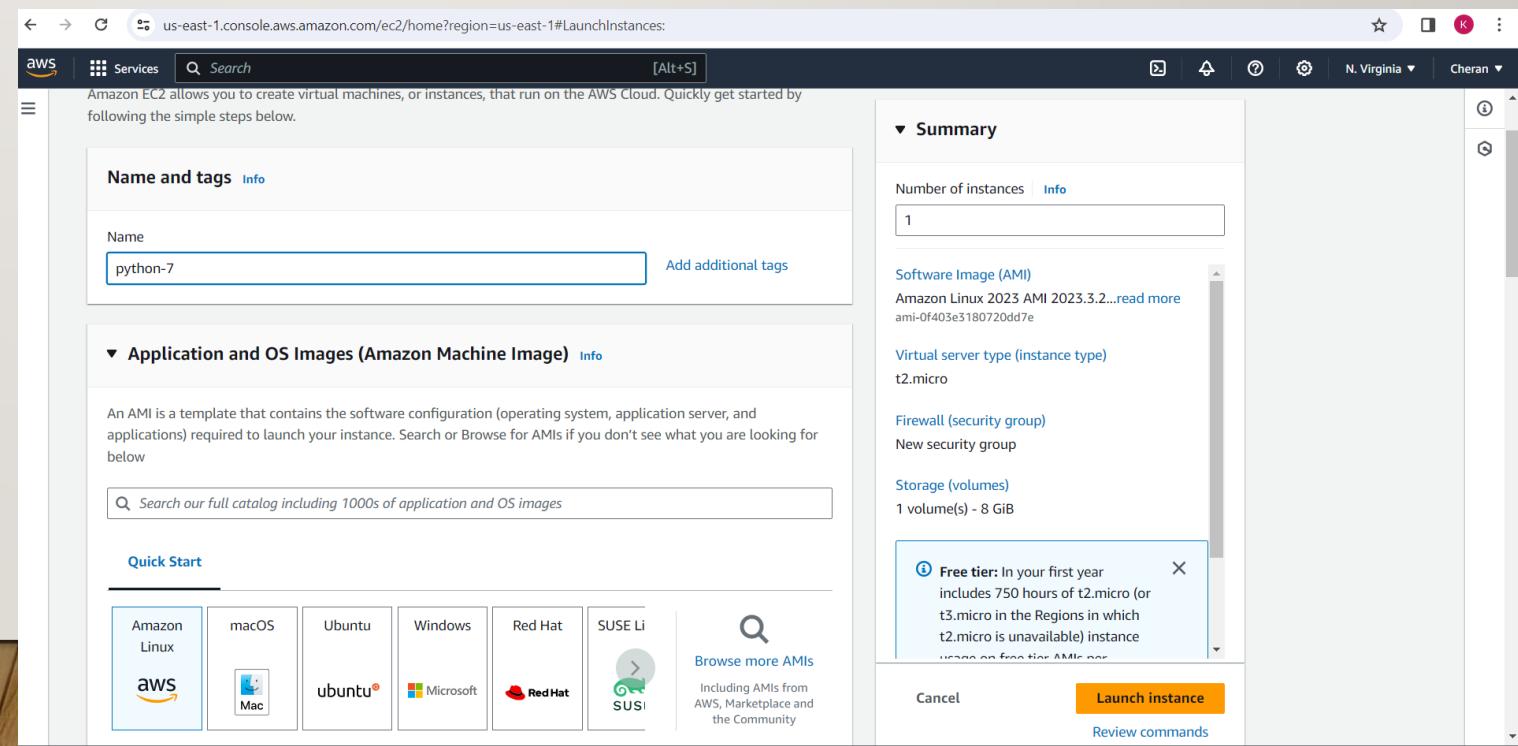
Environments Codespaces Pages

Code security and analysis Deploy keys Secrets and variables

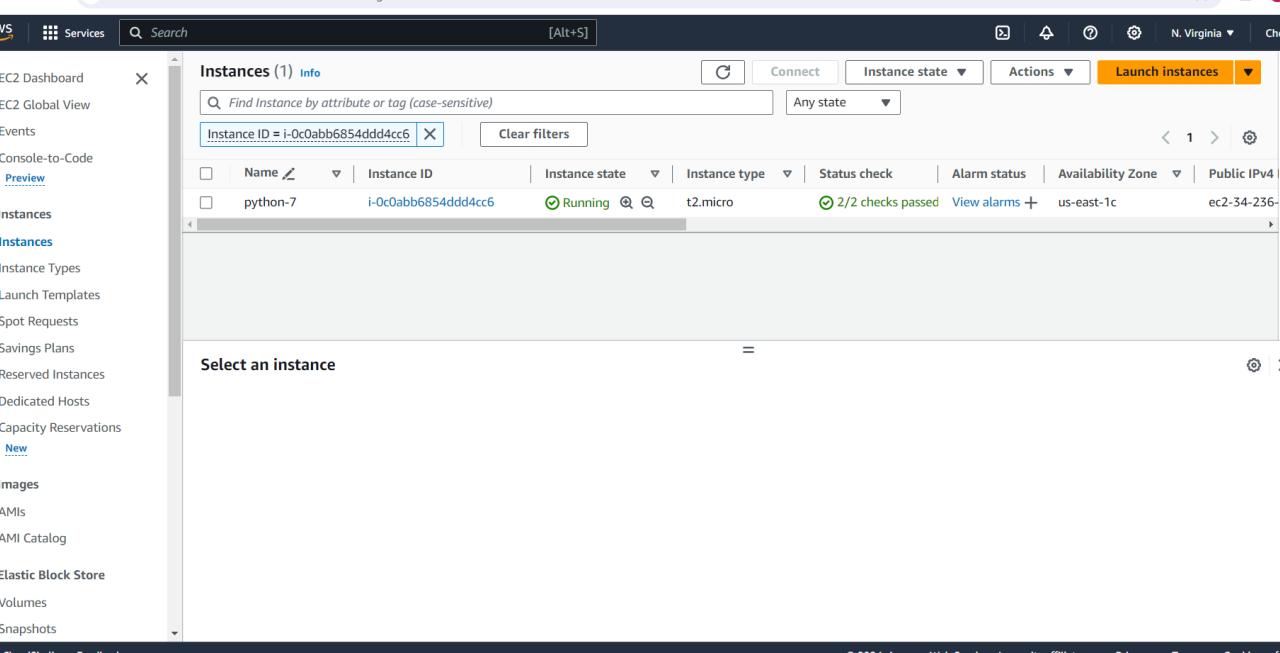
This screenshot shows the GitHub settings page for the repository 'rama861/python-terraform'. The 'Settings' tab is selected. On the left, a sidebar lists various settings categories like General, Access, and Code and automation. The 'Webhooks' category is highlighted with a blue bar. In the main content area, the 'Webhooks' section is displayed with a heading 'Webhooks' and a sub-section 'All events'. A single webhook is listed with the URL 'http://52.91.37.161:8080/github-we...' and a note '(all events)'. There are 'Edit' and 'Delete' buttons next to the webhook entry. Below the main content, there are sections for Environments, Codespaces, and Pages.

# METHOD 7

## BUILD AND DEPLOY PYTHON APPLICATION WITH DOCKER (WRITE DOCKER FILE AND RUN CONTAINERS) AND PUSH CREATED DOCKER IMAGE IN THE DOCKER HUB



