Capstone Project

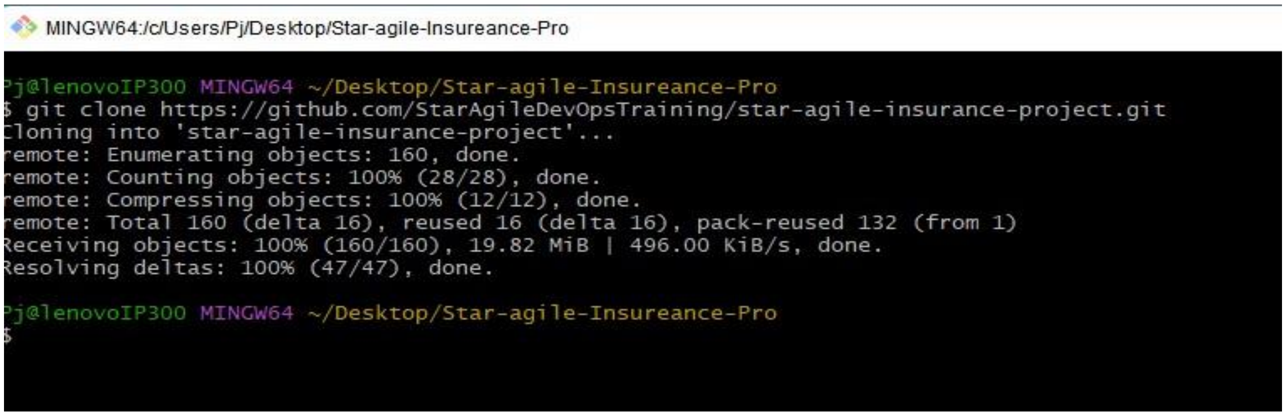
Insure-Me Life-Insurance Project

By:- Jonna Padmarao

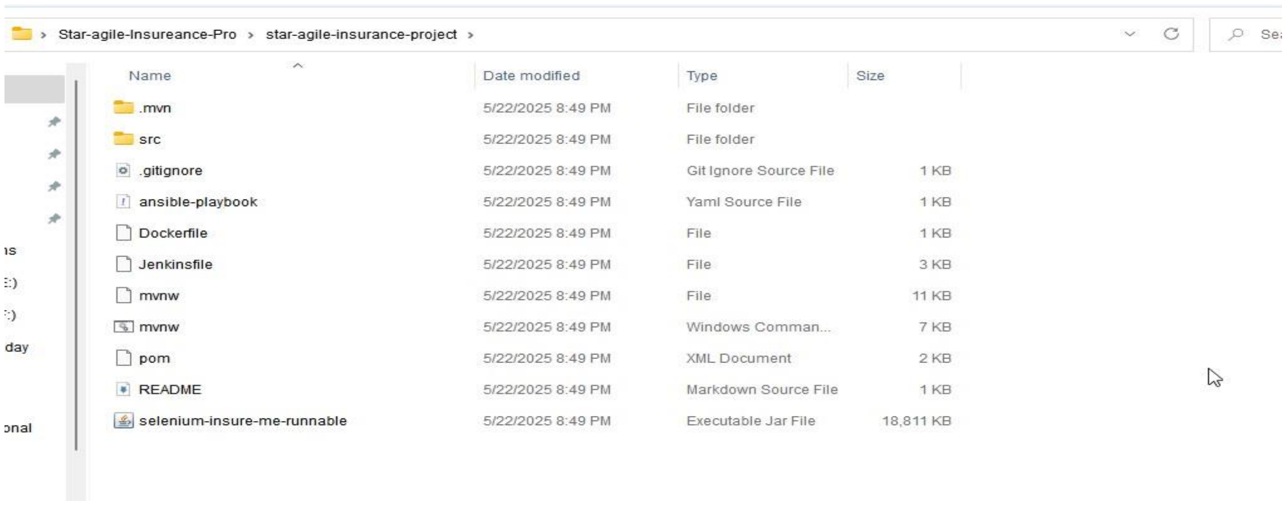
Source URL :- https://github.com/pj013525/star-agile-project-3.git

Step1:- On the desktop create a new folder (star-agile-Insurance-Pro) and enter into that folder and open the git bash in that folder

Step2:- Now give git clone https://github.com/StarAgileDevOpsTraining/star-agile-insuranceproject.git to get the project code in to that folder



```
MINGW64:/c/Users/Pj/Desktop/Star-agile-Insureance-Pro

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro
$ git clone https://github.com/StarAgileDevOpsTraining/star-agile-insurance-project.git
Cloning into 'star-agile-insurance-project'...
remote: Enumerating objects: 160, done.
remote: Counting objects: 100% (28/28), done.
remote: Compressing objects: 100% (12/12), done.
remote: Total 160 (delta 16), reused 16 (delta 16), pack-reused 132 (from 1)
Receiving objects: 100% (160/160), 19.82 MiB | 496.00 KiB/s, done.
Resolving deltas: 100% (47/47), done.

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro
$
```



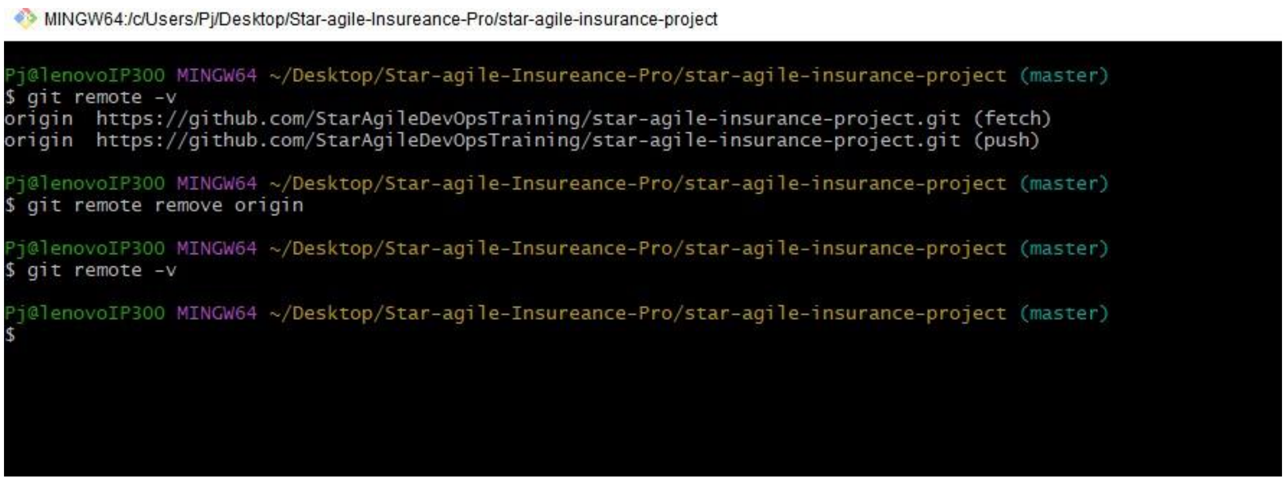
Star-agile-Insureance-Pro › star-agile-insurance-project ›

