**PROJECT-3**

**DEPLOYMENT OF PYTHON APPLICATION**

**Module-1:** Manual Deployment

**Module-2:** Deployment Through Terraform Script

**Module-3:** Automation Deployment Through Jenkins

**Module-4:** Deployment Through Jenkins by Using Terraform Script

**Module-5:** Automation Deployment Using Webhooks, POLSCM and Build Periodically

**What is Webhook?**

A webhook enables third-party services like GitHub to send real-time updates to an application. Updates are triggered by an event or an action by the webhook provider like push to a repository, any commit to a code or pull request creation.

**What is POLLSCM?**

Poll SCM polls the SCM periodically for checking if any changes or any new commits were made to the code and shall build the project if any new commits were pushed since the last build.

**What is Build Periodically?**

Build Periodically is process where we can set a time so that after every desired time the project will build automatically whether new changes or commits are happened or not.

**Tools used in this project are: -**

* Git
* Jenkins
* Terraform

**Repositories: -**

1. <https://github.com/Ajlaluddin/car-prediction.git>
2. <https://github.com/Ajlaluddin/Tweet-Detection.git>
3. <https://github.com/Ajlaluddin/fuel-prediction.git>

**MODULE-1**

**MANUAL DEPLOYMENT**

Deploying python application manually by using AWS resources.

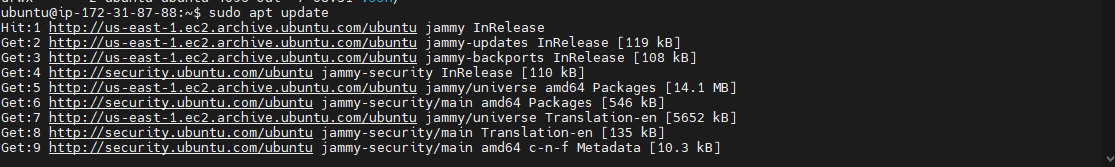
**Step-1: -** Launch a EC2 Instance (ubuntu) by adding ports 22,80&8080 in security group.

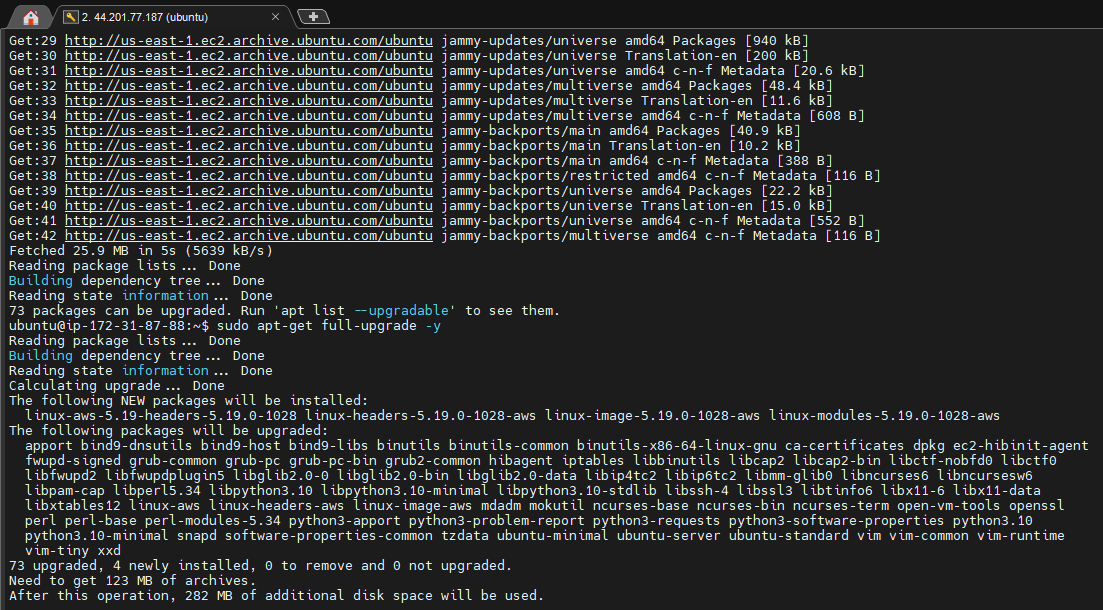
**Step-2: -** Connect to the Instance with SSH key.

**Step-3: -** Update your ubuntu machine with following commands

Commands: Sudo apt update

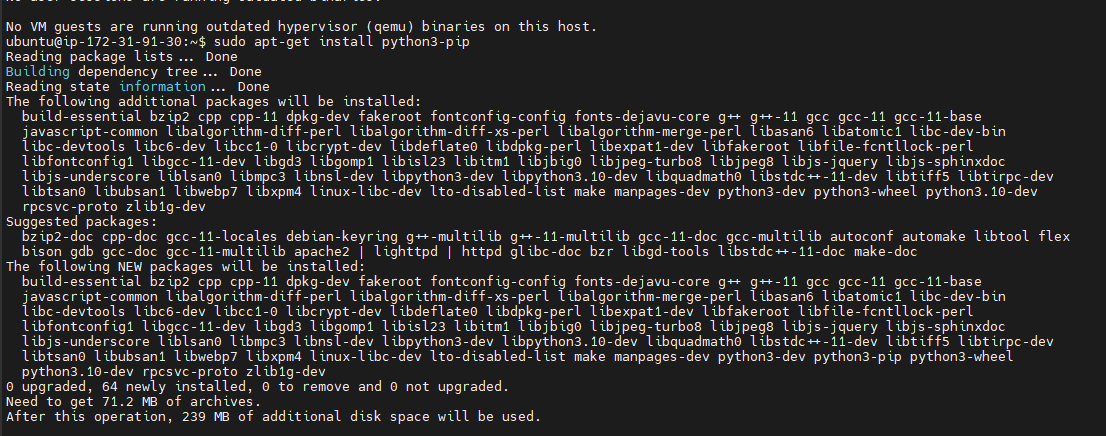
Sudo apt-get full upgrade -y





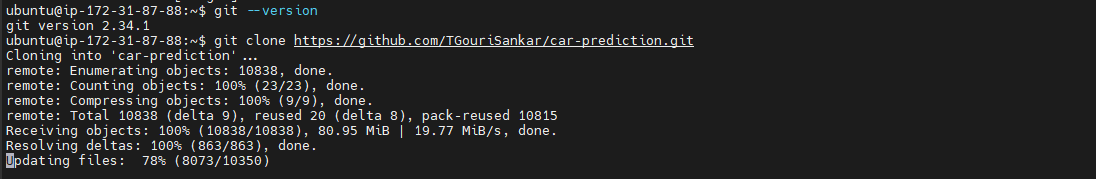
**Step-4: -** Install required package/tools that required for project deployment.