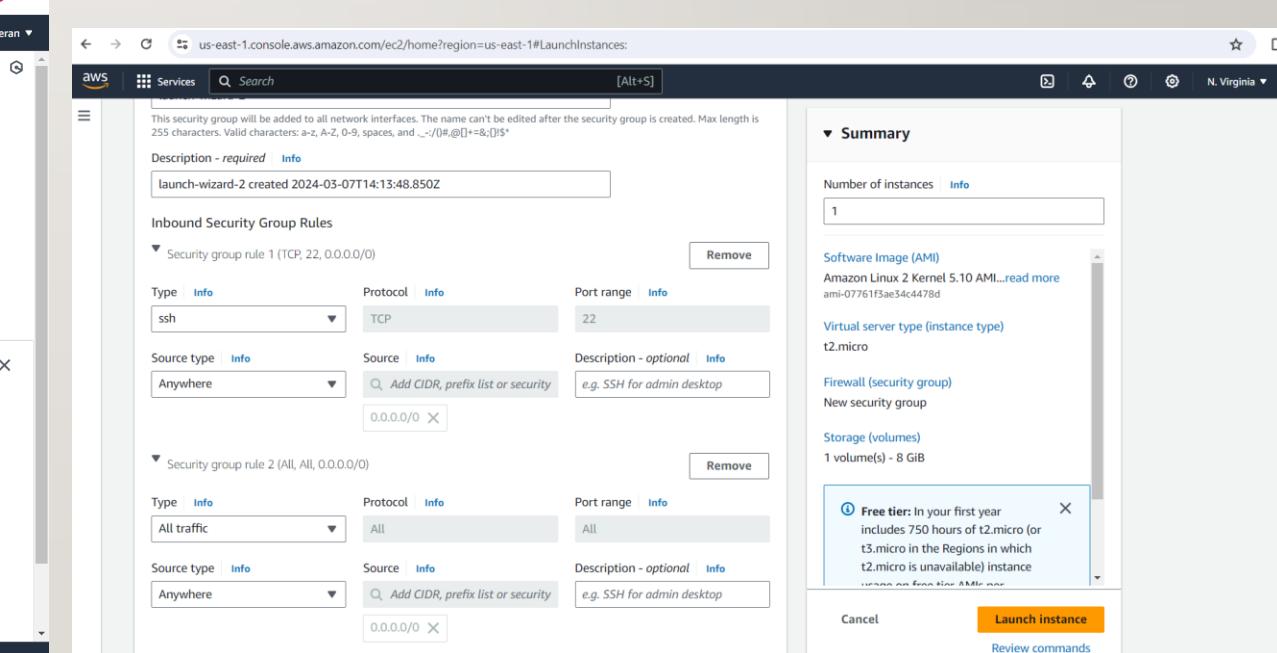
Launch an instance with given the name and instance type as t2.micro  
and give keypair and port numbers SSH and allow all traffic



The screenshot shows the AWS EC2 Instances page. It displays a single instance named "python-7" with the following details:

- Instance ID: i-0cabb6854ddd4cc6
- Name: python-7
- Instance state: Running
- Instance type: t2.micro
- Status check: 2/2 checks passed
- Alarm status: View alarms
- Availability Zone: us-east-1c
- Public IPv4: ec2-34-236-

A modal window titled "Select an instance" is open, showing the same instance details.



The screenshot shows the "Launch Instances" wizard on the "Security group" step. The security group is named "launch-wizard-2 created 2024-03-07T14:13:48.850Z". It contains two inbound rules:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**:
  - Type: ssh
  - Protocol: TCP
  - Port range: 22
  - Source type: Anywhere
  - Description: e.g. SSH for admin desktop
- Security group rule 2 (All, All, 0.0.0.0/0)**:
  - Type: All traffic
  - Protocol: All
  - Port range: All
  - Source type: Anywhere
  - Description: e.g. SSH for admin desktop

The "Summary" section on the right indicates 1 instance and provides links for the AMI, instance type (t2.micro), and security group. A note about the free tier is also present.

- After connect an instance to terminal next we need to install docker , srtart and enable it

---

```
Sudo yum install docker -y  
Sudo systemctl start docker  
Sudo systemctl enable docker
```

```
[ec2-user@ip-172-31-0-197 ~]$ sudo yum -y install docker  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
Resolving Dependencies  
--> Running transaction check  
--> Package docker.x86_64 0:20.10.25-1.amzn2.0.4 will be installed  
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.4.x86_64  
--> Processing Dependency: libcgROUP >= 0.40.rc1-5.15 for package: docker-20.10.25-1.amzn2.0.4.x86_64  
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.4.x86_64  
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.4.x86_64  
--> Running transaction check  
--> Package containerd.x86_64 0:1.7.11-1.amzn2.0.1 will be installed  
--> Package libcgROUP.x86_64 0:0.41-21.amzn2 will be installed  
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed  
--> Package runc.x86_64 0:1.1.11-1.amzn2 will be installed  
-> Finished Dependency Resolution | 3.6 kB 00:00
```

We are giving some permissions for docker and create dockerfile

---

```
sudo yum install -y docker
sudo systemctl start docker
sudo systemctl enable docker
sudo chmod 666 /var/run/docker.sock
sudo usermod -aG docker ec2-user
sudo curl -L "https://github.com/docker/compose/releases/latest/download/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
sudo chmod +x /usr/local/bin/docker-compose
```

- After we install start and enable docker and execute the commands
  - To pull the image from docker execute command `docker run -dt --name python-agri -p 5000:5000 python-agri`
- 

Sudo vi dockerfile

`sudo chmod 666 /var/run/docker.sock`

`docker build -t python-agri .`

`docker images`

`docker ps`

`docker run -dt --name python-agri -p 5000:5000 python-agri`

ec2-user@ip-172-31-0-19:~

```
FROM amazonlinux:2
run yum update -y
RUN yum install -y git python3-pip python3-setuptools
RUN git clone https://github.com/rama861/python-agri.git
WORKDIR /python-agri
RUN pip3 install --no-cache-dir -r requirements.txt
EXPOSE 5000
CMD ["python3", "./app.py"]
```

- Here we can check the docker files and check the status of docker images created or not

```
successfully tagged python-agri:latest
[ec2-user@ip-172-31-0-197 ~]$ docker images
REPOSITORY      TAG      IMAGE ID      CREATED      SIZE
python-agri     latest    3a5fe217ac76  12 seconds ago  1.85GB
amazonlinux     2         2c4ee8379f37  9 days ago   166MB
[ec2-user@ip-172-31-0-197 ~]$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS      NAMES
[ec2-user@ip-172-31-0-197 ~]$ docker run -dt --name python-agri -p 5000:5000 agri
Unable to find image 'agri:latest' locally
docker: Error response from daemon: pull access denied for agri, repository does not exist or may require 'docker login': den-
stated access to the resource is denied.
See 'docker run --help'.
[ec2-user@ip-172-31-0-197 ~]$ docker run -dt --name python-agri -p 5000:5000 python-agri
2bb74494e55352db373cc80113c5f40e1d0d0dda426afb721e8c122617d66ac1
[ec2-user@ip-172-31-0-197 ~]$ docker ps
CONTAINER ID      IMAGE      COMMAND      CREATED      STATUS      PORTS
2bb74494e553      python-agri    "python3 ./app.py"  11 seconds ago  Up 10 seconds  0.0.0.0:5000->5000/tcp, :::5000->5000/tcp
[ec2-user@ip-172-31-0-197 ~]$ ^C
[ec2-user@ip-172-31-0-197 ~]$
```

Now copy and browse the public ip of an instance along with port number 5000 the official page host of AGRI



# Crop Recommendation

Nitrogen

Phosphorous

Potassium

Temperature(C)

Humidity

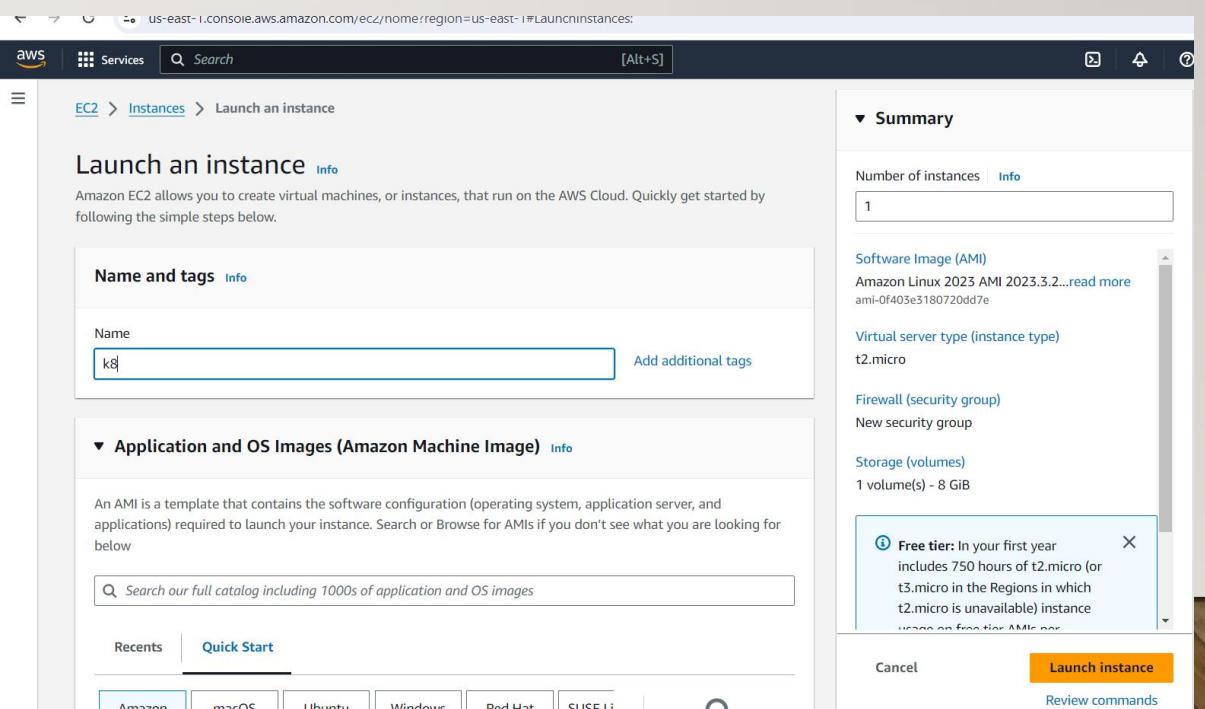
PH Value

Rainfall

PREDICT

# METHOD-8

## BUILD AND DEPLOY PYTHON APPLICATIONS WITH DOCKER AND K8 (EKS AND KOPS)(USE DECLARATIVE MANIFEST METHOD ALONG WITH DOCKER IMAGE)



- Launch an instance with given port numbers and keypair and instance type
- Here the instance will be created

The image displays two screenshots of the AWS Management Console interface.