Name	Date modified	Type	Size
.mvn	5/22/2025 8:49 PM	File folder	
src	5/22/2025 8:49 PM	File folder	
.gitignore	5/22/2025 8:49 PM	Git Ignore Source File	1 KB
ansible-playbook	5/22/2025 8:49 PM	Yaml Source File	1 KB
Dockerfile	5/22/2025 8:49 PM	File	1 KB
Jenkinsfile	5/22/2025 8:49 PM	File	3 KB
mvnw	5/22/2025 8:49 PM	File	11 KB
mvnw	5/22/2025 8:49 PM	Windows Comman...	7 KB
pom	5/22/2025 8:49 PM	XML Document	2 KB
README	5/22/2025 8:49 PM	Markdown Source File	1 KB
selenium-insure-me-runnable	5/22/2025 8:49 PM	Executable Jar File	18,811 KB

Step3:- Now go to the folder that we get from git clone and again opengit bash there and check the origin and remove that origin

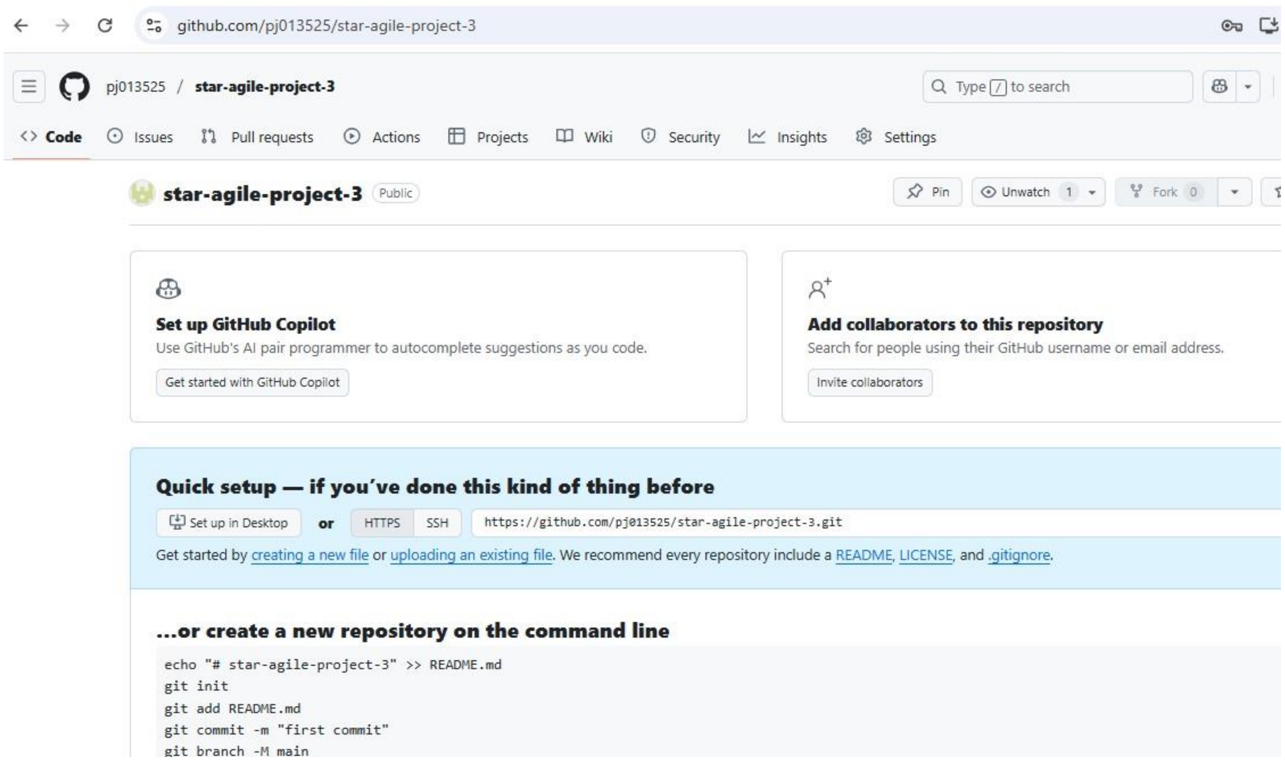
git remote -v --> To get origin list

git remote remove origin ==> to remove the origin


MINGW64:/c/Users/Pj/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project

```
Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git remote -v
origin  https://github.com/StarAgileDevOpsTraining/star-agile-insurance-project.git (fetch)
origin  https://github.com/StarAgileDevOpsTraining/star-agile-insurance-project.git (push)

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git remote remove origin

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git remote -v

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$
```

Step4:- Now go to github and create a new repo and copy the url in the gitbash



Set up GitHub Copilot
Use GitHub's AI pair programmer to autocomplete suggestions as you code.

Add collaborators to this repository
Search for people using their GitHub username or email address.

