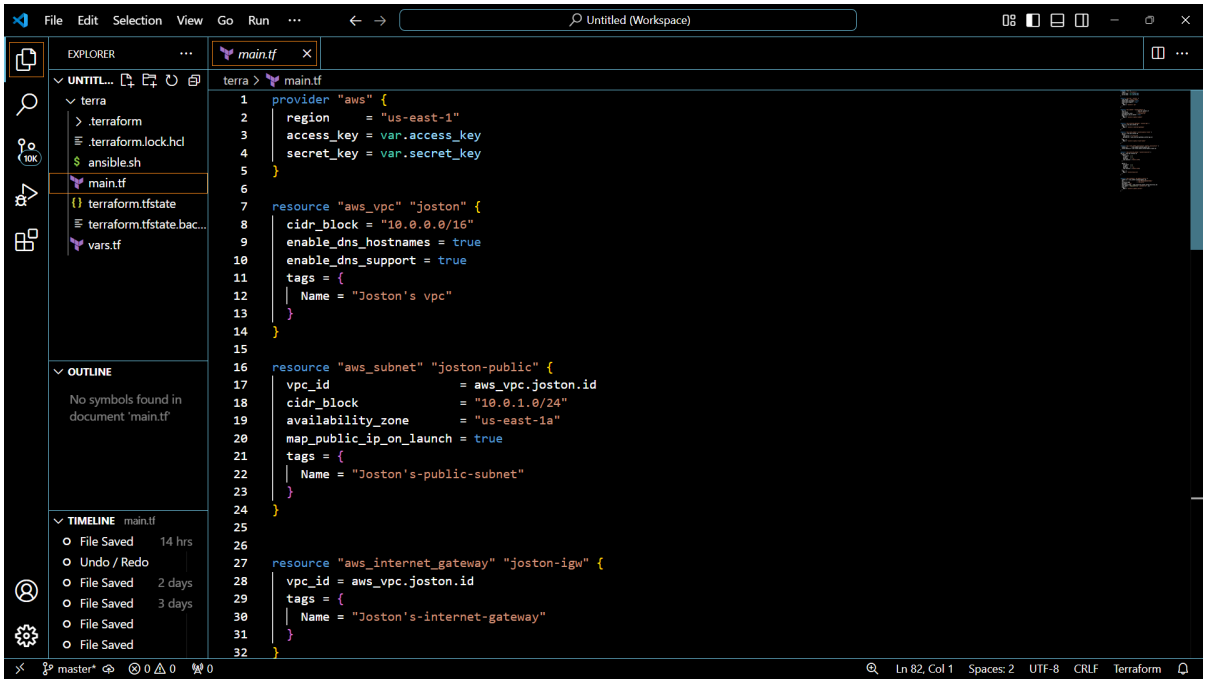
PROJECT-1

CERTIFICATION PROJECT – FINANCE ME

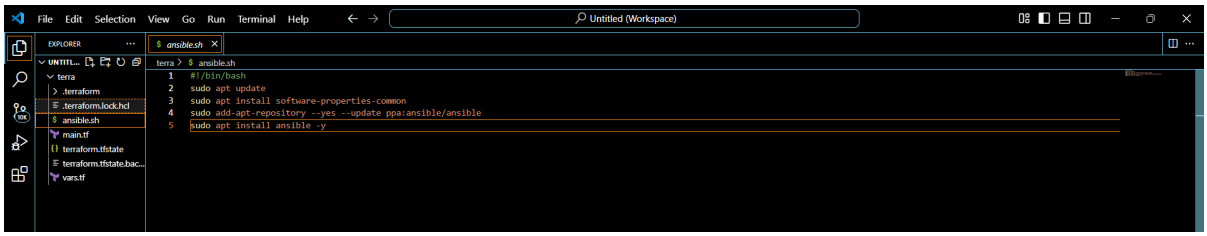
BANKING AND FINANCE DOMAIN

STEP-1

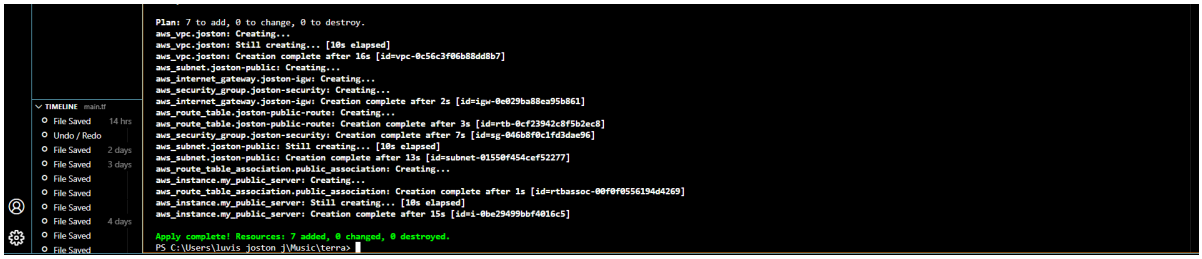
FIRST WRITE THE TERRAFORM FILE TO CREATE VPC,SUBNET,INTERNET GATEWAY AND ONE UNBUNTU INSTANCE AND INSTALL ASNIBLE ON THE INSTANCE.