Command: Sudo apt-get install python3-pip



**Step-5: -** Clone the project code from the git hub into your machine.

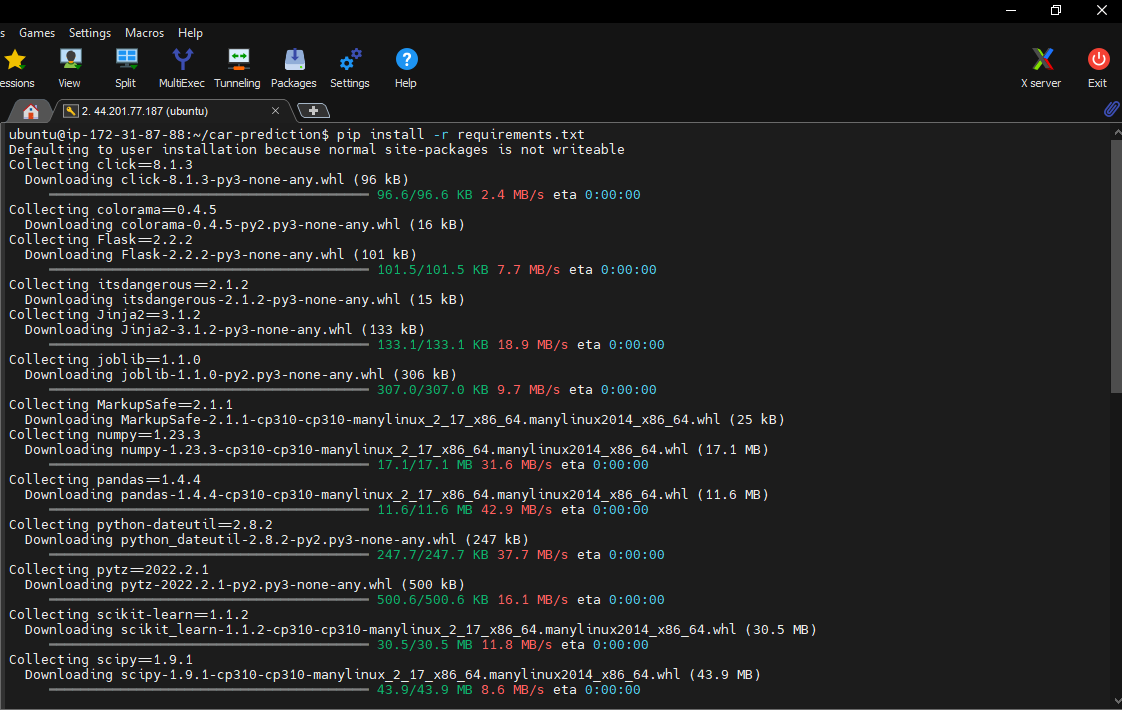
Command: Git clone <https://github.com/Ajlaluddin/car-prediction.git>



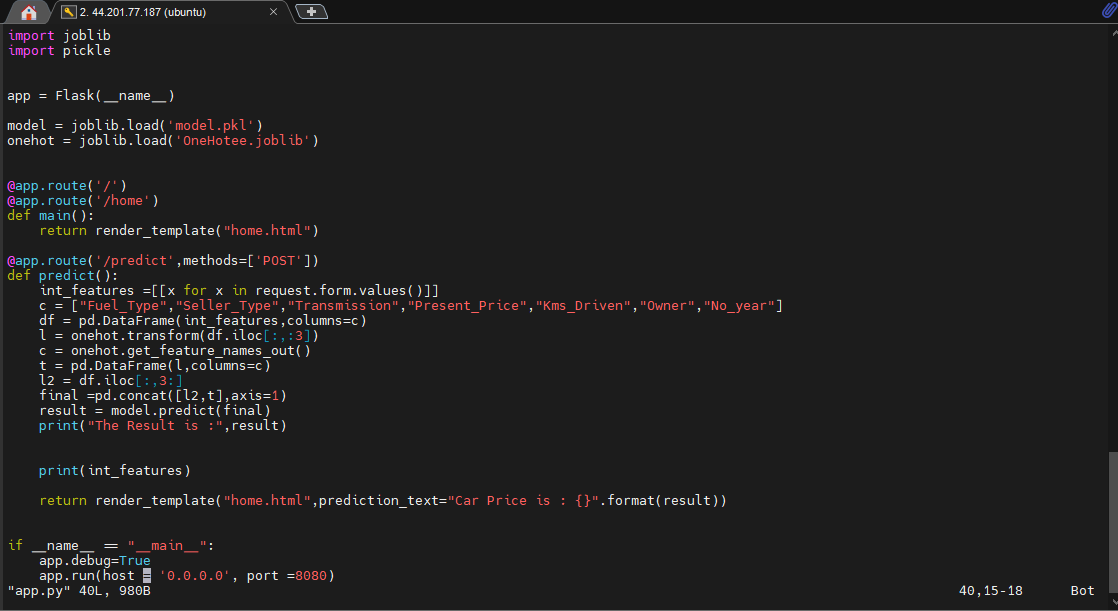
**Step-6: -** Go to the source directory by using cd command.