Quick setup — if you've done this kind of thing before

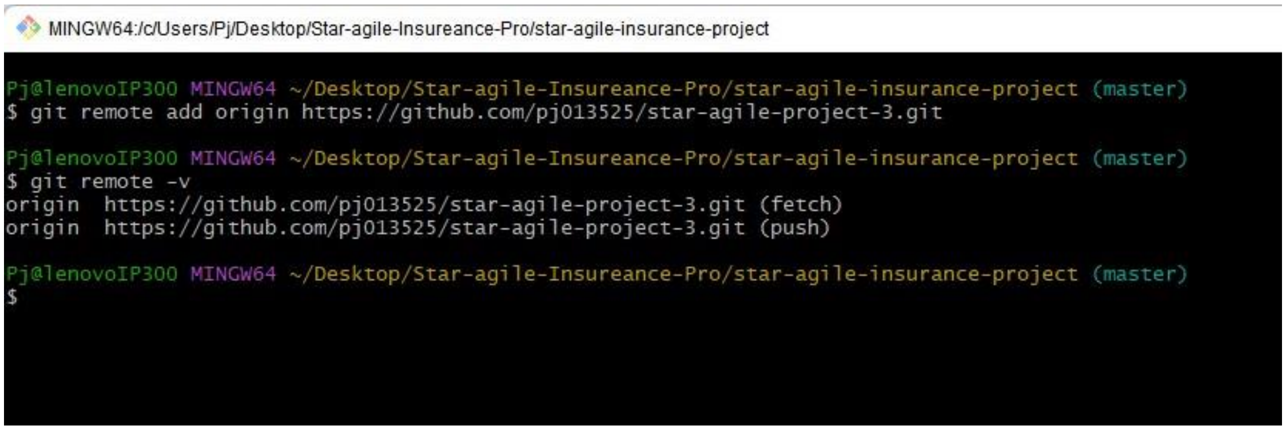
https://github.com/pj013525/star-agile-project-3.git

Get started by creating a new file or uploading an existing file. We recommend every repository include a README, LICENSE, and .gitignore.

...or create a new repository on the command line

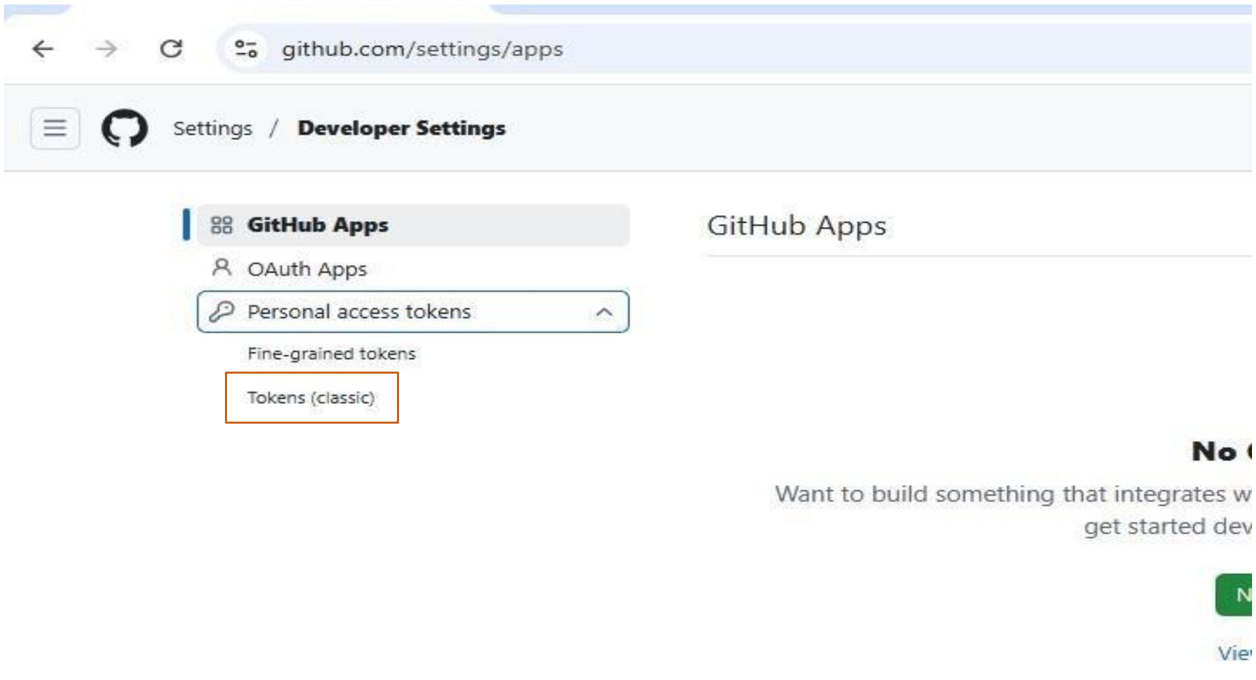
```
echo "# star-agile-project-3" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
```

Step5:- Now again go to the gitbash and add this git repo url in the project by using git remote add origin <git-repo-url> and verify

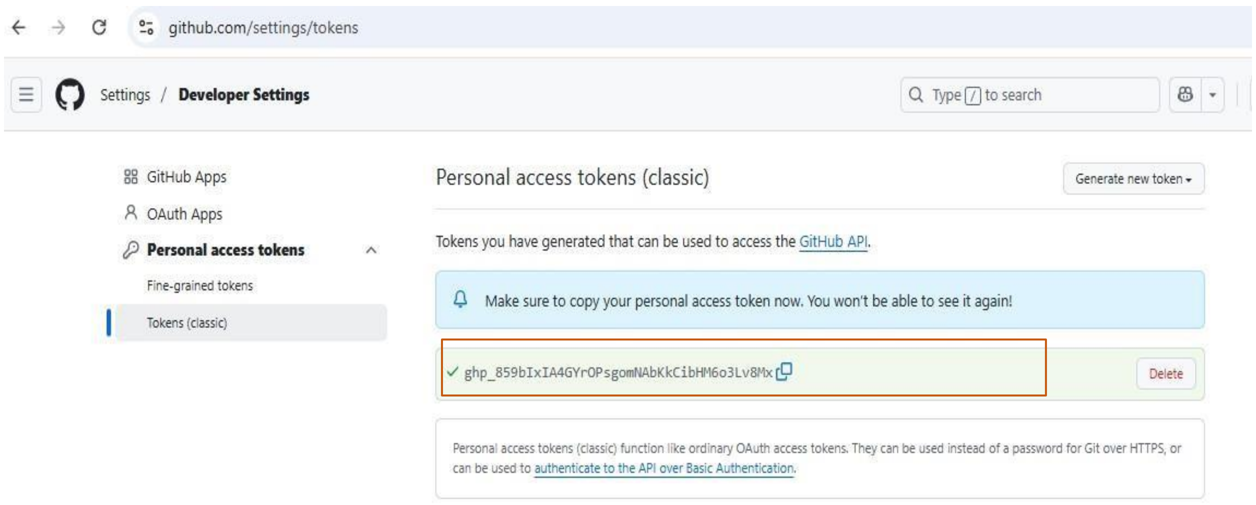


```
MINGW64:/c/Users/Pj/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git remote add origin https://github.com/pj013525/star-agile-project-3.git

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git remote -v
origin  https://github.com/pj013525/star-agile-project-3.git (fetch)
origin  https://github.com/pj013525/star-agile-project-3.git (push)

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$
```