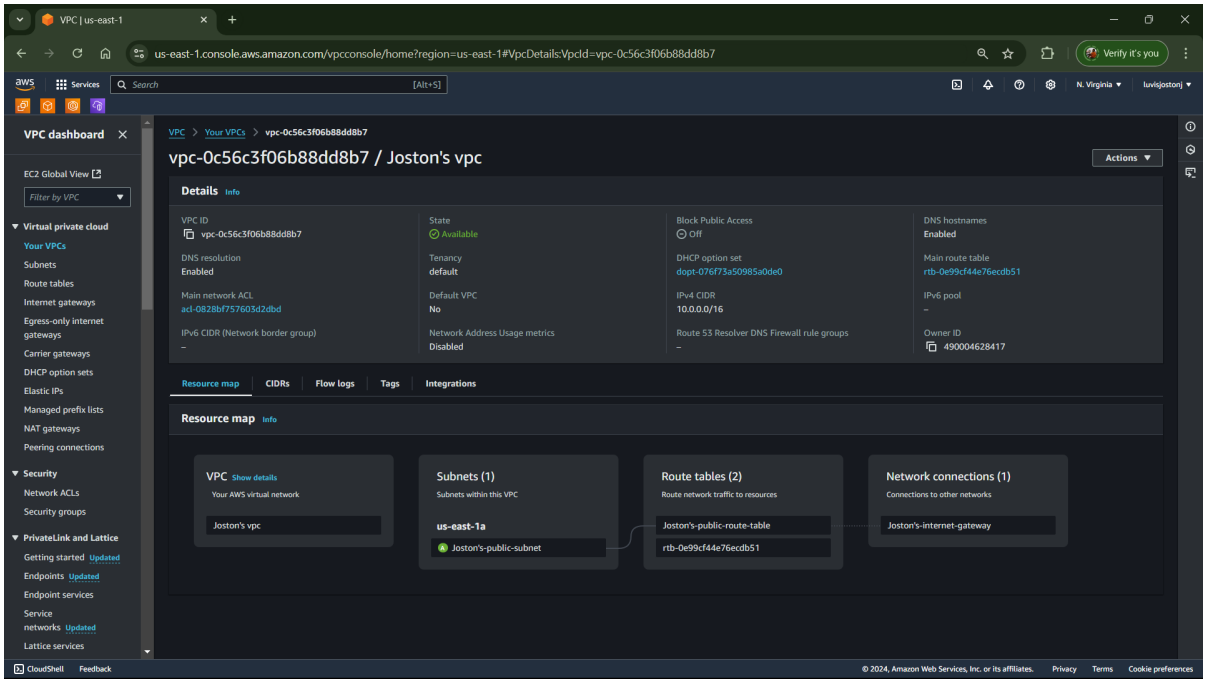


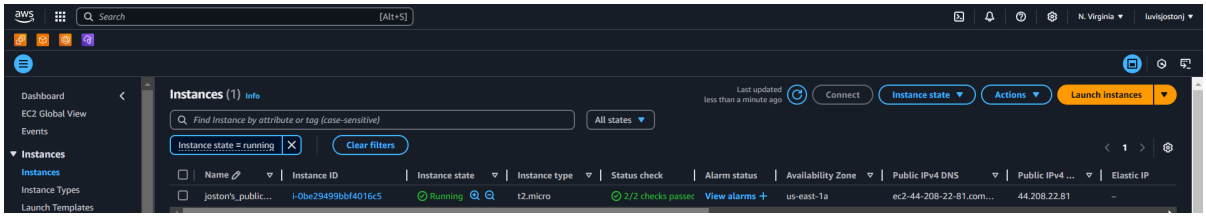
IM GONNA ATTACH TEMPLATEFILE FILE TO TERRAFORM FILE TO INSTALL ANSIBLE ON THE UBUNTU INSTANCE THAT WE GONNA CREATE USING TERRAFORM.



NOW APPLY, TERRAFORM FILE CREATE ENTIRE INFRASTRUCTURE FOR YOU IT WILL CREATE VPC,SUBNET AND ROUTE TABLES AND INTERNET GATEWAY AND AND ONE INSTANCE AND ATTACH SECURITY GROUP FOR IT AND INSTALL ANSIBLE ON IT.







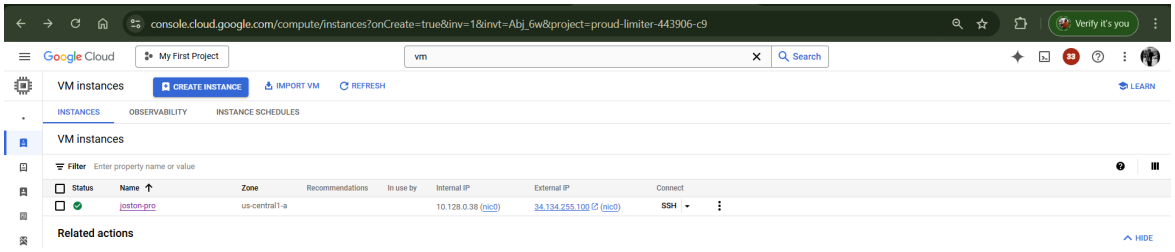
STEP-2

NOW CONNECT TO THAT INSTANCE THAT YOU CREATED FROM TERRAFORM AND YOU HAVE INSTALLED ANSIBLE ON IT.