**Step-7: -** Install requirements package

Command: pip install -r requirements.txt

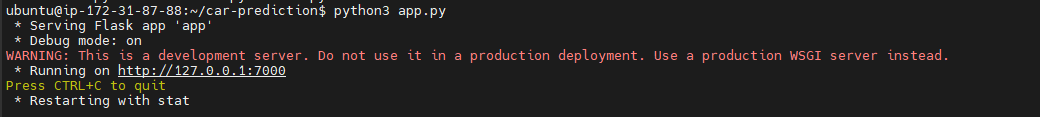


**Step-8: -** Change app.py file with vi command. Add 0.0.0.0 host to host from anywhere with 8080 port.



**Step-9: -** Run the flask server by executing below command

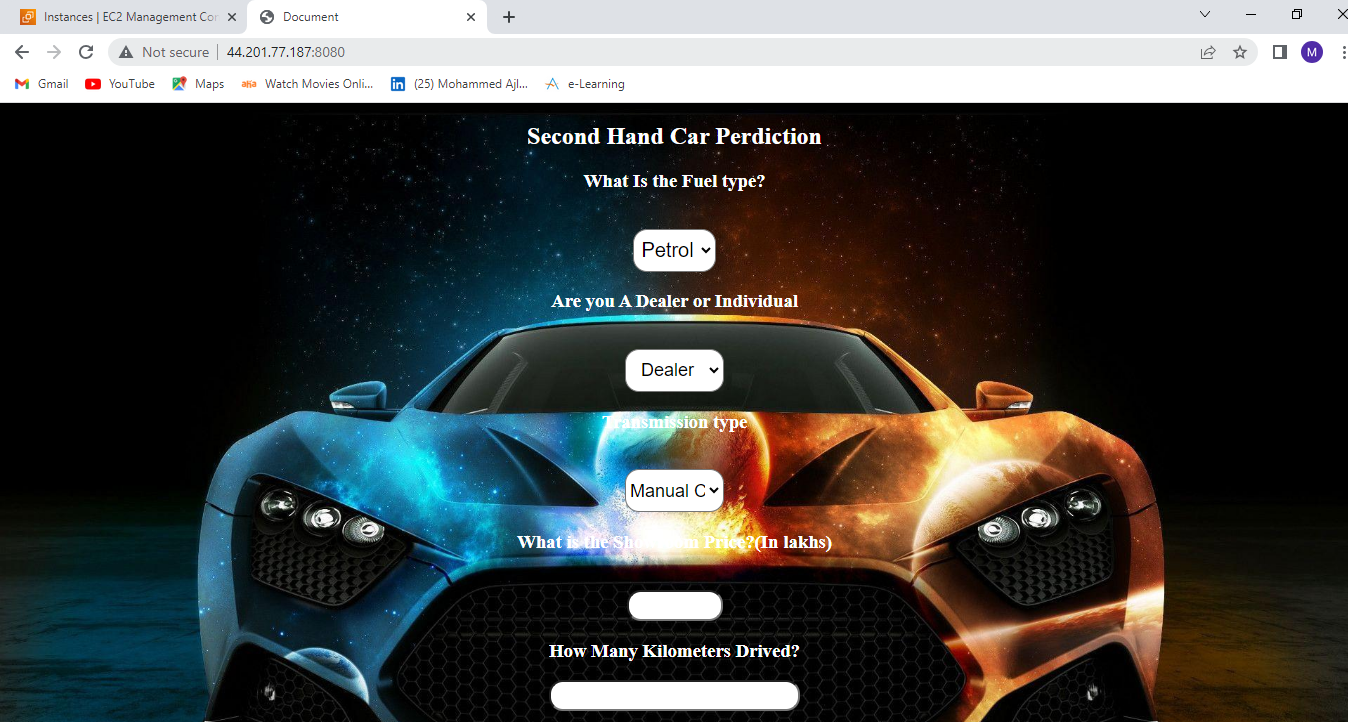
Command: python3 app.py



**Step-10: -** Execute Screen -m -d python3 app.py to run application continuously.

**Step-11: -** Browse the public Ip of instance with given port to check output.

**OUTPUT: -**



**MODULE-2**

**DEPLOYMENT THROUGH JENKINS**

**Step-1: -** Launch a EC2 Instance (ubuntu) by adding ports 22,80&8080 in security group.

**Step-2: -** Connect to the Instance with SSH key.

**Step-3: -** Install git in the instance.

Command: Sudo yum install git -y

**Step-4: -** Install Jenkins in the instance.

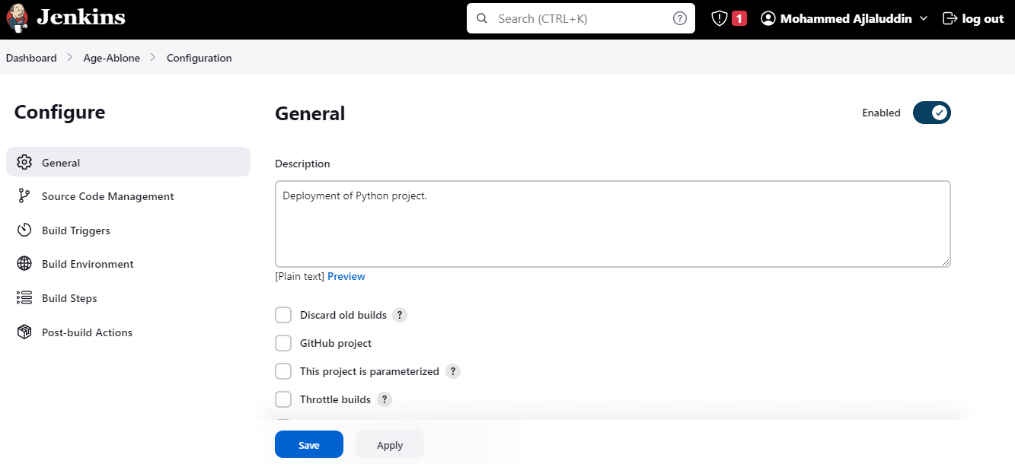
Command:

* + - sudo yum update –y
    - 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.key>
    - sudo yum upgrade
    - sudo amazon-linux-extras install java-openjdk11 -y
    - sudo yum install jenkins -y
    - sudo systemctl enable Jenkins
    - sudo systemctl start Jenkins
    - sudo systemctl status Jenkins

**Step-5: -** Connect to the Jenkins server by browsing public Ip of instance with 8080 port.

**Step-6: -** Setup Jenkins account by adding credentials.

**Step-7: -** Create a new job by selecting freestyle project and provide project description.



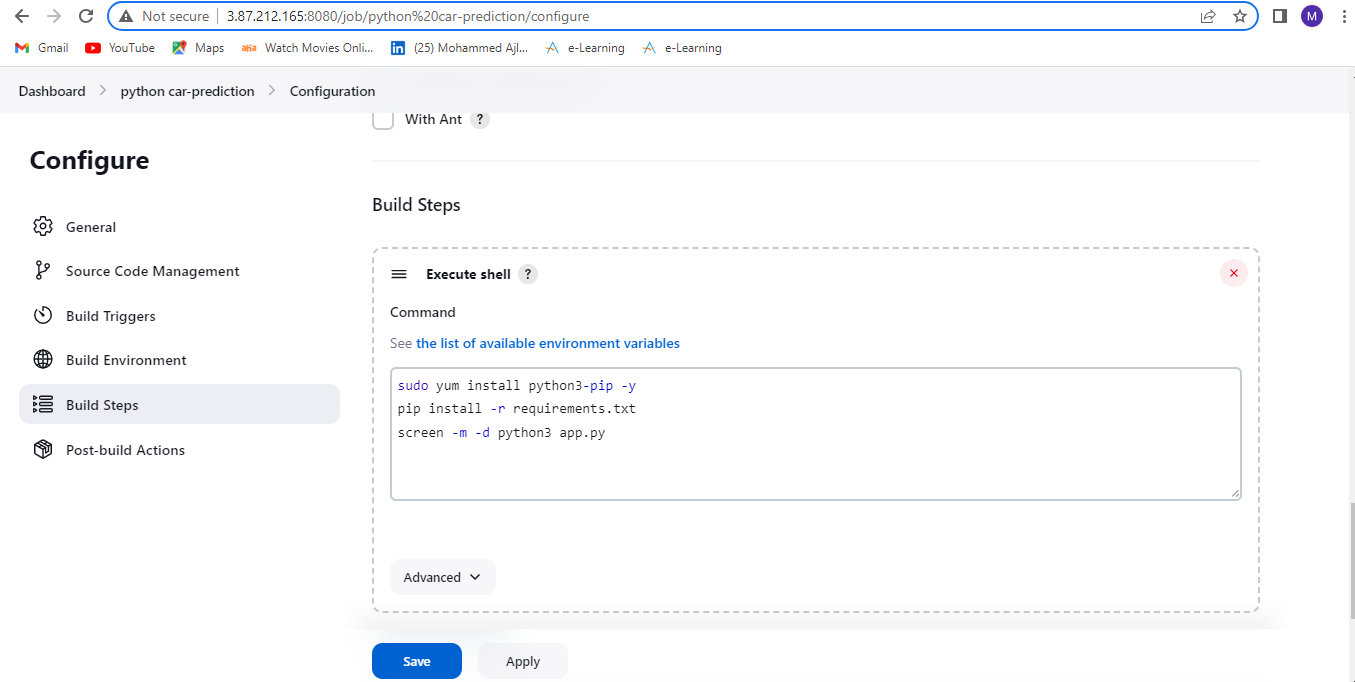
**Step-8: -** Select “Git” as a source code repository. Provide repo URL and credentials.