Step6:- Now again go to github ❼ Profile setting ❼Developer settings ❼ Personal access token Tokens(classic) ❼Generate new token



Step7:- Now a token will be generated , copy this token that generated since it is only available for one time only



Personal access tokens (classic)

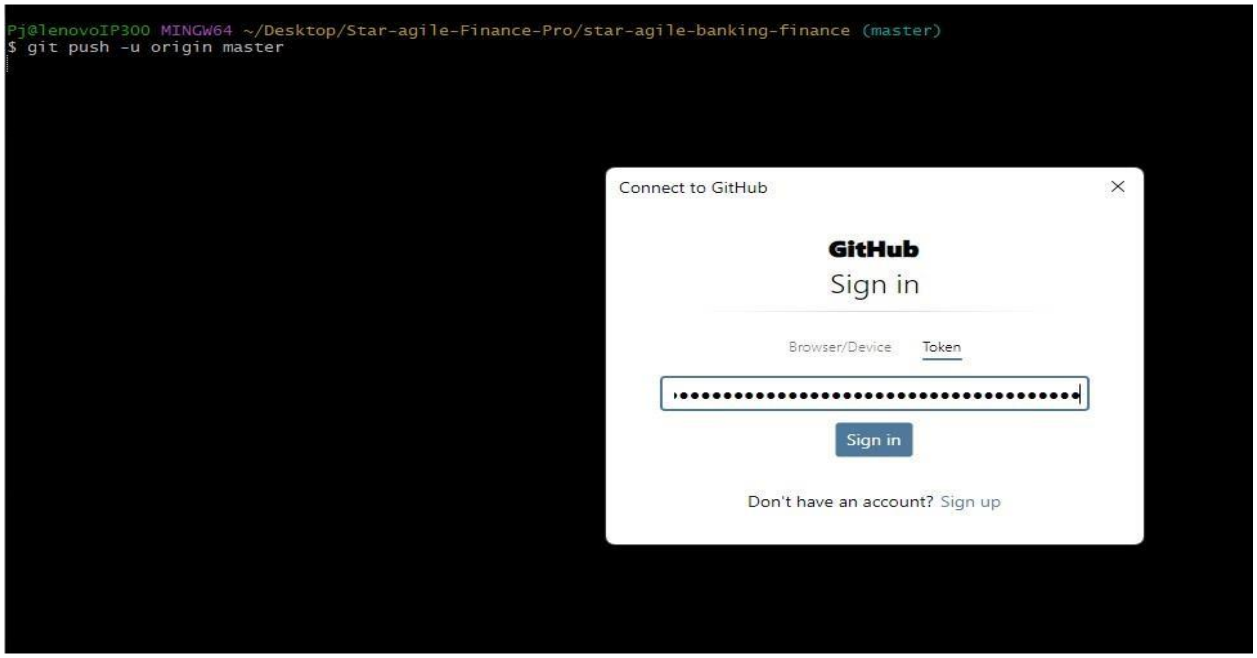
Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ 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

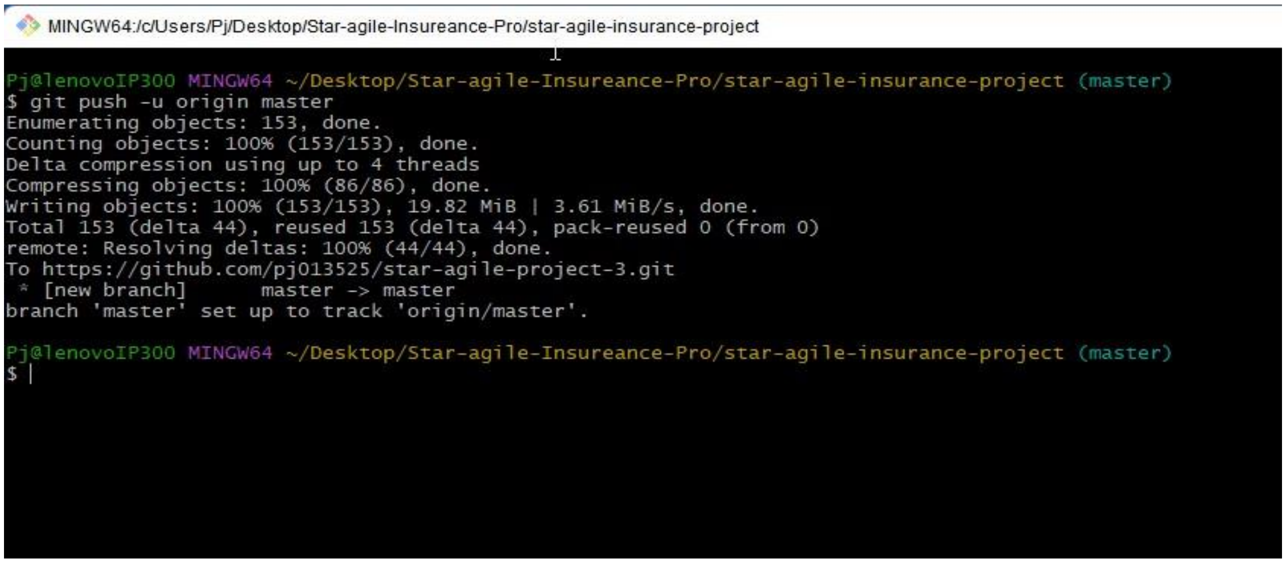
Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Step8:- Now give link this the remote repo with gitbash using this token git push -u origin master and paste the token the copied from the github and press sign in



```
Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Finance-Pro/star-agile-banking-finance (master)
$ git push -u origin master
```