ANSIBLE CONTROLER

```
root@ip-10-0-1-226:~# ansible --version
ansible [core 2.17.7]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Sep 11 2024, 14:17:37) [GCC 13.2.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@ip-10-0-1-226:~#
```

I HAVE CONNECT ONE UBUNTU INSTANCE FROM GCP AS A ANSIBLE WORKER VIA SSH.




```
joston@ip-10-0-1-226:~$ ansible -m ping joston
34.134.255.100 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.8"
    },
    "changed": false,
    "ping": "pong"
}
joston@ip-10-0-1-226:~$
```

START WTRING THE ANSIBLE PALYBOOK.YML FILE TO INSTALL JENKINS,DOCKER,GIT ,GIVE PERMISSION TO JENKINS USER TO RUN DOCKER COMMANDS AND START THE JENKINS AND DOCKER SERVICES.

To install jenkins im gonna use jenkins.sh file and im gonna execute this .sh file using playbook.

```
joston@ip-10-0-1-226:~$ vi jenkins.sh
```

```
sudo apt update
sudo apt install openjdk-17-jre-headless -y
sudo wget -O /usr/share/keyrings/jenkins-keyring.asc \
  https://pkg.jenkins.io/debian-stable/jenkins.io-2023.key
echo "deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc]" \
  https://pkg.jenkins.io/debian-stable binary/ | sudo tee \
  /etc/apt/sources.list.d/jenkins.list > /dev/null
sudo apt-get update
sudo apt-get install jenkins -y
~
~
~
~
~
```

```
- name: my playbook to do the project no-1
  hosts: joston
  become: true
  tasks:
    - name: update the packages
      command: apt update
    - name: copy the jenkins.sh file to ansible worker
      copy: src=/home/joston/jenkins.sh dest=/home/joston
    - name: installing jenkins using jenkins.sh file
      command: bash jenkins.sh
    - name: install docker and git on ansible worker
      package: name={{item}} state=present
      loop:
        - docker.io
        - git
    - name: start the jenkins and docker services
      service: name={{item}} state=started
      loop:
        - jenkins
        - docker
    - name: give permission jenkins user to run docker commads
      command: chmod 777 /var/run/docker.sock
```

STEP-3

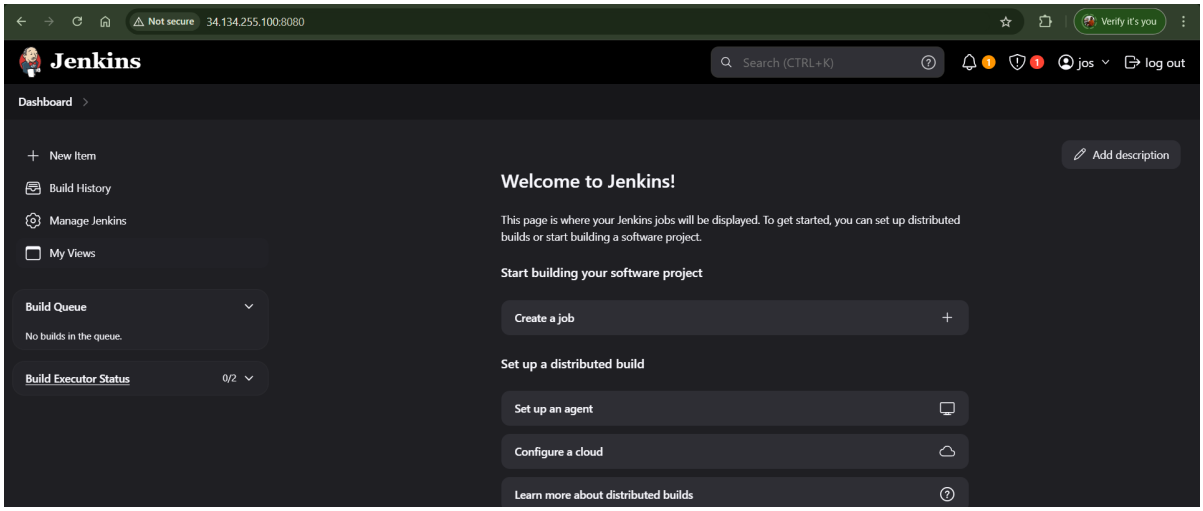
NOW RUN THE PLAYBOOK FILE.