**Step-9: -** Select “Execute shell” in Build Steps and write commands.

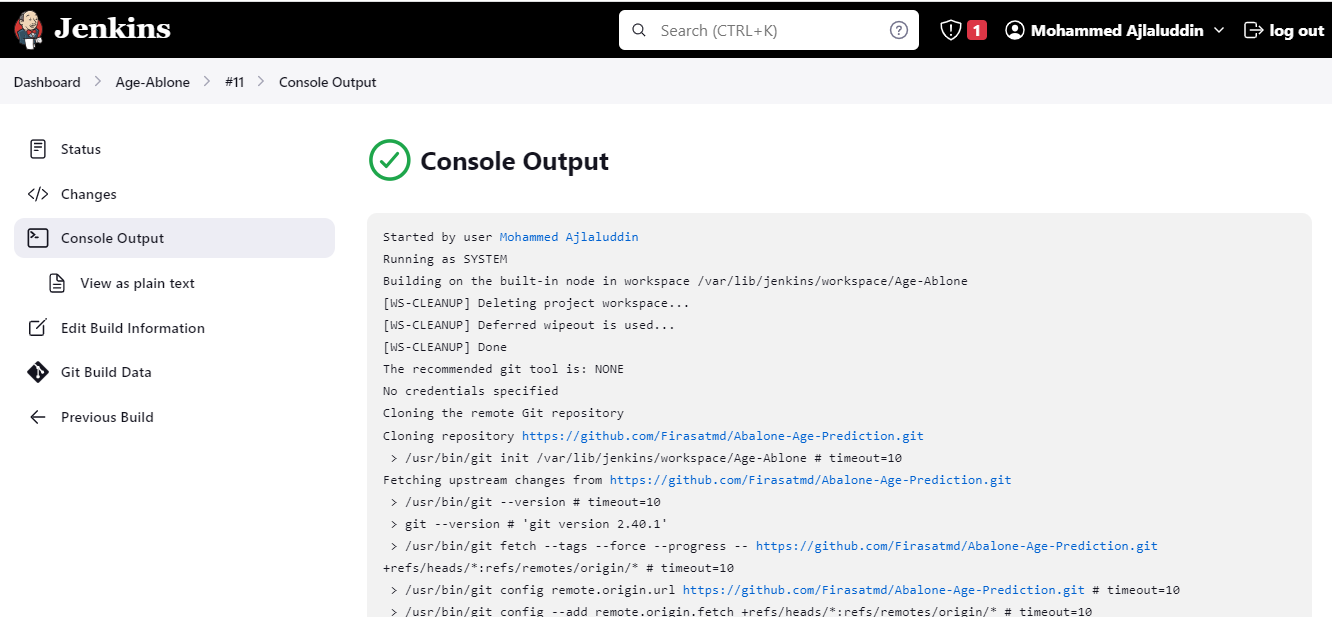
Commands:

* + - Sudo yum install python3-pip -y
    - Pip install -r requirements.txt
    - Screen -m -d python3 app.py

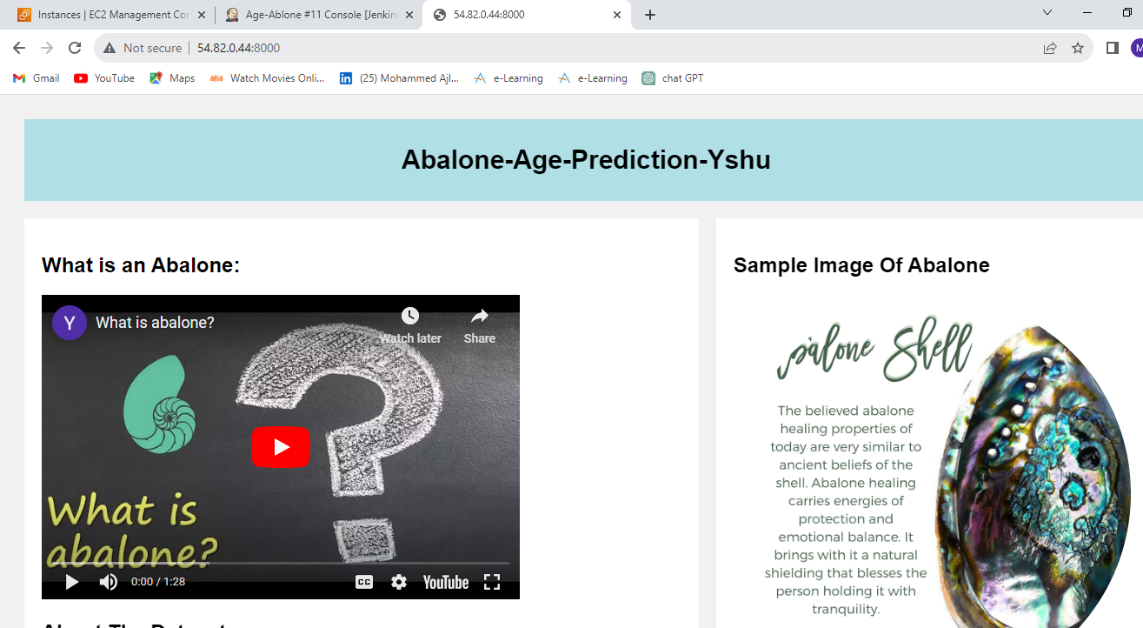


**Step-10: -** Apply and save the project configurations.

**Step-11: -** Build the project by clicking “Build Now”.



**Step-12: -** Check the output by browsing the public Ip of instance with provided port.



**MODULE-3**

**DEPLOYMENT THROUGH TERRAFORM**

**Step-1: -** Launch EC2 instance and connect to the instance with SSH key.

**Step-2: -** Install Terraform in the instance.

Commands:

* + - 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

**Step-3: -** Create a Directory with mkdir command and go inside the directory.

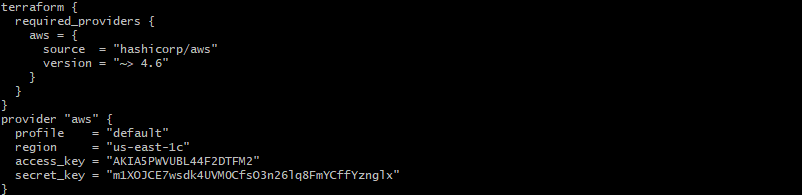
**Step-4: -** Initialize directory with terraform Init command.

**Step-5: -** Create a provider file by using touch command with .tf extension.

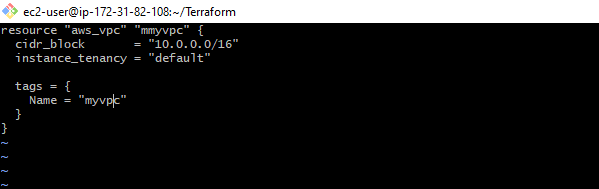
Commands: touch provider.tf

**Step-6: -** provide account credentials in provider.tf file by using vi command.

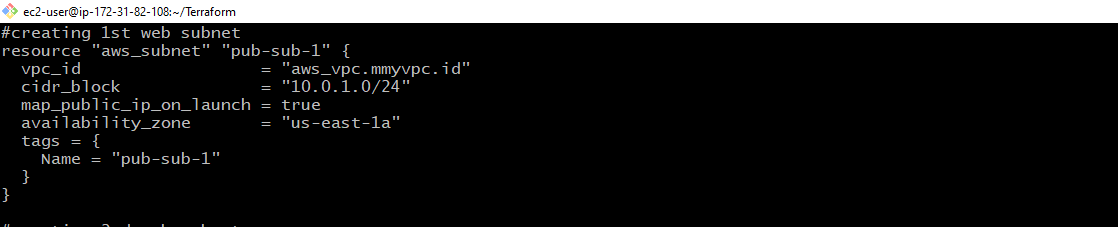
Command: vi provider.tf