**Left Screenshot: Security Group Configuration**

This screenshot shows the configuration of a new security group named "launch-wizard-5".

- Description:** launch-wizard-5 created 2024-03-07T16:07:45.173Z
- Inbound Security Group Rules:**
  - Security group rule 1 (TCP, 22, 0.0.0.0/0):** Type: ssh, Protocol: TCP, Port range: 22. Source type: Anywhere (0.0.0.0/0).
  - Security group rule 2 (All, All, 0.0.0.0/0):** Type: All traffic, Protocol: All, Port range: All. Source type: Anywhere (0.0.0.0/0).

**Right Screenshot: EC2 Instances Overview**

This screenshot shows the EC2 Instances page with one instance listed:

- Instances (1/1) Info:** Name: k8, Instance ID: i-0932fd5e764b08a4f, Instance state: Running, Instance type: t2.micro, Status check: 2/2 checks passed, Availability Zone: us-east-1c, Public IP: ec2-3-90-103-118.compute-1.amazonaws.com.
- Instance: i-0932fd5e764b08a4f (k8) - Details:**
  - Instance summary:** Instance ID: i-0932fd5e764b08a4f (k8), Public IPv4 address: 3.90.103.118, Instance state: Running, Private IP: ip-172-31-0-8.ec2.internal.
  - Networking:** Hostname type: Private IP DNS name (IPv4 only), IP name: ip-172-31-0-8.ec2.internal.

- Connect the instance with any one terminal and give the command sudo yum update for update

```
ADMIN@DESKTOP-SNJ538V MINGW64 ~/downloads
$ ssh -i "shiva.pem" ec2-user@ec2-3-90-103-118.compute-1.amazonaws.com
The authenticity of host 'ec2-3-90-103-118.compute-1.amazonaws.com (3.90.103.118)' can't be
ED25519 key fingerprint is SHA256:aUQewzXJVgPcgwb8WaKnDhRpdm+BA4cZ9D5GeRKghGs.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'ec2-3-90-103-118.compute-1.amazonaws.com' (ED25519) to the list
s.

      #_
  ~\_\####_          Amazon Linux 2
~~ \_\#####\
~~   \###|          AL2 End of Life is 2025-06-30.
~~     \#/_
~~       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/

10 package(s) needed for security, out of 10 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-172-31-0-8 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
```

Next we need to install docker and start and enable

```
Complete!
[ec2-user@ip-172-31-0-8 ~]$ sudo yum -y install docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
1 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:20.10.25-1.amzn2.0.4 will be installed
--> Processing Dependency: containerd >= 1.3.2 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: libcgroup >= 0.40.rc1-5.15 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: runc >= 1.0.0 for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Processing Dependency: pigz for package: docker-20.10.25-1.amzn2.0.4.x86_64
--> Running transaction check
--> Package containerd.x86_64 0:1.7.11-1.amzn2.0.1 will be installed
--> Package libcgroup.x86_64 0:0.41-21.amzn2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Package runc.x86_64 0:1.1.11-1.amzn2 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
          Package           Arch         Version        Repository      Size
=====
Installing:
  docker            x86_64       20.10.25-1.amzn2.0.4    amzn2extra-docker   43 M
Installing for dependencies:
  containerd        x86_64       1.7.11-1.amzn2.0.1    amzn2extra-docker   30 M
  libcgroup         x86_64       0.41-21.amzn2          amzn2-core          66 k
  pigz              x86_64       2.3.4-1.amzn2.0.1     amzn2-core          81 k
  runc              x86_64       1.1.11-1.amzn2        amzn2extra-docker   3.0 M

Transaction Summary
=====
Install 1 Package (+4 Dependent packages)

Total download size: 77 M
Installed size: 287 M
```

```
Complete!
[ec2-user@ip-172-31-0-8 ~]$ sudo systemctl start docker
[ec2-user@ip-172-31-0-8 ~]$ sudo systemctl enable docker
Created symlink from /etc/systemd/system/multi-user.target.wants/docker.service to /usr/lib/systemd/system/docker.service.
[ec2-user@ip-172-31-0-8 ~]$ |
```

## Create docker file with data

```
FROM amazonlinux:2
RUN yum update -y
RUN yum install -y git python3-pip python3-setuptools
RUN git clone https://github.com/rama861/python-carperdition.git
WORKDIR /python-carperdition
RUN pip3 install --no-cache-dir -r requirements.txt
EXPOSE 8000
CMD ["python3", "./app.py"]
~
```

## INSTALL KUBECTL

```
cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo
```

```
[kubernetes]
```

```
name=Kubernetes
```

```
baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/
```

```
enabled=1
```

```
gpgcheck=1
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repo/repodata/repomd.xml.key
```

```
EOF
```

```
Sudo yum install -y kubectl
```

```
Kubectl cluster-info
```

```
cat
```

```
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repo/repodata/repomd.xml.key  
[ec2-user@ip-172-31-0-159 ~]$ sudo yum install -y kubectl  
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd  
[ec2-user@ip-172-31-0-159 ~]$
```

```
[ec2-user@ip-172-31-0-8 ~]$ cat <<EOF | sudo tee /etc/yum.repos.d/kubernetes.repo  
> [kubernetes]  
> name=Kubernetes  
> baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/  
> enabled=1  
> gpgcheck=1  
> gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repo/repomd.xml.key  
> EOF  
[kubernetes]  
name=Kubernetes  
baseurl=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/  
enabled=1  
gpgcheck=1  
gpgkey=https://pkgs.k8s.io/core:/stable:/v1.29/rpm/repo/repomd.xml.key  
[ec2-user@ip-172-31-0-8 ~]$ next .....  
-bash: next: command not found  
[ec2-user@ip-172-31-0-8 ~]$ curl -Lo kops https://github.com/kubernetes/kops/releases/download/$(curl -s https://api.github.com/repos/kubernetes/kops/releases/latest | grep tag_name | cut -d '"' -f 4)/kops-linux-amd64  
% Total    % Received % Xferd  Average Speed   Time   Time  Current  
          Dload  Upload   Total Spent    Left  Speed  
 0       0      0      0      0      0      0  --:--:--  --:--:--  --:--:--     0  
100  177M  100  177M      0      0  114M  0  0:00:01  0:00:01  121M  
[ec2-user@ip-172-31-0-8 ~]$
```

## Install kops

- 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
- aws s3 mb s3://python-Kubernetes
- export KOPS\_STORE\_STATE=s3://python-kuberneteskops create cluster --name python.k8s.local --state s3://python-kubernetes --zones us-east-2a,us-east-2b --node-count 1 --yes

complete.

```
[ec2-user@ip-172-31-0-159 ~]$ 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
```