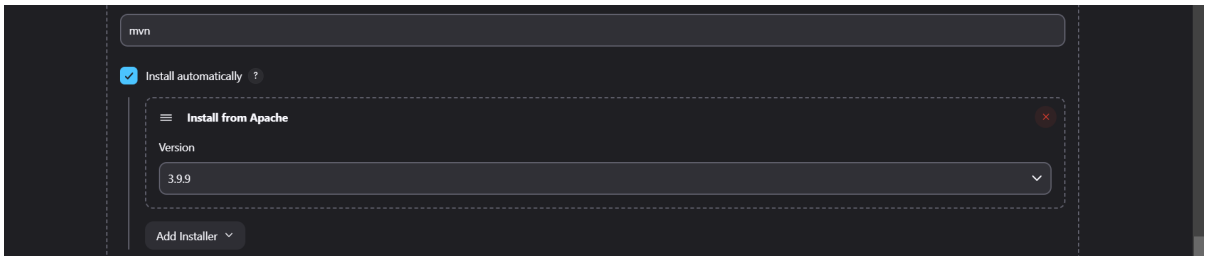
```
joston@ip-10-0-1-226:~$ ls
ansible.cfg  jenkins.sh  myinventory  playbook.yml
joston@ip-10-0-1-226:~$ ansible-playbook playbook.yml

PLAY [my playbook to do the project no-1] ***********************************************************************************************************************

TASK [Gathering Facts] ******************************************************************************************************************************************
ok: [34.134.255.100]

TASK [update the packages] **************************************************************************************************************************************
changed: [34.134.255.100]

TASK [copy the jenkins.sh file to ansible worker] ***************************************************************************************************************
changed: [34.134.255.100]

TASK [installing jenkins using jenkins.sh file] *****************************************************************************************************************
changed: [34.134.255.100]

TASK [install docker and git on ansible worker] *****************************************************************************************************************
changed: [34.134.255.100] => (item=docker.io)
ok: [34.134.255.100] => (item=git)

TASK [start the jenkins and docker services] ********************************************************************************************************************
ok: [34.134.255.100] => (item=jenkins)
ok: [34.134.255.100] => (item=docker)

TASK [give permission jenkins user to run docker commads] *******************************************************************************************************
changed: [34.134.255.100]

PLAY RECAP ******************************************************************************************************************************************************
34.134.255.100             : ok=7    changed=5    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

joston@ip-10-0-1-226:~$
```

```
SSH-in-browser                                                    UPLOAD FILE   DOWNLOAD FILE

joston@joston-pro:~$ ls
jenkins.sh
joston@joston-pro:~$ git --version
git version 2.25.1
joston@joston-pro:~$ docker --version
Docker version 24.0.7, build 24.0.7-0ubuntu2~20.04.1
joston@joston-pro:~$ jenkins --version
2.479.2
joston@joston-pro:~$
```

STEP-4

NOW CONNECT TO JENKINS AND AND CONFIGURE MAVEN TOOL.





STEP-5

NOW NAVIAGTE TO NEW ITEM AND PICK PIPELINE PROJECT AND START WRITING THE PIPELINE SCRIPT.

From this docker file docker image will be created and we wil create containter from this image.

Code | Blame | 5 lines (5 loc) · 117 Bytes | Code 55% faster with GitHub Copilot | Raw

```
FROM openjdk:11
ARG JAR_FILE=target/*.jar
COPY ${JAR_FILE} app.jar
EXPOSE 8081
ENTRYPOINT ["java","-jar","/app.jar"]
```

Not secure 34.134.255.100:8080/job/joston/configure

Dashboard > joston > Configuration

Configure

- General
- Advanced Project Options
- Pipeline

Script

```
pipeline {
    agent any
    tools{
        maven 'mvn'
    }
    stages {
        stage('cloning a repo') {
            steps {
                git 'https://github.com/luvis-joston-j/banking-finance-Project1.git'
            }
        }
        stage('testing the code') {
            steps {
                sh 'mvn test'
            }
        }
        stage('building the code') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('building docker image') {
            steps {
                sh 'docker build -t joston .'
            }
        }
         stage('building docker container') {
            steps {
                sh 'docker run -d -P joston'
            }
        }
    }
}
```

try sample Pipeline...

Save | Apply

NOW BUILD THE JOB.

Not secure 34.134.255.100:8080/job/joston/

Jenkins | Search (CTRL+K) | jos | log out

Dashboard > joston >

- Status
- Changes
- Build Now
- Configure
- Delete Pipeline
- Full Stage View
- Stages
- Rename
- Pipeline Syntax

Builds

Filter

Today

#1 10:37 AM

joston | Add description

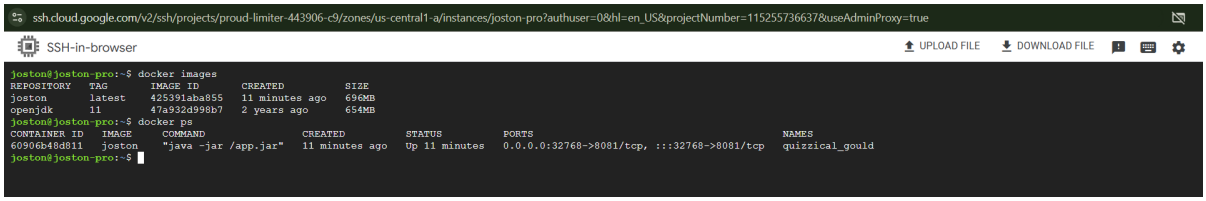
Stage View

	Declarative: Tool Install	cloning a repo	testing the code	building the code	building docker image	building docker container
Average stage times: (Average full run time: ~1min 35s)	988ms	1s	30s	32s	23s	1s
#1 Dec 13 16:07 No Changes	988ms	1s	30s	32s	23s	1s