**Step-7: -** Create a file for VPC (vpc.tf).

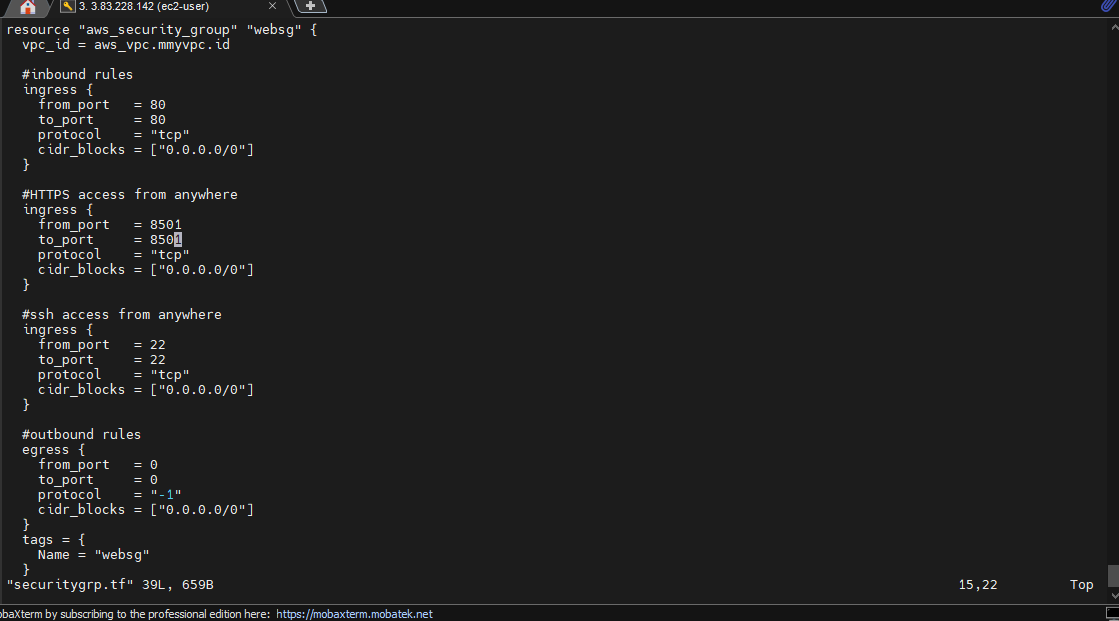


**Step-8: -** Create a file for subnet (subnet.tf).

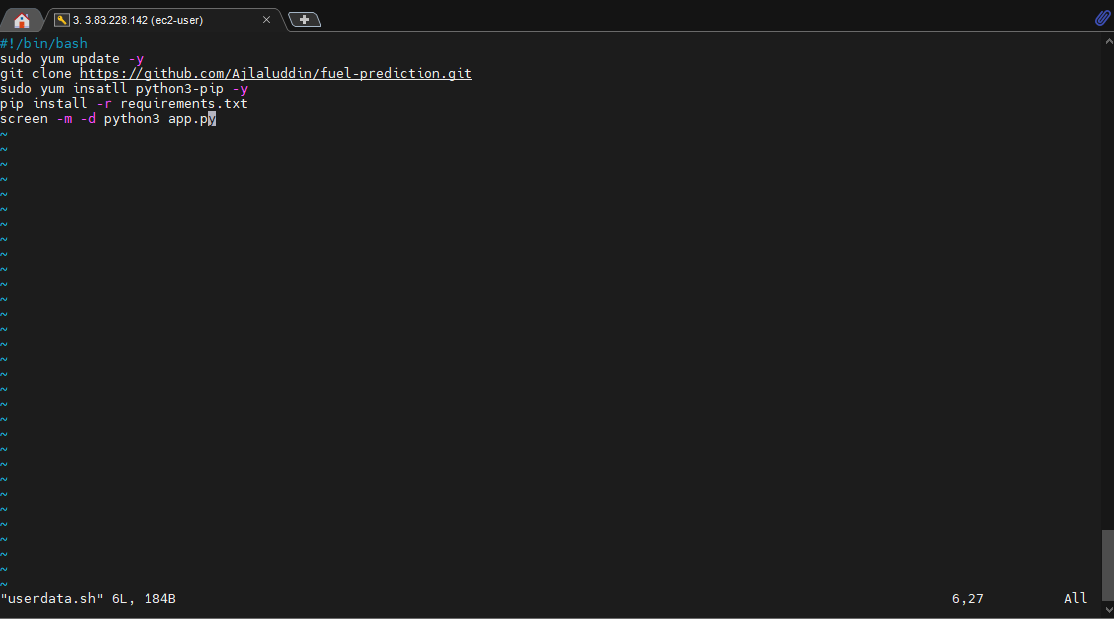


**Step-9: -** Create a file for Route Table (routetable.tf).

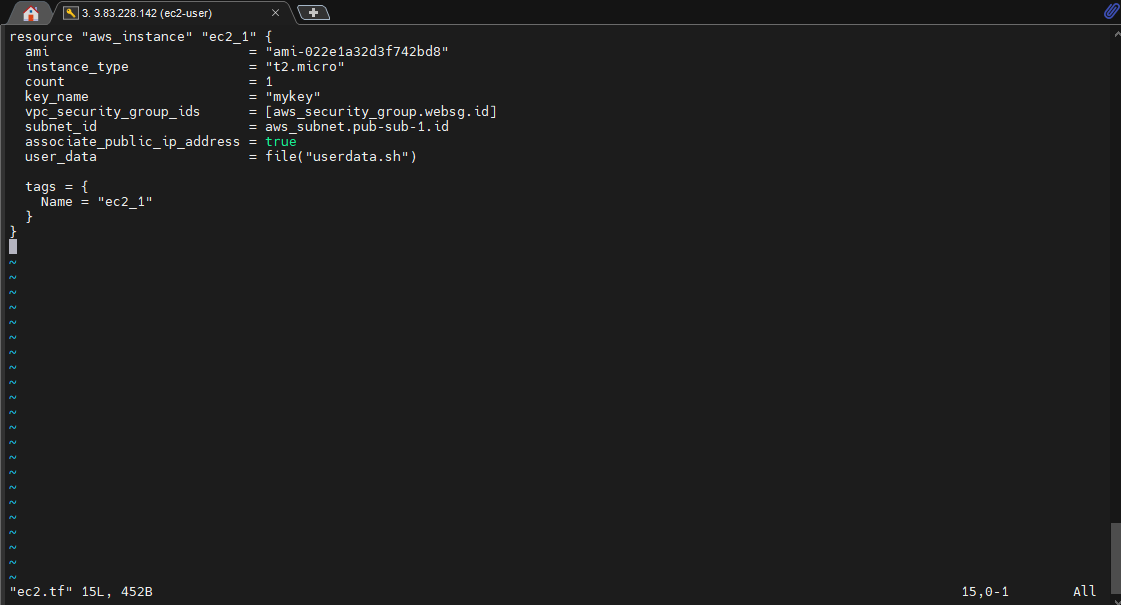
**Step-10: -** Create a file security group (securitygrp.tf).



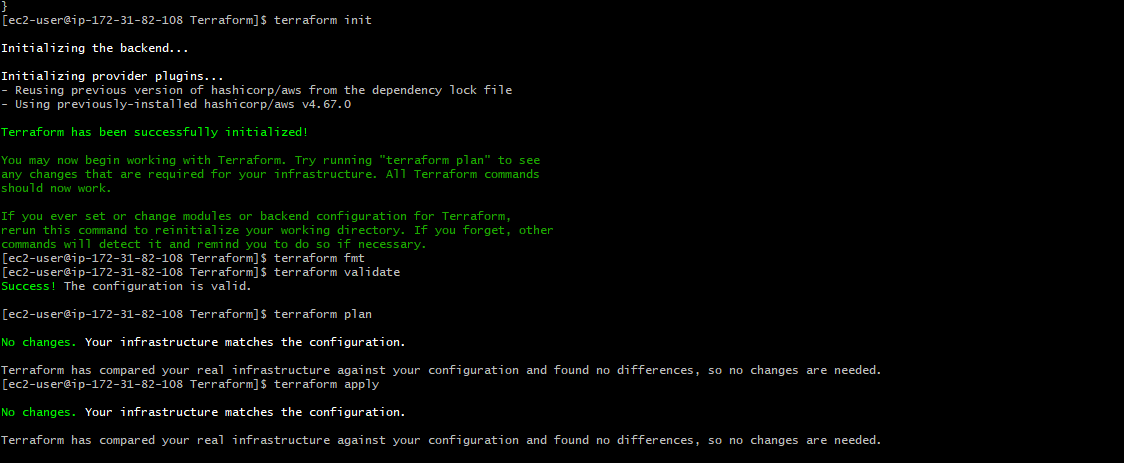
**Step-11: -** Create a file for user data with .sh extension (userdata.sh).



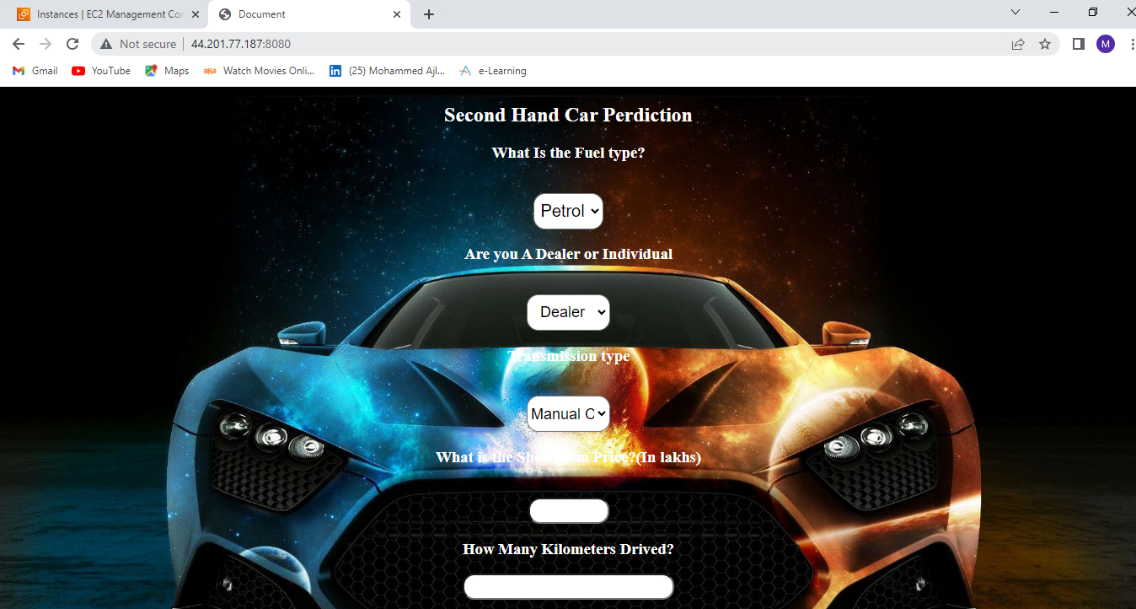
**Step-12: -** Create a file for AWS EC2 instance(ec2.tf) and write a script for EC2 Instance.