- Next we are giving AWS CONFIGURE with access key and secret key
  - After that we create S3 bucket
  - After that we export S3 bucket to kops
- 

```
[ec2-user@ip-172-31-0-159 ~]$ chmod +x kops
[ec2-user@ip-172-31-0-159 ~]$ sudo mv kops /usr/local/bin/kops
[ec2-user@ip-172-31-0-159 ~]$ aws s3 mb s3://rama-kubernetes
make_bucket failed: s3://rama-kubernetes Unable to locate credentials
[ec2-user@ip-172-31-0-159 ~]$ aws configure
AWS Access Key ID [None]: AKIAQ3EGWRWVOGMFV2VZ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
[ec2-user@ip-172-31-0-159 ~]$ aws s3 mb s3://rama-kubernetes
make_bucket: rama-kubernetes
[ec2-user@ip-172-31-0-159 ~]$ export KOPS_STORE_STATE=s3://rama-kubernetes
```

```
[ec2-user@ip-172-31-0-159 ~]$ aws configure
AWS Access Key ID [None]: AKIAQ3EGWRWVOGMFV2VZ
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]:
[ec2-user@ip-172-31-0-159 ~]$ aws s3 mb s3://rama-kubernetes
make_bucket: rama-kubernetes
[ec2-user@ip-172-31-0-159 ~]$ export KOPS_STORE_STATE=s3://rama-kubernetes
[ec2-user@ip-172-31-0-159 ~]$ kops create cluster --name python.k8s.local --state s3://python-kubernetes --zones us-east-2a,us-east-2b --node-count 1 --yes
```

## Check kops version and kubectl version

```
[ec2-user@ip-172-31-0-8 ~]$ kops version
Client version: 1.28.4 (git-v1.28.4)
[ec2-user@ip-172-31-0-8 ~]$ kubectl version
Client Version: v1.29.2
Kustomize Version: v5.0.4-0.20230601165947-6ce0bf390ce3
The connection to the server localhost:8080 was refused
[ec2-user@ip-172-31-0-8 ~]$
```

## Deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: fish
spec:
  replicas: 2
  selector:
    matchLabels:
      app: fish
  template:
    metadata:
      labels:
        app: fish
    spec:
      containers:
        - name: fish
          image: alaric123/python-fish
      ports:
        - containerPort: 80
```

## Service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: fish-lb
spec:
  selector:
    app: fish
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8000
  type: LoadBalancer
```

- After creating `deployment.yaml` and `service.yaml` file execute commands
- `Kubectl apply -f deployment.yaml`
- `Kubectl apply -f service.yaml`
- Next give the command for all resources created
- `Kubectl get all`

```
[ec2-user@ip-172-31-0-159 ~]$ sudo vi deployment.yaml
[ec2-user@ip-172-31-0-159 ~]$ vi service.yaml
[ec2-user@ip-172-31-0-159 ~]$ kubectl apply -f deployment.yaml
Error from server (BadRequest): error when creating "deployment.yaml": Deployment in version "v1" cannot be han-
dled as a Deployment: json: cannot unmarshal string into Go struct field Container.spec.template.spec.container
s.ports of type v1.ContainerPort
[ec2-user@ip-172-31-0-159 ~]$ sudo vi deployment.yaml
[ec2-user@ip-172-31-0-159 ~]$ kubectl apply -f deployment.yaml
deployment.apps/python-fish created
[ec2-user@ip-172-31-0-159 ~]$ kubectl apply -f service.yaml
service/python-fish created
[ec2-user@ip-172-31-0-159 ~]$ kubectl get all
```

```
[ec2-user@ip-172-31-0-8 ~]$ vi service.yaml
[ec2-user@ip-172-31-0-8 ~]$ kubectl get all
NAME                 READY   STATUS            RESTARTS   AGE
pod/fish-686fd8858-1g8l2  0/1    ImagePullBackoff  0          9m43s
pod/fish-686fd8858-wqkbj  0/1    ImagePullBackoff  0          9m43s

NAME           TYPE        CLUSTER-IP      EXTERNAL-IP
              PORT(S)     AGE
service/fish-lb   LoadBalancer  100.67.191.36  adbe912befee64ab487df96437b687e9-525968268.us-east-1.elb.amazonaws.com
azonaws.com     80:32324/TCP  7m45s
service/kubernetes ClusterIP  100.64.0.1    <none>
                  443/TCP    35m

NAME             READY   UP-TO-DATE   AVAILABLE   AGE
deployment.apps/fish  0/2       2           0           9m43s

NAME           DESIRED   CURRENT   READY   AGE
replicaset.apps/fish-686fd8858  2         2         0         9m43s
[ec2-user@ip-172-31-0-8 ~]$
```

Next copy the external ip and browse it in the browser  
The official page of fish will be host



# FISH WEIGHT

**Species \***

Bream

**Vertical length in cm \***

23.2

**Diagonal length in cm \***

25.4

**Cross length in cm \***

30.0

**Height \***

11.5200

**Width \***

4.0200

**SUBMIT**

We created docker image using k8 method

---

A screenshot of a web browser displaying the Docker Hub website at [hub.docker.com](https://hub.docker.com). The page has a blue header with navigation links for 'Explore', 'Repositories' (which is underlined), and 'Organizations'. A search bar says 'Search Docker Hub' with a 'ctrl+K' keyboard shortcut. Below the header, there are dropdown menus for 'rama899' (selected), 'Search by repository name', and 'All Content'. A large blue button says 'Create repository'. In the main content area, a repository card for 'rama899 / python-fish' is shown, indicating it contains an image and was last pushed 6 days ago. The repository is marked as 'Security unknown', has 0 stars, 10 downloads, and is public. To the right of the repository card, there is a red circular icon with a white 'P' and a dashed blue line.

## Resources will be created

The screenshot shows the AWS EC2 Instances page with the following details:

**Instances (4) Info**

| Name              | Instance ID         | Instance state | Instance type | Status check      | Alarm status                | Availability Zone | Public IPv4 DNS  |
|-------------------|---------------------|----------------|---------------|-------------------|-----------------------------|-------------------|------------------|
| nodes-us-east...  | i-0304fb7a8f7829cad | Running        | t3.medium     | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1b        | ec2-3-92-179-154 |
| k8                | i-0932fd5e764b08a4f | Running        | t2.micro      | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1c        | ec2-3-90-103-118 |
| control-plane-... | i-08bd255d366406574 | Running        | t3.medium     | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1a        | ec2-44-204-245-1 |
| nodes-us-east...  | i-017a93350559bf487 | Running        | t3.medium     | 2/2 checks passed | <a href="#">View alarms</a> | us-east-1a        | ec2-3-235-86-134 |

**Select an instance**

The screenshot shows the AWS VPC dashboard with the following details:

**Your VPCs (1/3) Info**

| Name           | VPC ID                | State     | IPv4 CIDR     | IPv6 CIDR                | DHCP o  |
|----------------|-----------------------|-----------|---------------|--------------------------|---------|
| vpc            | vpc-044c7a3ad637c7164 | Available | 10.0.0.0/16   | -                        | dopt-04 |
| -              | vpc-00db2eefe23cdafe0 | Available | 172.31.0.0/16 | -                        | dopt-04 |
| rama.k8s.local | vpc-0682dc41aa8fe3feb | Available | 172.20.0.0/16 | 2600:1f18:284d:8c00::/56 | dopt-07 |

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

The screenshot shows the AWS VPC Subnets page. The left sidebar is collapsed. The main area title is "Subnets (2/4) Info". A search bar says "Find resources by attribute or tag". The table has columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. Two subnets are listed as available:

| Name                      | Subnet ID                | State     | VPC                              | IPv4 CIDR       |
|---------------------------|--------------------------|-----------|----------------------------------|-----------------|
| us-east-1a.rama.k8s.local | subnet-0baa460d81c3e1468 | Available | vpc-0682dc41aa8fe3feb   rama...  | 172.20.0.0/17   |
| us-east-1b.rama.k8s.local | subnet-0ff304468afef4592 | Available | vpc-0682dc41aa8fe3feb   rama.... | 172.20.128.0/17 |
| python1                   | subnet-0cc8b5e8e7a506113 | Available | vpc-00db2eeefe23cdafe0           | 172.31.0.0/24   |
| subnet                    | subnet-011fa0f36e7303768 | Available | vpc-044c7a3ad637c7164   vpc      | 10.0.1.0/24     |

Actions: Create subnet

us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#route-tables:

The screenshot shows the AWS VPC Route tables page. The left sidebar is collapsed. The main area title is "Route tables (1/5) Info". A search bar says "Find resources by attribute or tag". The table has columns: Name, Route table ID, Explicit subnet associations, Edge associations, Main, and VPC. One route table is listed as Main:

| Name           | Route table ID        | Explicit subnet associations | Edge associations | Main | VPC                    |
|----------------|-----------------------|------------------------------|-------------------|------|------------------------|
| -              | rtb-02928865eee17829b | -                            | -                 | Yes  | vpc-0682dc41aa8fe3feb  |
| routetable     | rtb-0ece9dcf7d7bd6540 | subnet-011fa0f36e7303...     | -                 | No   | vpc-044c7a3ad637c7164  |
| -              | rtb-0110951872c70f76d | -                            | -                 | Yes  | vpc-044c7a3ad637c7164  |
| -              | rtb-0797070512abe5f39 | subnet-0cc8b5e8e7a506...     | -                 | Yes  | vpc-00db2eeefe23cdafe0 |
| rama.k8s.local | rtb-00dc12ee54343759c | 2 subnets                    | -                 | No   | vpc-0682dc41aa8fe3feb  |