Permalinks

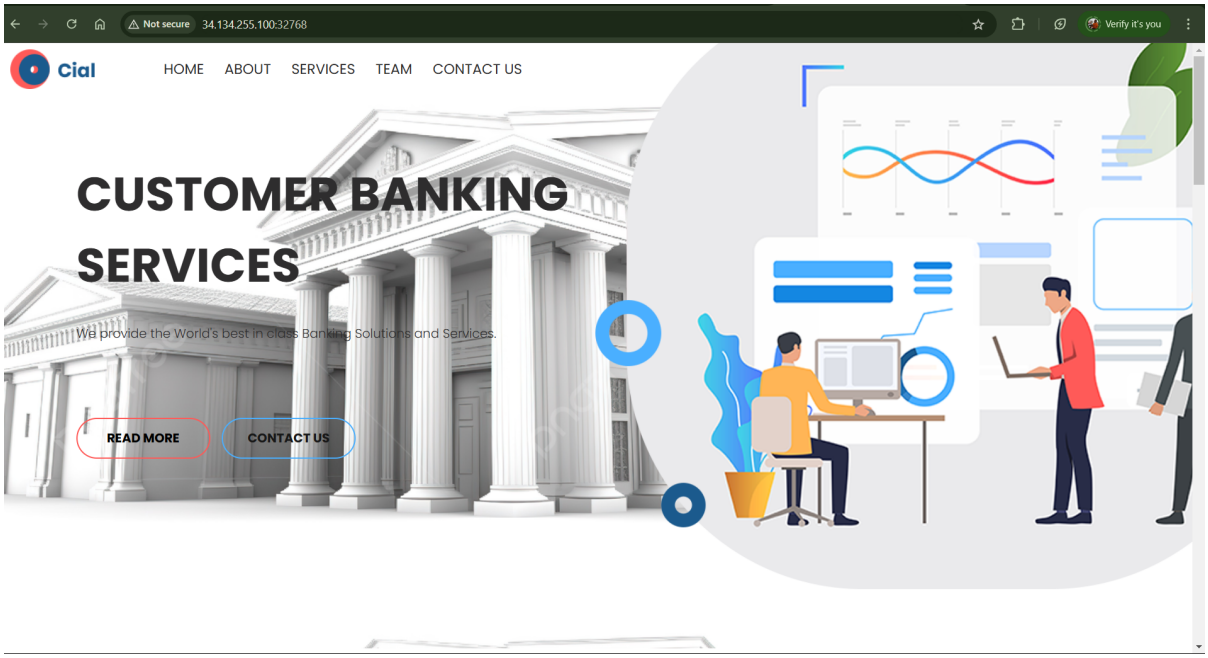
- Last build (#1), 6 min 53 sec ago
- Last stable build (#1), 6 min 53 sec ago
- Last successful build (#1), 6 min 53 sec ago
- Last completed build (#1), 6 min 53 sec ago

REST API | Jenkins 2.479.2



You can see created image and conatiner with port number.

Now access the banking app from internet using container port.

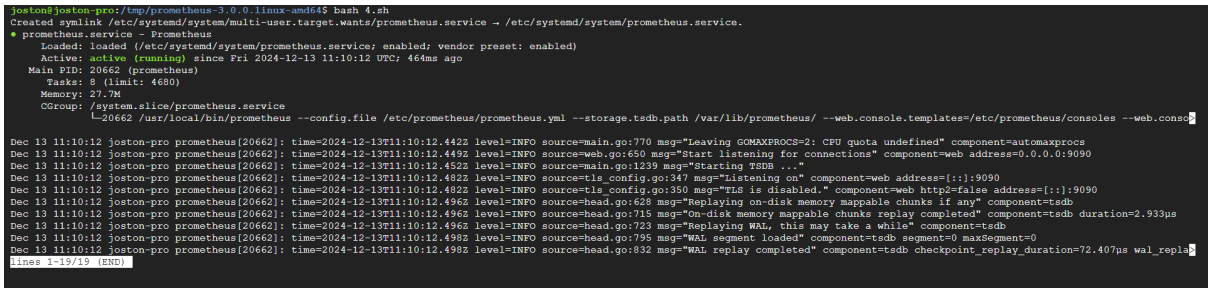


We successfully clone the repo and test the code and build the code and built a docker image and Created a docker container from that image.