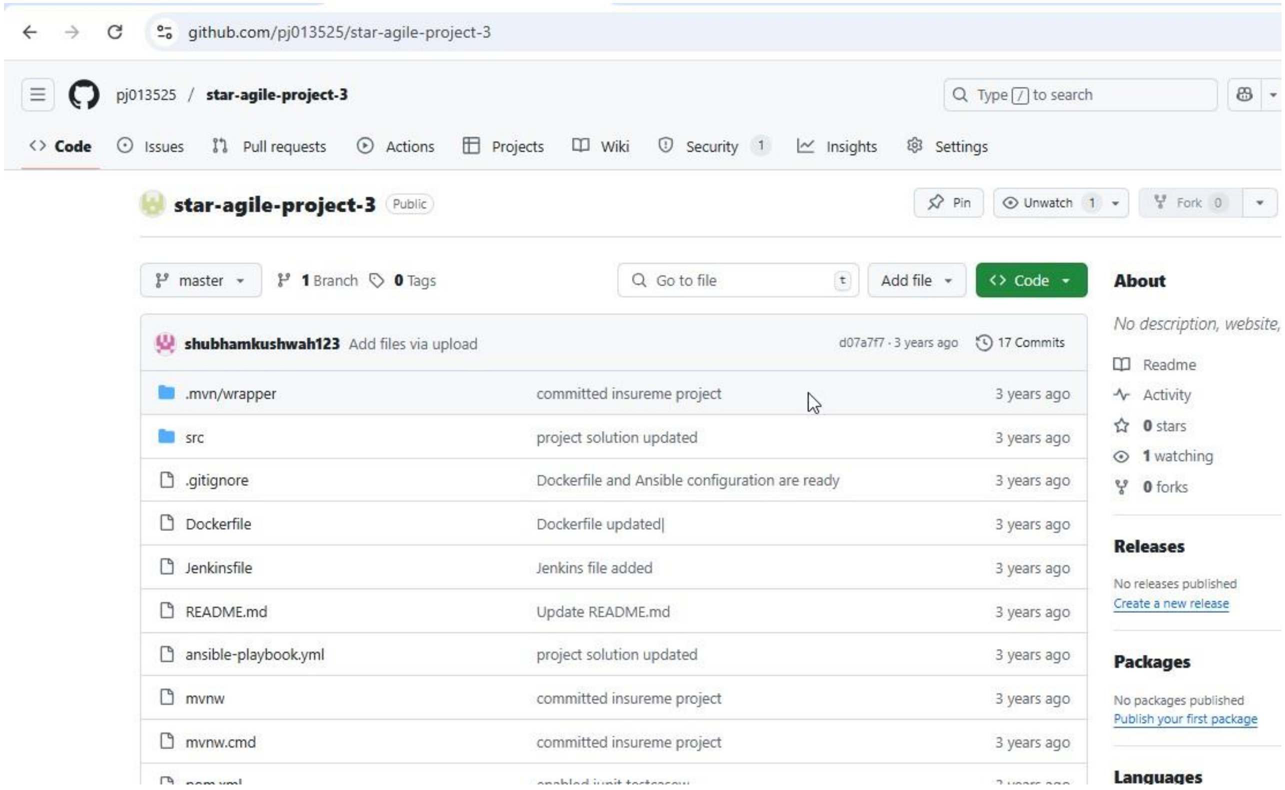
Step9:- Now the master branch will be set to our repo by default



MINGW64:/c/Users/Pj/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project

```
Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ git push -u origin master
Enumerating objects: 153, done.
Counting objects: 100% (153/153), done.
Delta compression using up to 4 threads
Compressing objects: 100% (86/86), done.
Writing objects: 100% (153/153), 19.82 MiB | 3.61 MiB/s, done.
Total 153 (delta 44), reused 153 (delta 44), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (44/44), done.
To https://github.com/pj013525/star-agile-project-3.git
 * [new branch]      master -> master
branch 'master' set up to track 'origin/master'.

Pj@lenovoIP300 MINGW64 ~/Desktop/Star-agile-Insureance-Pro/star-agile-insurance-project (master)
$ |
```

Step10:- Now go to the github repo and you will see the source code in that repo



github.com/pj013525/star-agile-project-3

pj013525 / star-agile-project-3

star-agile-project-3 Public

master · 1 Branch · 0 Tags

shubhamkushwah123 Add files via upload — d07a7f7 · 3 years ago — 17 Commits

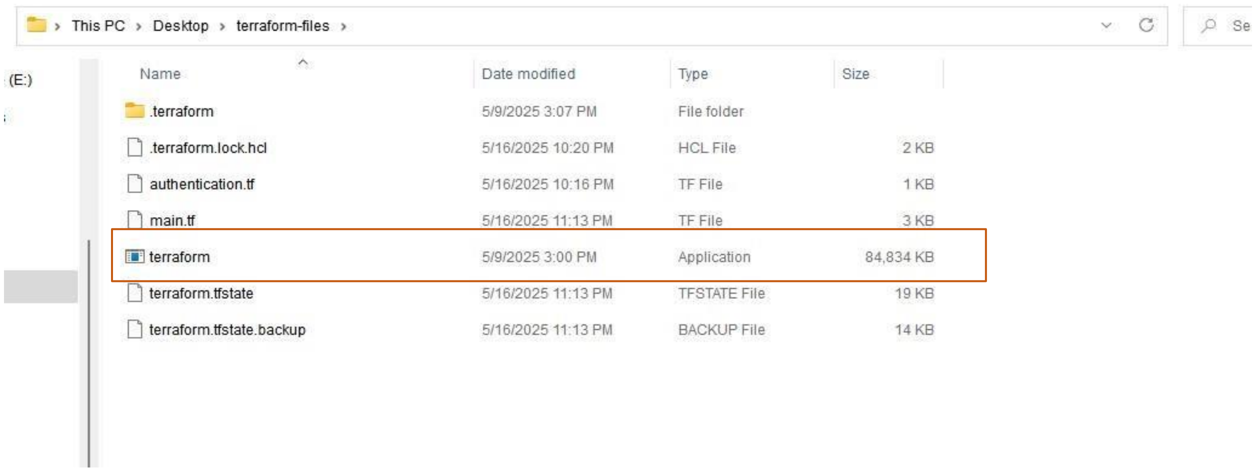
.mvn/wrapper	committed insureme project	3 years ago
src	project solution updated	3 years ago
.gitignore	Dockerfile and Ansible configuration are ready	3 years ago
Dockerfile	Dockerfile updated|	3 years ago
Jenkinsfile	Jenkins file added	3 years ago
README.md	Update README.md	3 years ago
ansible-playbook.yml	project solution updated	3 years ago
mvnw	committed insureme project	3 years ago
mvnw.cmd	committed insureme project	3 years ago