**Step-13: -** Execute terraform fmt, terraform validate, terraform plan and terraform apply command.



**Step-14: -** Check the output by browsing public Ip of instance with given port.



**MODULE-4**

**DEPLOYMENT THROUGH JENKINS USING TERRAFORM SCRIPT**

Deploying python web application through Jenkins by using terraform script.

Create data.sh file and give below commands:

#!bin/bash

Sudo yum install git -y

Sudo yum install -y yum-utils

Sudo yum-config-manager --add-repo <https://rpm.releases.hashicorp.com/AmazonLinux/hashicorp.repo>

Sudo yum install terraform -y

Sudo git clone <https://github.com/Ajlaluddin/python-infra.git>

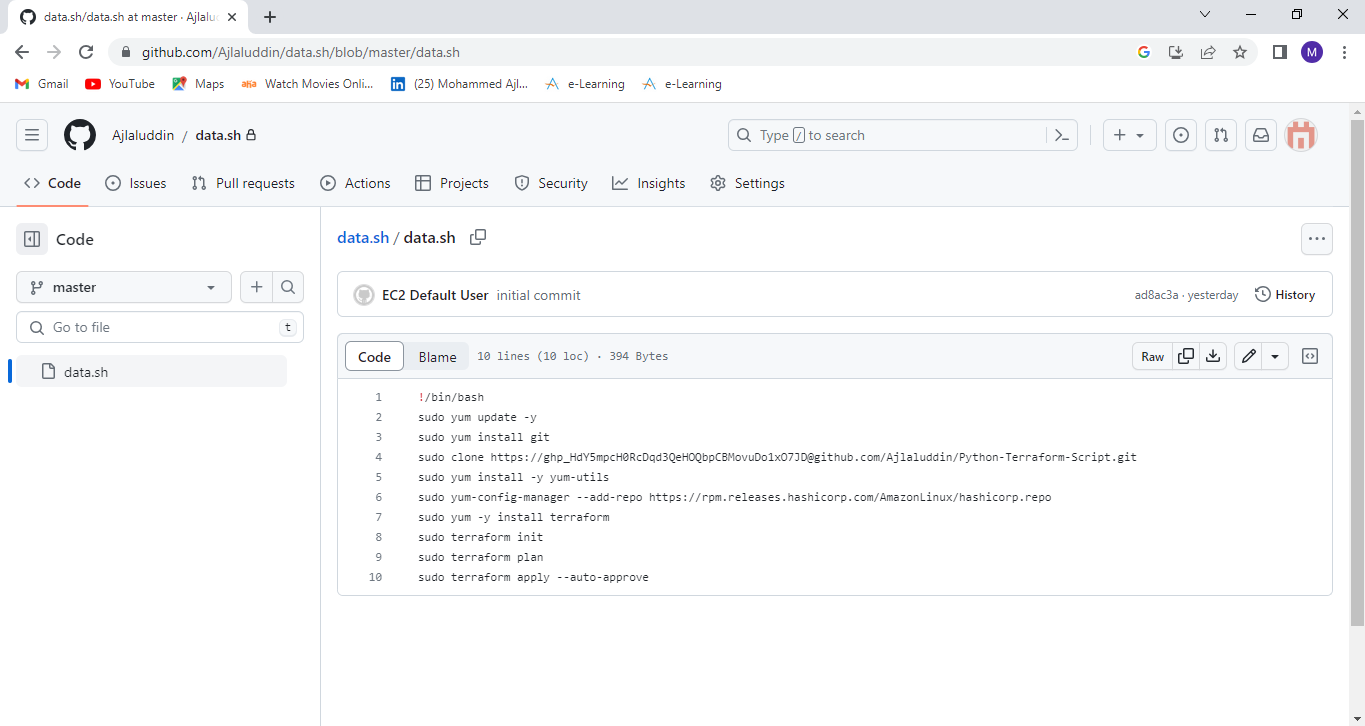
cd car-prediction

Terraform init

Terraform plan

Terraform apply –auto-approve

* Push data.sh file into repository.



**Step-1: -** Launch a EC2 Instance by giving 8080 port and connect to the instance by using SSH key.

**Step-2: -** Install git in Instance.

Command: sudo yum install git -y

**Step-3: -** Install Jenkins in Instance by executing installation commands.

Command:

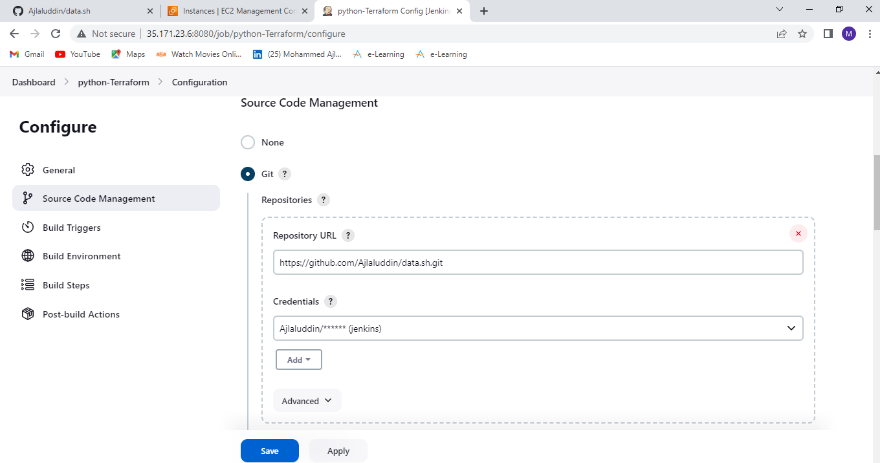
* + - sudo yum update –y
    - sudo wget -O /etc/yum.repos.d/jenkins.repo \  
      <https://pkg.jenkins.io/redhat-stable/jenkins.repo>
    - sudo rpm --import <https://pkg.jenkins.io/redhat-stable/jenkins.io.key>
    - sudo yum upgrade
    - sudo amazon-linux-extras install java-openjdk11 -y
    - sudo yum install jenkins -y
    - sudo systemctl enable Jenkins
    - sudo systemctl start Jenkins
    - sudo systemctl status Jenkins