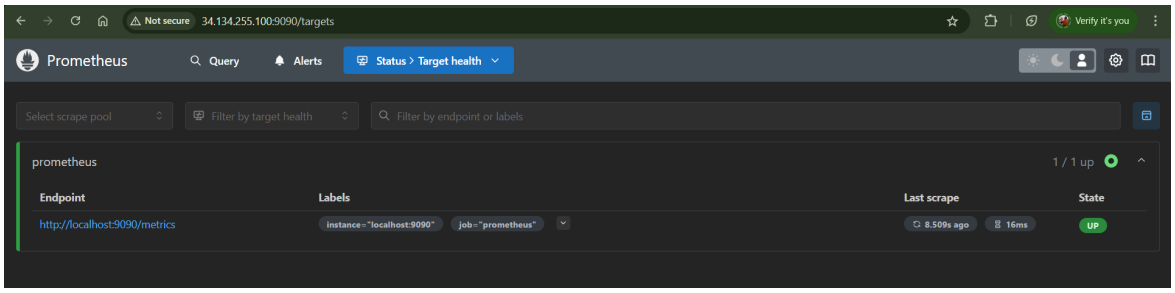
STEP-6

NOW INSTALL PROMETHEUS AND GRAFANA TO MONITER THE JENKINS.

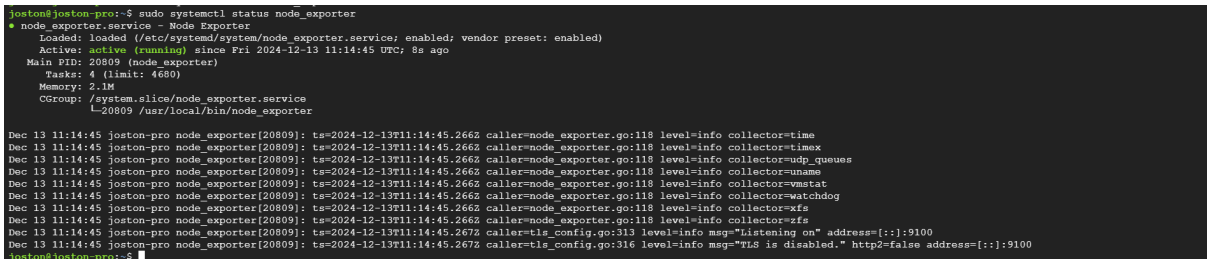
I HAVE INSTALLED PROMETHEUS IN MY INSTANCE.



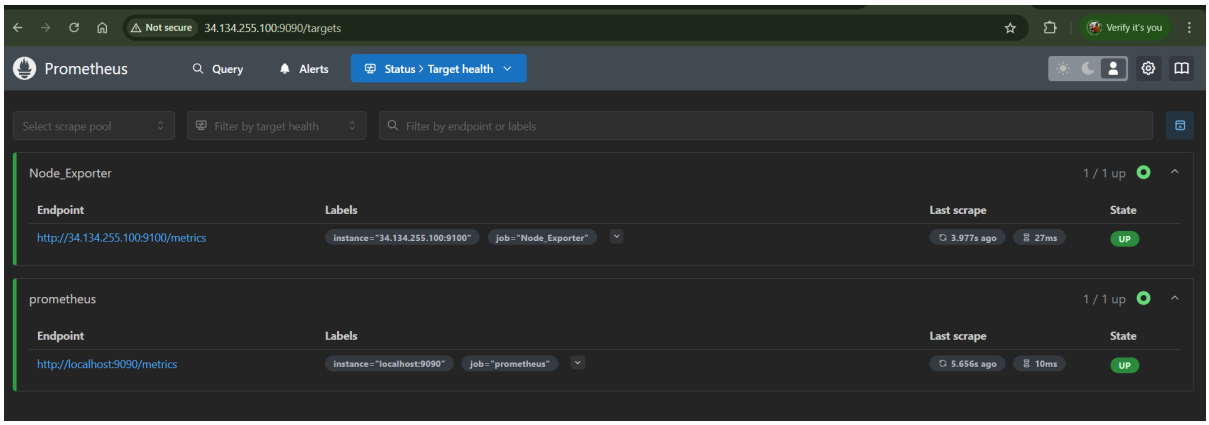
Its up and running.



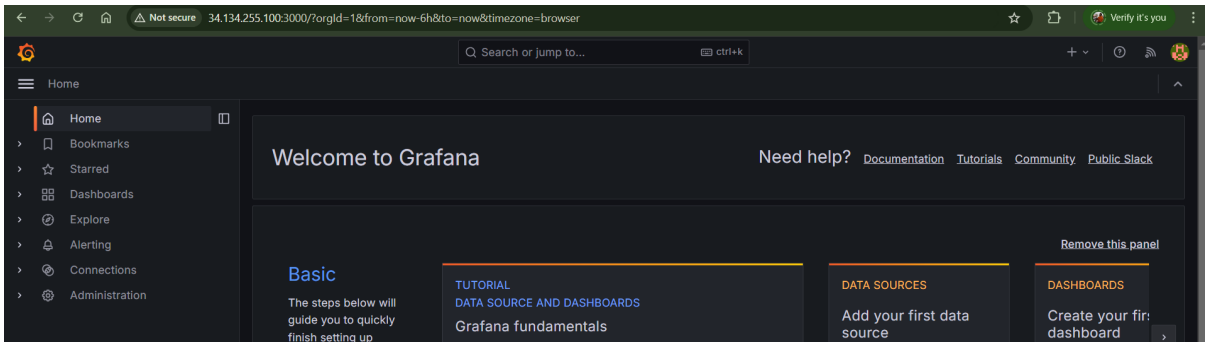
NOW INSTALL NODE EXPORTER TO COLLECT THE METRICS FROM JENKINS.