Actions: Create route table

AWS Services Search [Alt+S] N. Virginia Cheran

VPC dashboard EC2 Global View Filter by VPC: Select a VPC Virtual private cloud Your VPCs Subnets Route tables Internet gateways

### Internet gateways (1/3) Info

| Name           | Internet gateway ID   | State    | VPC ID                                 | Owner        |
|----------------|-----------------------|----------|----------------------------------------|--------------|
| -              | igw-0c8c9bb4a5aa8a513 | Attached | vpc-044c7a3ad637c7164   vpc            | 058264554922 |
| -              | igw-0d6e98e7a387966d2 | Attached | vpc-00db2eefe23cdafe0                  | 058264554922 |
| rama.k8s.local | igw-0e2b4b3747b65654d | Attached | vpc-0682dc41aa8fe3feb   rama.k8s.local | 058264554922 |

## METHOD-10

BUILD AND DEPLOY PYTHON APPLICATIONS WITH  
Pipeline Method (Create Clone Job And Build Job)  
WITH BUILD PERIODICALLY,POLL SCM AND WEBHOOKS)

The screenshot shows the AWS EC2 "Launch an instance" wizard. The process is currently at the "Name and tags" step.

**Name and tags**

Name: pipeline

Add additional tags

**Application and OS Images (Amazon Machine Image)**

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Search our full catalog including 1000s of application and OS images

**Quick Start**

Amazon macOS Ubuntu Windows Red Hat SUSE Li

**Summary**

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.3.2...read more  
ami-0f403e3180720dd7e

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

**Free tier:** In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on frontier AMIs per month

Cancel Launch instance Review commands

AWS CloudFormation console showing the configuration of a new AWS Lambda function.

**Subnet:** subnet-0cc8b5e8e7a506113 | python1

**Auto-assign public IP:** Info

**Enable:**

**Firewall (security groups):** Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group    Select existing security group

**Common security groups:** Info

Select security groups

default sg-07520cf9272b05dfc X  
VPC: vpc-00db2eef23cdaf0

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Configure storage:** Info

Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

**Summary:**

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-02d7fd1c2af6ead0
- Virtual server type (instance type): t2.micro
- Firewall (security group): default
- Storage (volumes): 1 volume(s) - 8 GiB

Cancel   Launch instance   Review commands

AWS CloudFormation console showing the configuration of a new AWS Lambda function.

**Subnet:** subnet-0cc8b5e8e7a506113 | python1

**Auto-assign public IP:** Info

**Enable:**

**Firewall (security groups):** Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group    Select existing security group

**Common security groups:** Info

Select security groups

default sg-07520cf9272b05dfc X  
VPC: vpc-00db2eef23cdaf0

Compare security group rules

Security groups that you add or remove here will be added to or removed from all your network interfaces.

**Configure storage:** Info

Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage X

**Summary:**

- Number of instances: 1
- Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-02d7fd1c2af6ead0
- Virtual server type (instance type): t2.micro
- Firewall (security group): default
- Storage (volumes): 1 volume(s) - 8 GiB

Cancel   Launch instance   Review commands

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.  
<https://aws.amazon.com/linux/amazon-linux-2023/>

```
[ec2-user@ip-172-31-0-155 ~]$ sudo yum update
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[ec2-user@ip-172-31-0-155 ~]$
```

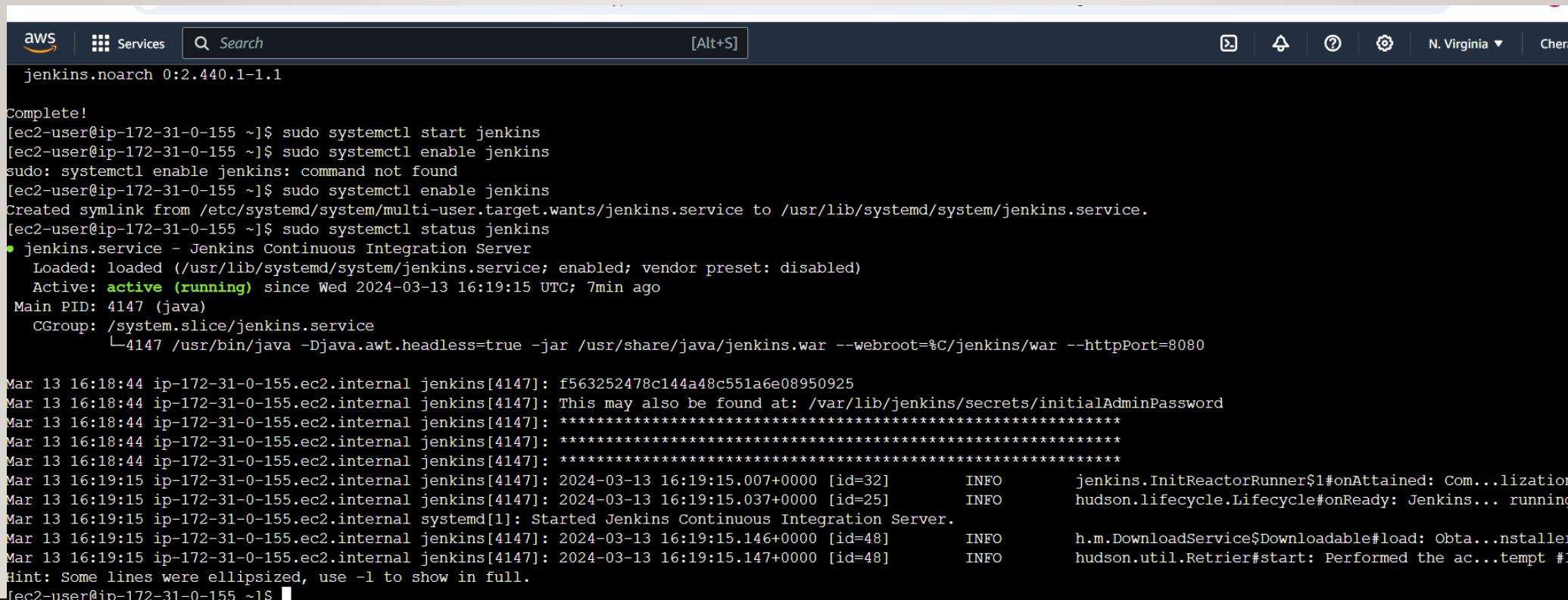
- We need to install Jenkins with commands
- sudo wget -O /etc/yum.repos.d/jenkins.repo \  
<https://pkg.jenkins.io/redhat-stable/jenkins.repo>
- sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io-2023.key>
- sudo yum upgrade -y
- sudo yum install -y java-17\*
- sudo yum install -y Jenkins
- sudo systemctl start Jenkins
- sudo systemctl enable jenkins

```
No packages marked for update
[ec2-user@ip-172-31-0-155 ~]$ sudo wget -O /etc/yum.repos.d/jenkins.repo \
>     https://pkg.jenkins.io/redhat-stable/jenkins.repo
--2024-03-13 16:17:37--  https://pkg.jenkins.io/redhat-stable/jenkins.repo
Resolving pkg.jenkins.io (pkg.jenkins.io)... 146.75.38.133, 2a04:4e42:78::645
Connecting to pkg.jenkins.io (pkg.jenkins.io)|146.75.38.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 85
Saving to: '/etc/yum.repos.d/jenkins.repo'
```

Complete!

```
[ec2-user@ip-172-31-0-155 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-0-155 ~]$ sudo systemctl enable jenkins
sudo: systemctl enable jenkins: command not found
[ec2-user@ip-172-31-0-155 ~]$ 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-172-31-0-155 ~]$
```

- After start and enabling Jenkins next
  - Check the status of Jenkins and it also give the administrative password for unlocking jenkins
- 