About: No description, website, · Readme · Activity · 0 stars · 1 watching · 0 forks

Releases: No releases published · Create a new release

Packages: No packages published · Publish your first package

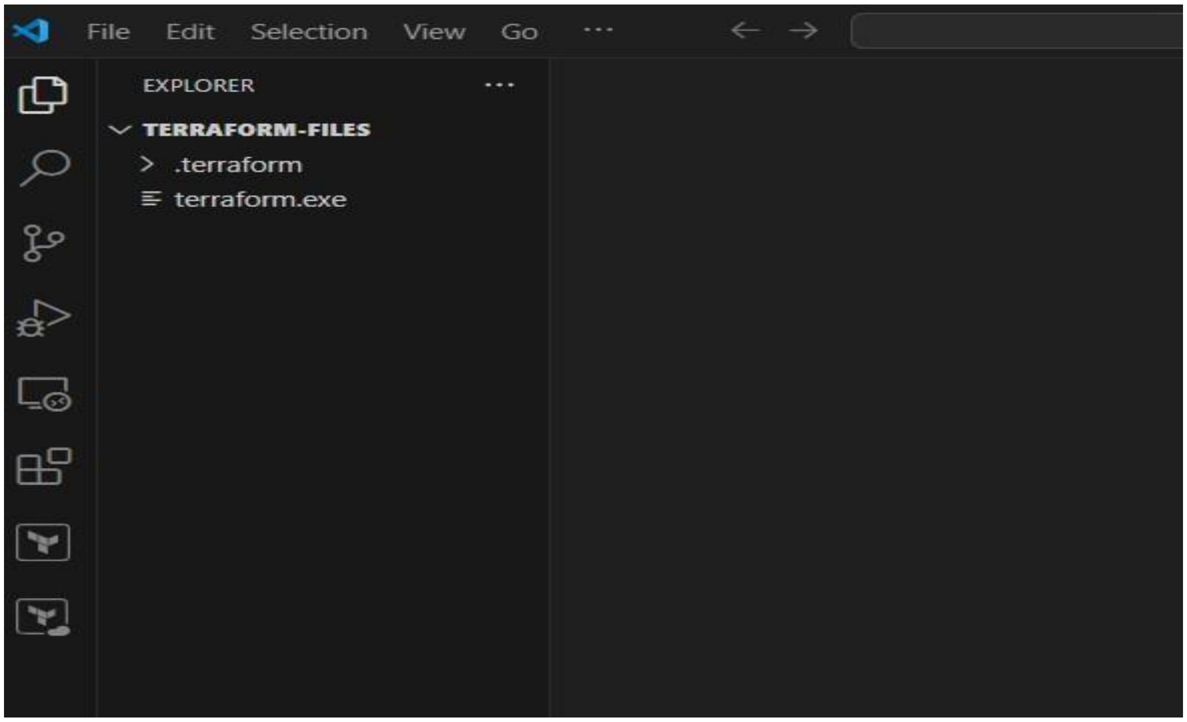
Step11:- Now create an instance using terraform as Iaac , and for that create a folder on desktop and go to browser download terraform for windows then a terraform application will be generated , now copy this application in to that folder and save



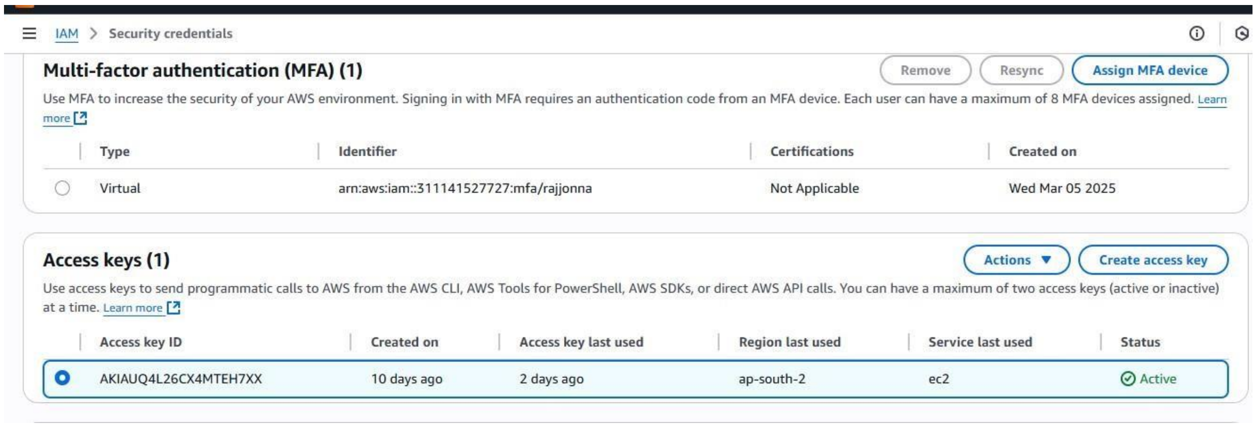
This PC › Desktop › terraform-files ›

Name	Date modified	Type	Size
.terraform	5/9/2025 3:07 PM	File folder	
.terraform.lock.hcl	5/16/2025 10:20 PM	HCL File	2 KB
authentication.tf	5/16/2025 10:16 PM	TF File	1 KB
main.tf	5/16/2025 11:13 PM	TF File	3 KB
terraform	5/9/2025 3:00 PM	Application	84,834 KB
terraform.tfstate	5/16/2025 11:13 PM	TFSTATE File	19 KB
terraform.tfstate.backup	5/16/2025 11:13 PM	BACKUP File	14 KB

Step12:- open visual studio code and go to terraform folder



Step13:- Now create a file authentication.tf and give the provider and for that select the region in which you want to launch the server and go to aws account and go to profile ❼ credentials and go to access keys



Multi-factor authentication (MFA) (1)