Now create a job for node exporter to collect metrics and give ip to targets.



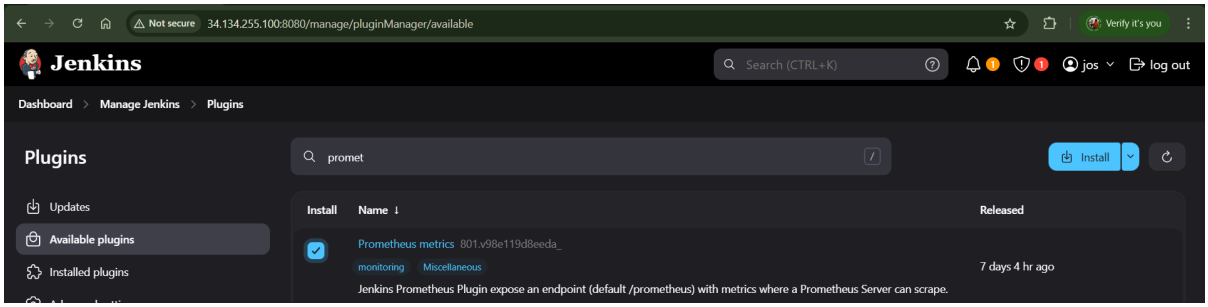
NOW INSTALL GRAFANA TO DIPLAY THE METRICS NODE EXPORTER HAVE COLLECTED.

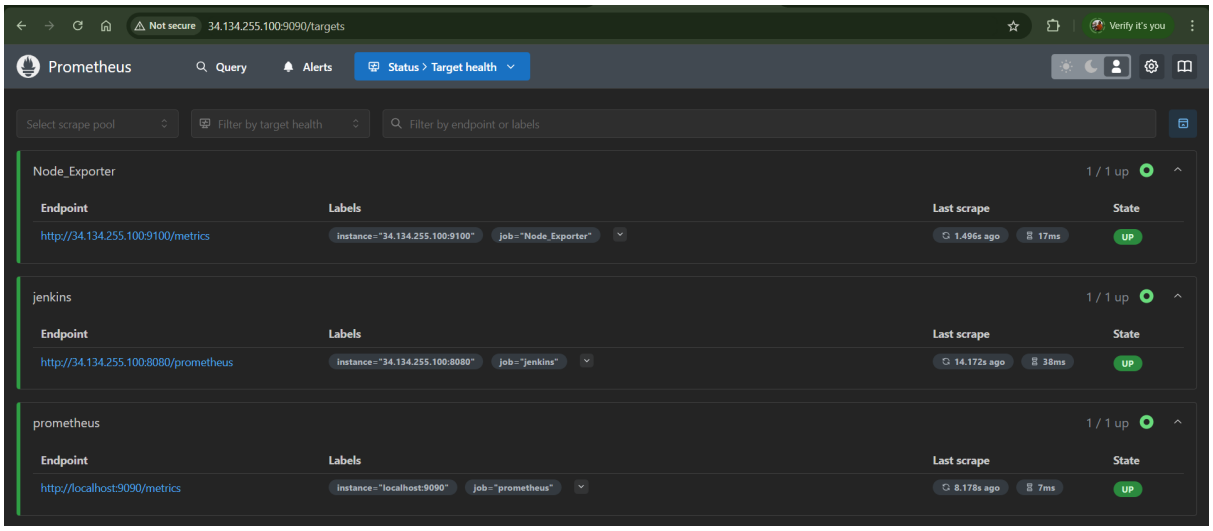


AND JENKINS JOB TO PROMETHEUS YML FILE TO COLLECT METRIC FROM JENKINS.

```
    - targets: ['34.134.255.100:9100']
  - job_name: 'jenkins'
    metrics_path: '/prometheus'
    static_configs:
      - targets: ['34.134.255.100:8080']
```

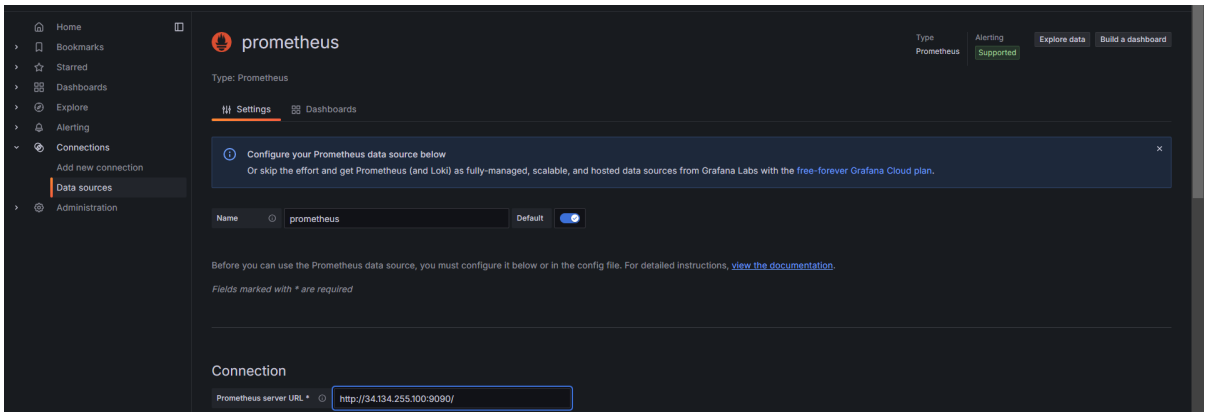
And install prometheus metrics plugin in jenkins dashboard.