The screenshot shows a terminal window within the AWS CloudShell interface. The terminal is running a series of commands to manage the Jenkins service. It starts by checking the Jenkins version, then performs several systemctl operations (start, enable, enable) which fail due to a missing command. It then creates a symbolic link for the service, starts it, and checks its status. The status output shows the Jenkins service is active and running. The terminal then lists log entries from March 13, 2024, at 16:18:44, showing initial admin password generation and Jenkins startup logs. A note at the bottom indicates some lines were ellipsized.

```

jenkins.noarch 0:2.440.1-1.1

Complete!
[ec2-user@ip-172-31-0-155 ~]$ sudo systemctl start jenkins
[ec2-user@ip-172-31-0-155 ~]$ sudo systemctl enable jenkins
sudo: systemctl enable jenkins: command not found
[ec2-user@ip-172-31-0-155 ~]$ 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-172-31-0-155 ~]$ 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 Wed 2024-03-13 16:19:15 UTC; 7min ago
      Main PID: 4147 (java)
        CGroup: /system.slice/jenkins.service
               └─4147 /usr/bin/java -Djava.awt.headless=true -jar /usr/share/java/jenkins.war --webroot=%C/jenkins/war --httpPort=8080

Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: f563252478c144a48c551a6e08950925
Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: This may also be found at: /var/lib/jenkins/secrets/initialAdminPassword
Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: ****
Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: ****
Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: ****
Mar 13 16:18:44 ip-172-31-0-155.ec2.internal jenkins[4147]: 2024-03-13 16:19:15.007+0000 [id=32]           INFO    jenkins.InitReactorRunner$1#onAttained: Com...lization
Mar 13 16:19:15 ip-172-31-0-155.ec2.internal jenkins[4147]: 2024-03-13 16:19:15.037+0000 [id=25]           INFO    hudson.lifecycle.Lifecycle#onReady: Jenkins... running
Mar 13 16:19:15 ip-172-31-0-155.ec2.internal systemd[1]: Started Jenkins Continuous Integration Server.
Mar 13 16:19:15 ip-172-31-0-155.ec2.internal jenkins[4147]: 2024-03-13 16:19:15.146+0000 [id=48]           INFO    h.m.DownloadService$Downloadable#load: Obta...nstaller
Mar 13 16:19:15 ip-172-31-0-155.ec2.internal jenkins[4147]: 2024-03-13 16:19:15.147+0000 [id=48]           INFO    hudson.util.Retrier#start: Performed the ac...tempt #1
Hint: Some lines were ellipsized, use -l to show in full.
[ec2-user@ip-172-31-0-155 ~]$ 
```

The image displays three screenshots of the Jenkins 'Getting Started' wizard, showing the progression from unlocking Jenkins to customizing it.

**Step 1: Unlock Jenkins**

The first screenshot shows the 'Unlock Jenkins' step. It instructs the user to ensure Jenkins is securely set up by reading the password from the log or the file at `/var/lib/jenkins/secrets/initialAdminPassword`. A redacted input field for the administrator password is shown, along with a 'Continue' button.

**Step 2: Customize Jenkins**

The second screenshot shows the 'Customize Jenkins' step. It highlights two options: 'Install suggested plugins' (blue box) and 'Select plugins to install' (light gray box). Both steps involve selecting plugins to extend Jenkins' functionality.

**Step 3: Jenkins Home Page**

The third screenshot shows the Jenkins home page after customization. It features a 'Getting Started' section with a table of available plugins:

| ✓ Folders     | ✓ OWASP Markup Formatter | ✓ Build Timeout                     | ✓ Credentials Binding | ✓ Pipeline: Stage View | ✓ Pipeline: SCM Strategy | ✓ Pipeline: Basic S |
|---------------|--------------------------|-------------------------------------|-----------------------|------------------------|--------------------------|---------------------|
| ✓ Timestamper | ✓ Workspace Cleanup      | ✓ Ant                               | ○ Gradle              | ○ Pipeline: Stage View | ○ Pipeline: SCM Strategy | ** Bootstrap 5 API  |
| ○ Pipeline    | ○ GitHub Branch Source   | ○ Pipeline: GitHub Groovy Libraries | ○ PAM Authentication  | ○ Pipeline: Stage View | ○ Pipeline: SCM Strategy | ** JQuery3 API      |
| ○ Git         | ○ SSH Build Agents       | ○ Matrix Authorization Strategy     | ○ Dark Theme          | ○ Pipeline: Stage View | ○ Pipeline: SCM Strategy | ** ECharts API      |
| ○ LDAP        | ○ Email Extension        | ✓ Mailer                            | ○ Dark Theme          | ○ Pipeline: Stage View | ○ Pipeline: SCM Strategy | ** Display URL API  |

The Jenkins logo and a wooden floor background are visible at the bottom of the screenshots.

## Getting Started

### Create First Admin User

Username

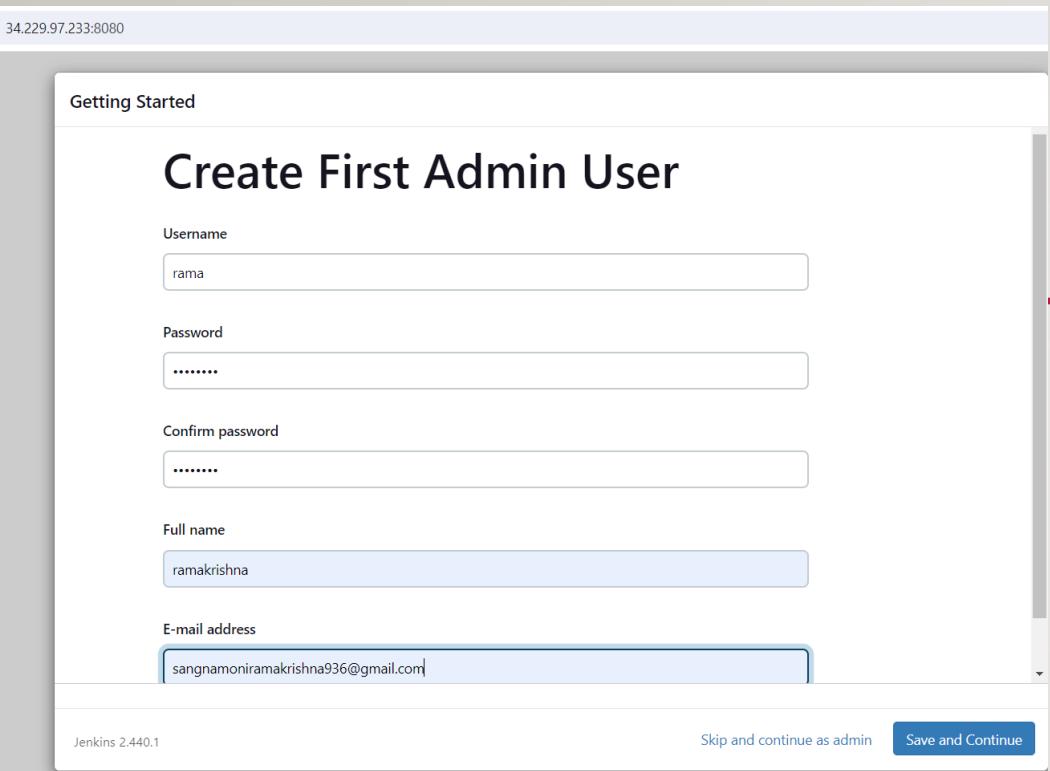
Password

Confirm password

Full name

E-mail address

Jenkins 2.440.1      [Skip and continue as admin](#)      [Save and Continue](#)