Use MFA to increase the security of your AWS environment. Signing in with MFA requires an authentication code from an MFA device. Each user can have a maximum of 8 MFA devices assigned. Learn more

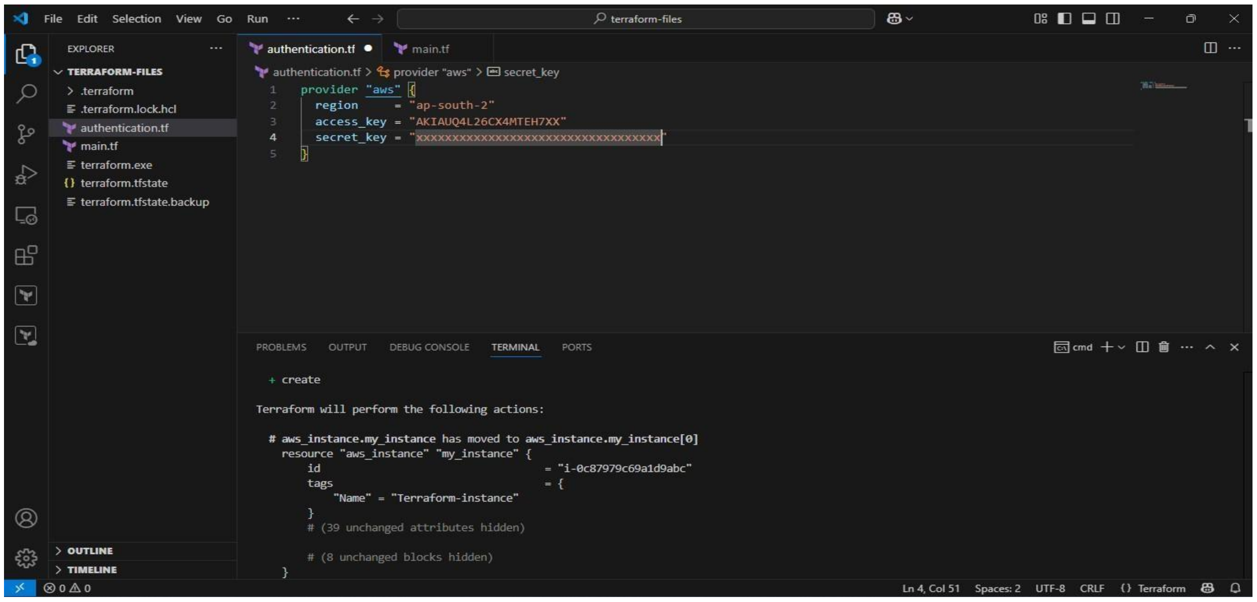
	Type	Identifier	Certifications	Created on
○	Virtual	arn:aws:iam::311141527727:mfa/rajjonna	Not Applicable	Wed Mar 05 2025

Access keys (1)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

	Access key ID	Created on	Access key last used	Region last used	Service last used	Status
●	AKIAUQ4L26CX4MTEH7XX	10 days ago	2 days ago	ap-south-2	ec2	✓ Active

Step14:- now copy this access key details and paste it in this authentication.tf file and initialize it