**Step-4: -** Connect to the Jenkins server by browsing public Ip of instance with 8080 port.

**Step-5: -** Setup Jenkins account by adding credentials.

**Step-6: -** Create a new job by selecting freestyle project and provide project description.

**Step-7: -** Select “Git” as a source code repository. Provide data.sh repo URL and credentials.

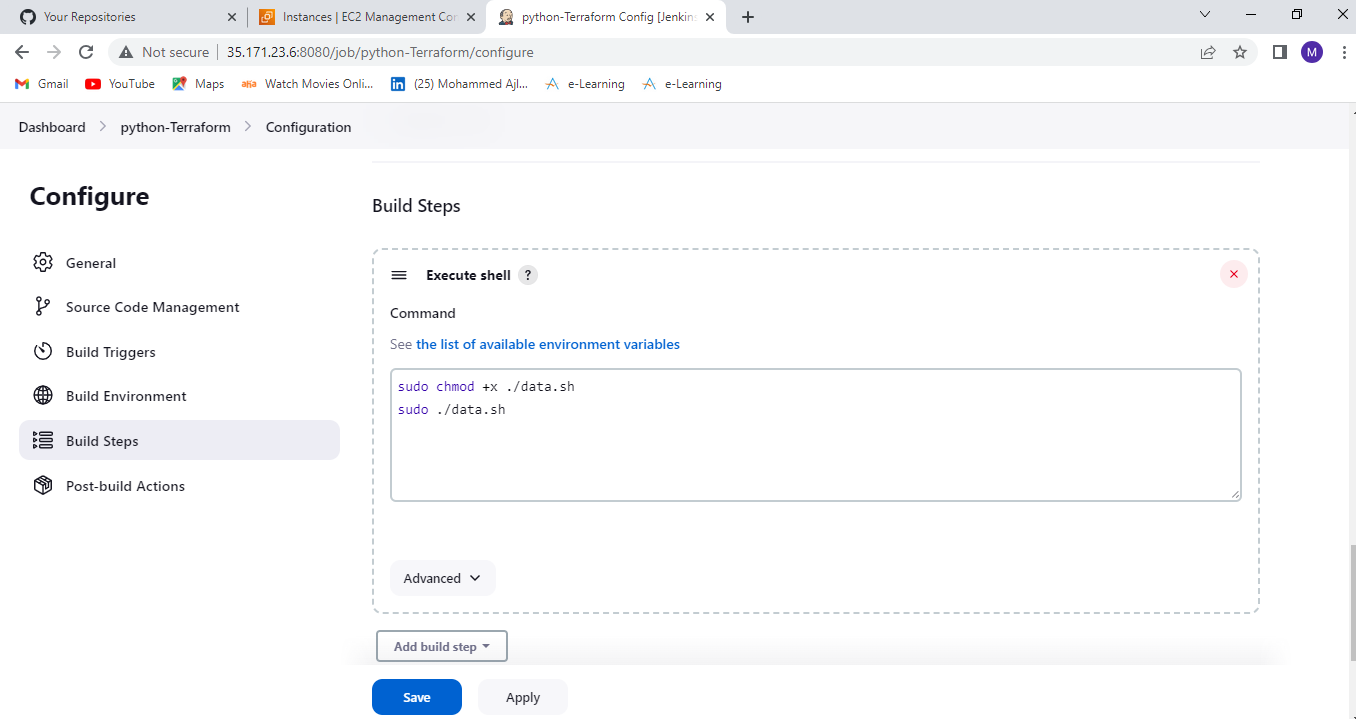


**Step-8: -** Select “Execute shell” in Build Steps and write commands.

Commands:

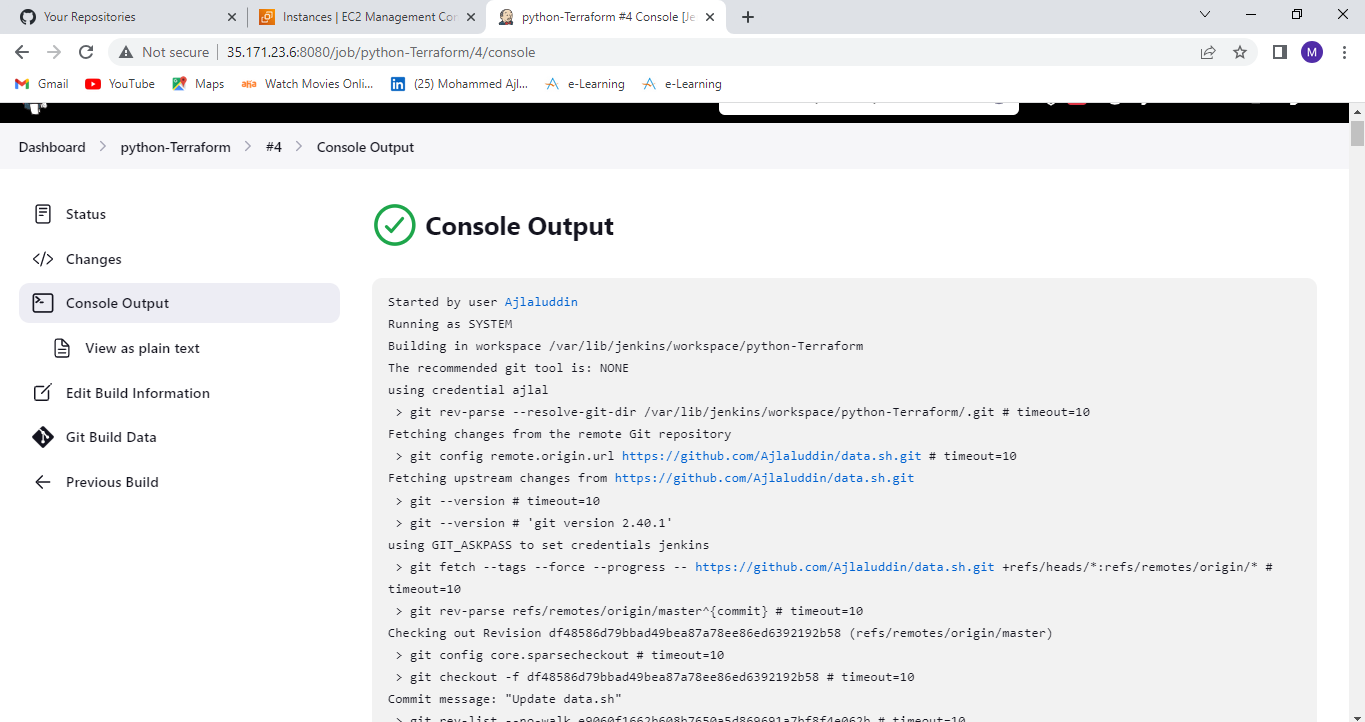
Sudo chmod +x ./data.sh

Sudo ./data.sh



**Step-9: -** Build the project by clicking “Build Now”.

**Step-10: -** Check the console output to see the output.



**Step-11: -** Browse public Ip of instance with provided port number to see the python web application.