## Getting Started

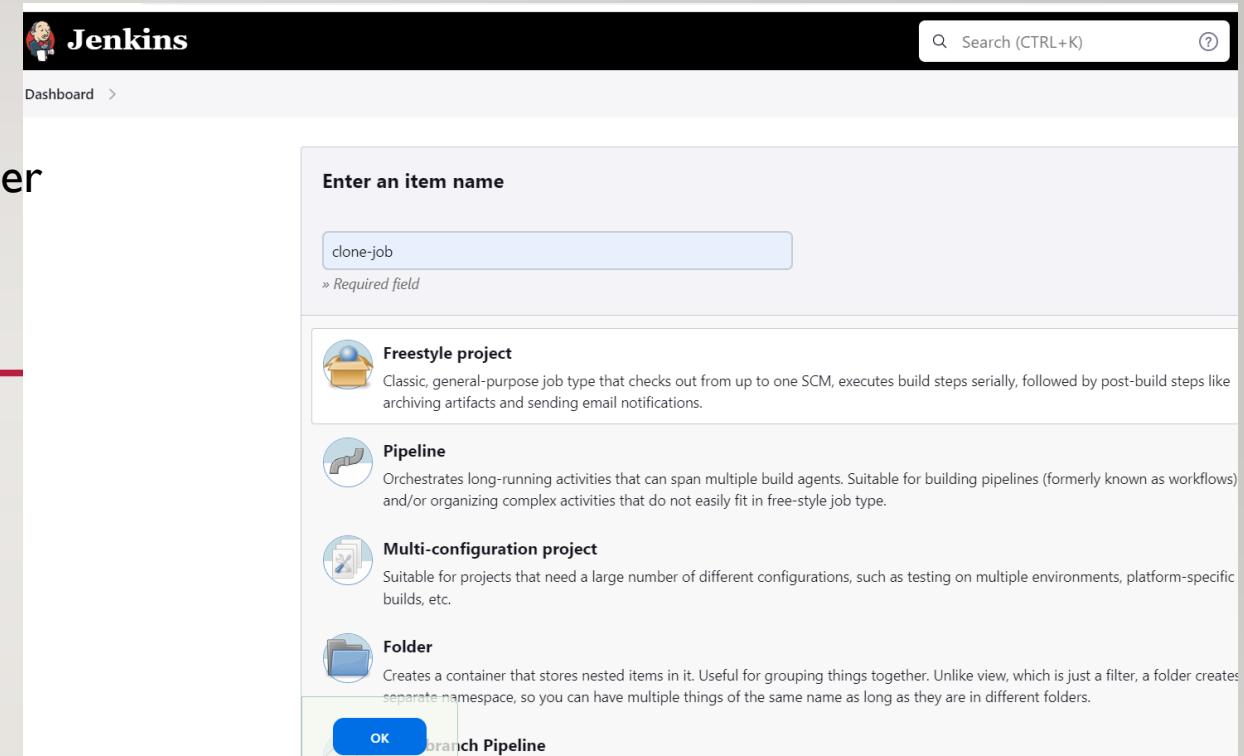
# Instance Configuration

Jenkins URL:

The Jenkins URL is used to provide the root URL for absolute links to various Jenkins resources. That means this value is required for proper operation of many Jenkins features including email notifications, PR status updates, and the `BUILD_URL` environment variable provided to build steps.

The proposed default value shown is **not saved yet** and is generated from the current request, if possible. The best practice is to set this value to the URL that users are expected to use. This will avoid confusion when sharing or viewing links.

Give aws configure or attached to roles and policies to the user



The screenshot shows the Jenkins dashboard. At the top right is a search bar with the placeholder "Search (CTRL+K)". Below it is a "Dashboard" link. The main area has a heading "Enter an item name" with a text input field containing "clone-job". A note below the input says "» Required field". Below the input are four project type options: "Freestyle project" (classic job type), "Pipeline" (orchestrates long-running activities), "Multi-configuration project" (for multiple configurations), and "Folder" (creates a container for nested items). At the bottom right of the list is a blue "OK" button.

```
[ec2-user@ip-172-31-0-155 ~]$ aws configure
AWS Access Key ID [None]: AKIAQ3EGWRWVMWW2CHNN
AWS Secret Access Key [None]:
MW9I1AmLg/Ysyu69jgjDMQQ8o2rWufTff2YQkNPPDefault region name [None]: us-east-1
Default output format [None]: json
[ec2-user@ip-172-31-0-155 ~]$
```

## Enter an item name

build-job

» A job already exists with the name 'build-job'



## 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 folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.



## Multibranch Pipeline

OK

## Configure

## Build Triggers

 Trigger builds remotely (e.g., from scripts) ? Build after other projects are built ?

## Projects to watch

clone-job,

 Trigger only if build is stable Trigger even if the build is unstable Trigger even if the build fails Always trigger, even if the build is aborted Build periodically ? GitHub hook trigger for GITScm polling ? Poll SCM ?

 Status Changes Console Output View as plain text Edit Build Information Delete build '#2' Git Build Data Previous Build Console OutputStarted by user [ramakrishna](#)

Running as SYSTEM

Building in workspace /var/lib/jenkins/workspace/clone-job

The recommended git tool is: NONE

No credentials specified

Cloning the remote Git repository

Cloning repository <https://github.com/rama861/python-fish.git>

&gt; git init /var/lib/jenkins/workspace/clone-job # timeout=10

Fetching upstream changes from <https://github.com/rama861/python-fish.git>

&gt; git --version # timeout=10

&gt; git --version # 'git version 2.40.1'

> git fetch --tags --force --progress -- <https://github.com/rama861/python-fish.git> +refs/heads/\*\:refs/remotes/origin/\* # timeout=10> git config remote.origin.url <https://github.com/rama861/python-fish.git> # timeout=10

&gt; git config --add remote.origin.fetch +refs/heads/\*\:refs/remotes/origin/\* # timeout=10

Avoid second fetch

&gt; git rev-parse refs/remotes/origin/master^{commit} # timeout=10

Checking out Revision 1d84e35c8862b160a4006b1b3864e870eb75e338 (refs/remotes/origin/master)

&gt; git config core.sparsecheckout # timeout=10

&gt; git checkout -f 1d84e35c8862b160a4006b1b3864e870eb75e338 # timeout=10

Commit message: "Update requirements.txt"

First time build. Skipping changelog.

Triggering a new build of [build-job](#)

Finished: SUCCESS

## Configure

### General

### Source Code Management

### Build Triggers

### Build Environment

### Build Steps

### Post-build Actions

- Inspect build log for published build scans
- Terminate a build if it's stuck
- With Ant [?](#)

### Build Steps

#### Execute shell [?](#)

##### Command

See [the list of available environment variables](#)

```
cd /var/lib/jenkins/workspace/clone-job
pip3 install -r requirements.txt
screen -m -d python app.py
```

Advanced ▾

Add build step ▾

Save

Apply



### Status

### Changes

### Console Output

[View as plain text](#)

[Edit Build Information](#)

[Delete build '#5'](#)

[← Previous Build](#)

### Console Output

Started by user ramakrishna  
 Running as SYSTEM  
 Building in workspace /var/lib/jenkins/workspace/build-job  
 [build-job] \$ /bin/sh -xe /tmp/jenkins6955245427552935850.sh  
 + cd /var/lib/jenkins/workspace/clone-job  
 + pip3 install -r requirements.txt  
 Defaulting to user installation because normal site-packages is not writeable  
 Collecting click==8.1.3  
 Downloading click-8.1.3-py3-none-any.whl (96 kB)  
 Collecting requests==2.26.0  
 Downloading requests-2.26.0-py2.py3-none-any.whl (62 kB)  
 Collecting Flask==2.2.2  
 Downloading Flask-2.2.2-py3-none-any.whl (101 kB)  
 Collecting importlib-metadata==4.12.0  
 Downloading importlib\_metadata-4.12.0-py3-none-any.whl (21 kB)  
 Collecting itsdangerous==2.1.2  
 Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)  
 Collecting Jinja2==3.1.2  
 Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)  
 Collecting joblib==1.1.0  
 Downloading joblib-1.1.0-py2.py3-none-any.whl (306 kB)  
 Collecting MarkupSafe==2.1.1  
 Downloading MarkupSafe-2.1.1-cp37-cp37m-manylinux\_2\_17\_x86\_64.manylinux2014\_x86\_64.whl (25 kB)  
 Collecting numpy==1.21.6

Not secure 34.229.97.233:8080/job/clone-job/

# Jenkins

Dashboard > clone-job >

Status clone-job

</> Changes

Workspace

Build Now

Configure

Delete Project

Rename

Build History trend

Filter... /

#2 Mar 13, 2024, 5:07 PM

#1 Mar 13, 2024, 5:04 PM

Atom feed for all Atom feed for failures

Search (CTRL+K) ?

1 ramakrishna log out

Add description

Disable Project

## Downstream Projects

build-job

## Permalinks

- Last build (#2), 18 min ago
- Last stable build (#2), 18 min ago
- Last successful build (#2), 18 min ago
- Last failed build (#1), 21 min ago
- Last unsuccessful build (#1), 21 min ago
- Last completed build (#2), 18 min ago

The screenshot shows the Jenkins interface for the 'clone-job' project. The top navigation bar includes links for 'Dashboard', 'clone-job', 'Status', 'Changes', 'Workspace', 'Build Now', 'Configure', 'Delete Project', and 'Rename'. On the right, there are buttons for 'Add description' and 'Disable Project'. The main content area displays the job name 'clone-job' with a green checkmark icon. Below it is a section titled 'Downstream Projects' with a link to 'build-job'. A 'Permalinks' section lists several recent builds with their status and timestamps. To the left, a 'Build History' sidebar shows two builds: '#2' (Mar 13, 2024, 5:07 PM) which is successful, and '#1' (Mar 13, 2024, 5:04 PM) which is failed. At the bottom, there are links for 'Atom feed for all' and 'Atom feed for failures'.