STEP-7

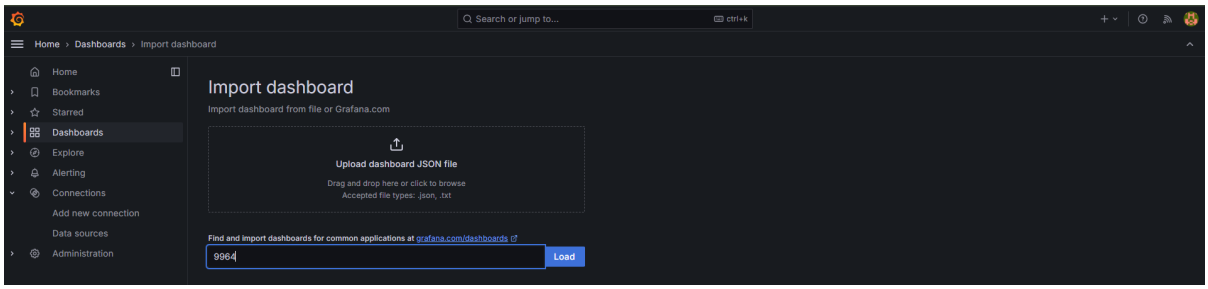
NOW NAVIGATE TO GRAFANA AND CLICK ON CONNECTION ADD THE DATASOURCE AS PROMETHEUS AND GIVE THE URL OF PROMETHEUS AND CLICK ON SAVE AND TEST.

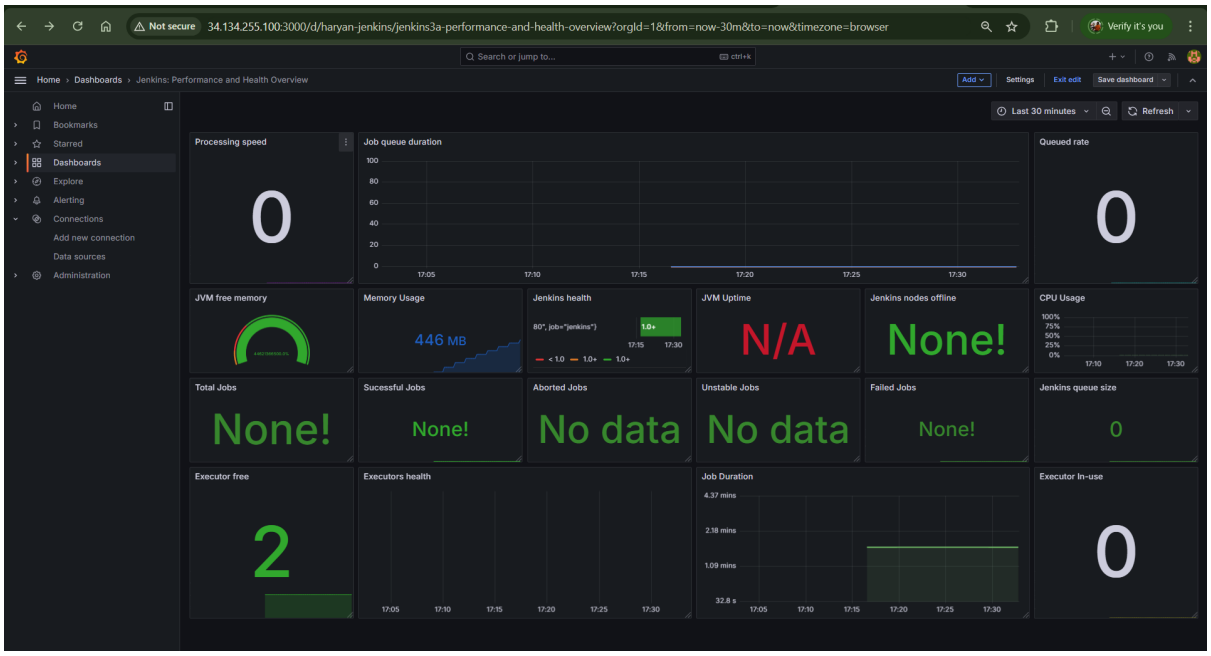


STEP-8

NOW IMPORT THE DASHBOARD TO MONITOR JENKINS.

USING 9964 ID NUMBER.





You can see the metrics that are collected from prometheus using grafana dashboard.

See its monitoring.

1. CPU utilization

2. Disk Space Utilization

3. Total Available Memory