Copy the ip of instance and browse it in the browser

# BUILD PERIODICALLY

Not secure 34.229.97.233:8080/job/build-job/configure

Dashboard > build-job > Configuration

## Configure

- General
- Source Code Management
- Build Triggers**
- Build Environment
- Build Steps
- Post-build Actions

**Build after other projects are built** Projects to watch: clone-job

Trigger only if build is stable

Trigger even if the build is unstable

Trigger even if the build fails

Always trigger, even if the build is aborted

**Build periodically** ? Schedule: \* \* \* \* \* **No schedules so will never run**

GitHub hook trigger for GITScm polling ?

Poll SCM ?

**Save** **Apply**

Not secure 34.229.97.233:8080/job/build-job/

Jenkins

Dashboard > build-job >

Status: **build-job** build-job

Changes

Workspace

Build Now

Configure

Delete Project

Rename

**Upstream Projects**

clone-job

**Permalinks**

- Last build (#5), 56 min ago
- Last stable build (#5), 56 min ago
- Last successful build (#5), 56 min ago
- Last failed build (#4), 1 hr 1 min ago
- Last unsuccessful build (#4), 1 hr 1 min ago
- Last completed build (#5), 56 min ago

Build History trend

Filter...

| #  | Date                  |
|----|-----------------------|
| #8 | Mar 13, 2024, 6:14 PM |
| #7 | Mar 13, 2024, 6:13 PM |
| #6 | Mar 13, 2024, 6:13 PM |

# POLL SCM

The screenshot shows the Jenkins 'Configure' screen for a build job. The left sidebar lists several configuration sections: General, Source Code Management, Build Triggers, Build Environment, Build Steps, and Post-build Actions. The 'Build Triggers' section is currently active, indicated by a highlighted background. Within this section, there are several options:

- Trigger even if the build is unstable
- Trigger even if the build fails
- Always trigger, even if the build is aborted
- Build periodically ?
- GitHub hook trigger for GITScm polling ?
- Poll SCM ?

Below the triggers, there is a 'Schedule' section containing a text input field with the value '\*\*\*\*\*'. A note below the schedule says: "No schedules so will only run due to SCM changes if triggered by a post-commit hook". There is also an unchecked checkbox for "Ignore post-commit hooks ?".

At the bottom of the configuration page, there are two buttons: 'Save' and 'Apply'.

github.com/rama861/python-fish/tree/master

Code Issues Pull requests Projects Wiki Security Insights Settings

master / python-fish /

rama861 Create identify c686213 · now History

| Name                     | Last commit message     | Last commit date |
|--------------------------|-------------------------|------------------|
| fish                     | fishweight              | 2 years ago      |
| templates                | Update main.html        | 2 years ago      |
| Fish market.ipynb        | fishweight              | 2 years ago      |
| Fish.csv                 | fishweight              | 2 years ago      |
| Procfile                 | fishweight              | 2 years ago      |
| Random.pkl               | fishweight              | 2 years ago      |
| app.py                   | Update app.py           | last week        |
| gitattributes            | fishweight              | 2 years ago      |
| <a href="#">identify</a> | Create identify         | now              |
| one_joblib               | fishweight              | 2 years ago      |
| requirements.txt         | Update requirements.txt | last week        |

Not secure 34.229.97.233:8080/job/build-job/

Dashboard > build-job >

Changes workspace build-job

Build Now

Configure

Delete Project

Polling Log

Rename

Build History trend

Filter... /

#17 Mar 13, 2024, 6:22 PM ↑

#16 Mar 13, 2024, 6:21 PM ↑

#15 Mar 13, 2024, 6:21 PM ↓

#14 Mar 13, 2024, 6:21 PM

## Upstream Projects

clone-job

## Permalinks

- Last build (#13), 3 min 25 sec ago
- Last stable build (#13), 3 min 25 sec ago
- Last successful build (#13), 3 min 25 sec ago
- Last failed build (#4), 1 hr 9 min ago
- Last unsuccessful build (#4), 1 hr 9 min ago
- Last completed build (#13), 3 min 25 sec ago

# WEBHOOKS

Not secure 34.229.97.233:8080/job/build-job/configure

Dashboard > build-job > Configuration

Git ?

## Configure

General Source Code Management Build Triggers Build Environment Build Steps Post-build Actions

### Build Triggers

Trigger builds remotely (e.g., from scripts) ?  Build after other projects are built ?

Projects to watch  
clone-job

Trigger only if build is stable  
 Trigger even if the build is unstable  
 Trigger even if the build fails  
 Always trigger, even if the build is aborted

Build periodically ?  GitHub hook trigger for GITScm polling ?  Poll SCM ?

Save Apply

github.com/rama861/python-fish/settings rama861 / python-fish

Code Issues Pull requests Projects Wiki Security Insights Settings

General

Repository name python-fish Rename

Template repository Template repositories let users generate new repositories with the same directory structure and files. Learn more

Require contributors to sign off on web-based commits Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Sig contributors to affirm that their commit complies with the repository's terms, commonly the [Developer Certificate of Origin](#). Learn more about signing off on commits.

Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions Webhooks Environments Codespaces Pages

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests are automatically made, unless you specify a different branch.  
master

Social preview

Upload an image to customize your repository's social media preview. Images should be at least 640x320px (1280x640px for best display). Download template

 General

Access

 Collaborators Moderation options

Code and automation

 Branches Tags Rules Actions Webhooks Environments Codespaces Pages

Security

 Code security and analysis Deploy keys Secrets and variables

Integrations

 GitHub Apps Email notifications

## Webhooks / Add webhook

We'll send a POST request to the URL below with details of any subscribed events. You can also specify which data format you'd like to receive (JSON, x-www-form-urlencoded, etc). More information can be found in [our developer documentation](#).

## Payload URL \*

## Content type



## Secret

## Which events would you like to trigger this webhook?

- Just the push event.
- Send me everything.
- Let me select individual events.

 Active

We will deliver event details when this hook is triggered.

**Add webhook**

github.com/rama861/python-fish/settings/hooks

rama861 / python-fish

Type  to search

Code Issues Pull requests Projects Wiki Security Insights Settings

General Webhooks Add webhook

Access Collaborators Moderation options

Code and automation Branches Tags Rules Actions

Webhooks Environments Codespaces Pages

Code security and analysis Deploy keys Secrets and variables

Webhooks

Webhooks allow external services to be notified when certain events happen. When the specified events happen, we'll send a POST request to each of the URLs you provide. Learn more in our [Webhooks Guide](#).

✓ <http://34.229.97.233:8080/github-w...> (all events) Edit Delete

github.com/rama861/python-fish/tree/master

Code Issues Pull requests Projects Wiki Security Insights Settings

master python-fish /

rama861 Create hlo.txt 617bf09 · 1 minute ago History

| Name                    | Last commit message     | Last commit date |
|-------------------------|-------------------------|------------------|
| fish                    | fishweight              | 2 years ago      |
| templates               | Update main.html        | 2 years ago      |
| Fish market.ipynb       | fishweight              | 2 years ago      |
| Fish.csv                | fishweight              | 2 years ago      |
| Profile                 | fishweight              | 2 years ago      |
| Random.pkl              | fishweight              | 2 years ago      |
| app.py                  | Update app.py           | last week        |
| gitattributes           | fishweight              | 2 years ago      |
| <a href="#">hlo.txt</a> | Create hlo.txt          | 1 minute ago     |
| identify                | Create identify         | 14 minutes ago   |
| one_joblib              | fishweight              | 2 years ago      |
| requirements.txt        | Update requirements.txt | last week        |

<https://github.com/rama861/python-fish/blob/master/hlo.txt>

Not secure 34.229.97.233:8080/job/build-job/

# Jenkins

Search (CTRL+K) ? 1 ramakrishna log out

Dashboard > build-job >

Status build-job

</> Changes build-job

Workspace

Build Now

Configure

Delete Project

GitHub Hook Log

Rename

Build History trend Filter... /

#21 Mar 13, 2024, 6:34 PM

#20 Mar 13, 2024, 6:34 PM

#19

Upstream Projects

clone-job

Permalinks

- Last build (#18), 9 min 54 sec ago
- Last stable build (#18), 9 min 54 sec ago
- Last successful build (#18), 9 min 54 sec ago
- Last failed build (#4), 1 hr 21 min ago
- Last unsuccessful build (#4), 1 hr 21 min ago
- Last completed build (#18), 9 min 54 sec ago

Edit description Disable Project

After complete the process next copy the public ip of an instance and browse it in the browser



# FISH WEIGHT

**Species \***

Bream

**Vertical length in cm \***

23.2

**Diagonal length in cm \***

25.4

**Cross length in cm \***

30.0

**Height \***

11.5200

**Width \***

4.0200

**SUBMIT**