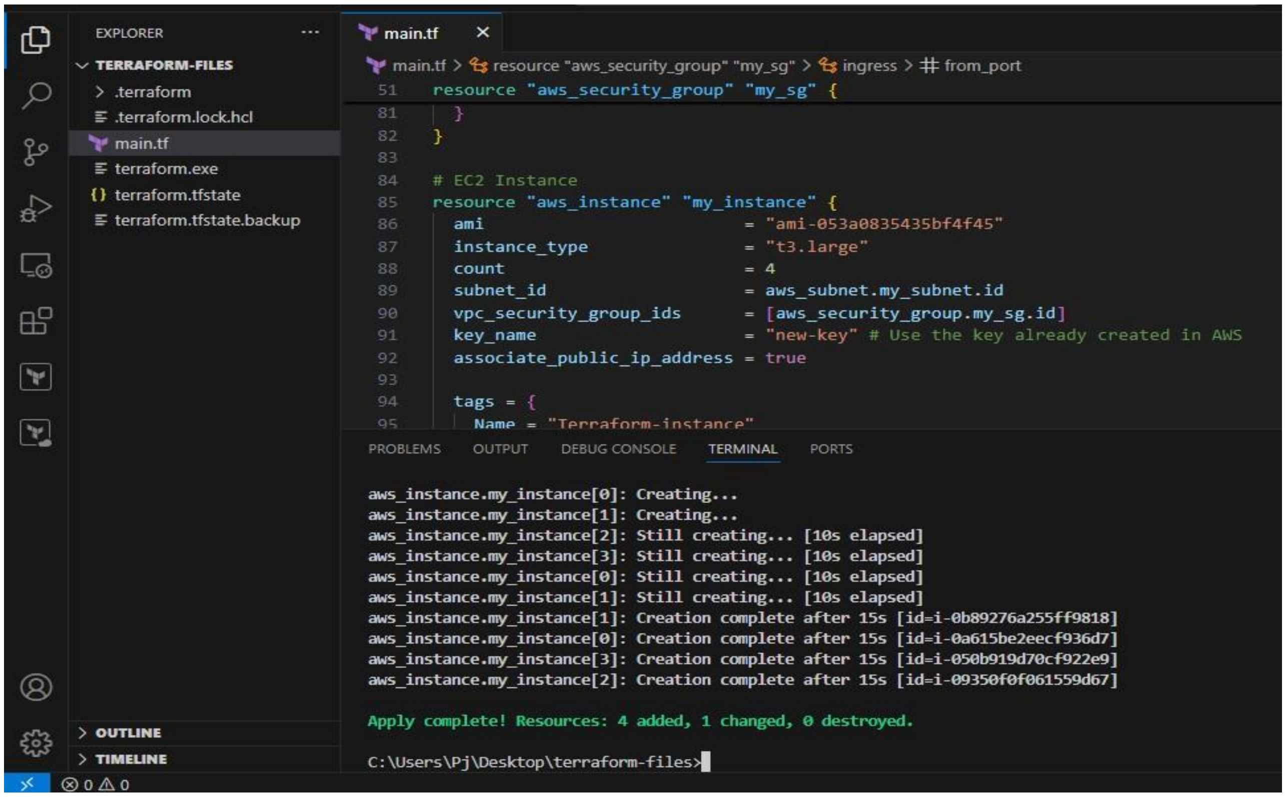


File Edit Selection View Go Run ...
terraform-files
EXPLORER
TERRAFORM-FILES
.terraform
.terraform.lock.hcl
authentication.tf
main.tf
terraform.exe
terraform.tfstate
terraform.tfstate.backup
OUTLINE
TIMELINE
authentication.tf > provider "aws" > secret_key
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Ln 4, Col 51 Spaces: 2 UTF-8 CRLF {} Terraform

```
provider "aws" {
  region     = "ap-south-2"
  access_key = "AKIAUQ4L26CX4MTEH7XX"
  secret_key = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
}
```

```
  + create

Terraform will perform the following actions:

  # aws_instance.my_instance has moved to aws_instance.my_instance[0]
    resource "aws_instance" "my_instance" {
        id                                   = "i-0c87979c69a1d9abc"
        tags                                 = {
            "Name" = "Terraform-instance"
        }
        # (39 unchanged attributes hidden)

        # (8 unchanged blocks hidden)
    }
```

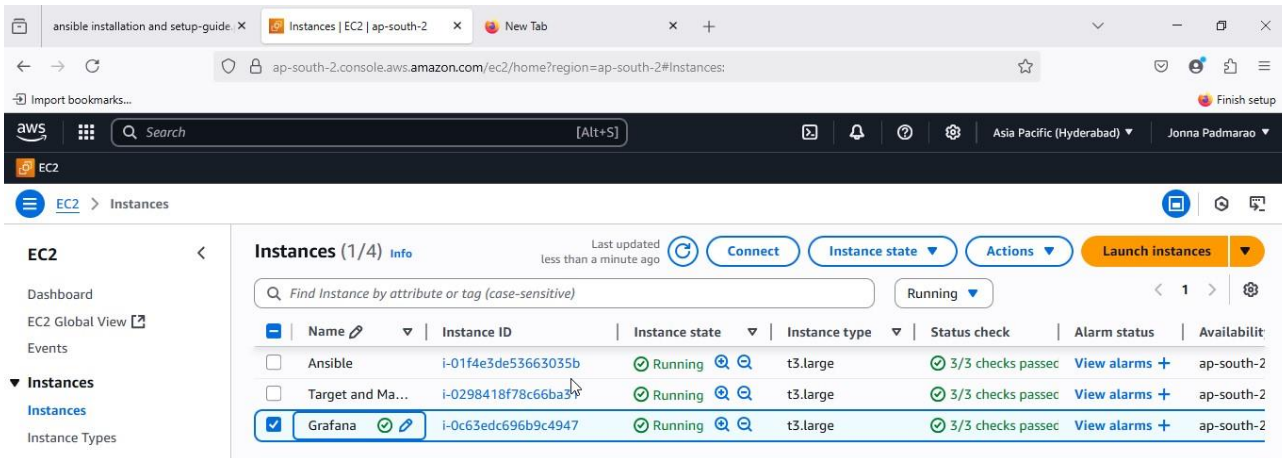
Step15:- After it is successful now create a new file main.tf and give resources details to create instance



```
resource "aws_security_group" "my_sg" {
  }
}

# EC2 Instance
resource "aws_instance" "my_instance" {
  ami                         = "ami-053a0835435bf4f45"
  instance_type               = "t3.large"
  count                       = 4
  subnet_id                   = aws_subnet.my_subnet.id
  vpc_security_group_ids      = [aws_security_group.my_sg.id]
  key_name                    = "new-key" # Use the key already created in AWS
  associate_public_ip_address = true

  tags = {
    Name = "Terraform-instance"
```

```
aws_instance.my_instance[0]: Creating...
aws_instance.my_instance[1]: Creating...
aws_instance.my_instance[2]: Still creating... [10s elapsed]
aws_instance.my_instance[3]: Still creating... [10s elapsed]
aws_instance.my_instance[0]: Still creating... [10s elapsed]
aws_instance.my_instance[1]: Still creating... [10s elapsed]
aws_instance.my_instance[1]: Creation complete after 15s [id=i-0b89276a255ff9818]
aws_instance.my_instance[0]: Creation complete after 15s [id=i-0a615be2eecf936d7]
aws_instance.my_instance[3]: Creation complete after 15s [id=i-050b919d70cf922e9]
aws_instance.my_instance[2]: Creation complete after 15s [id=i-09350f0f061559d67]

Apply complete! Resources: 4 added, 1 changed, 0 destroyed.

C:\Users\Pj\Desktop\terraform-files>
```

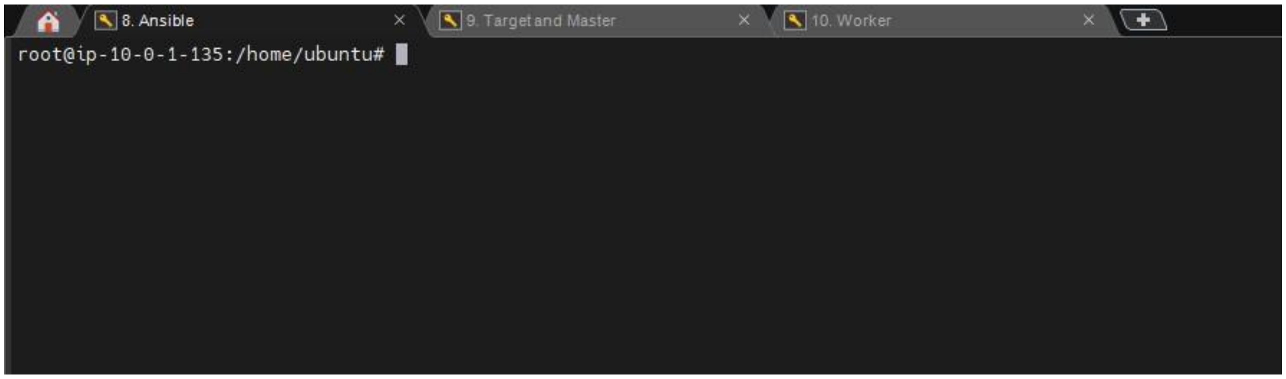
Step16:- After it is successful go and check the aws console and rename them as Ansible , Target and Master, Worker and Grafana instances



Instances (1/4)

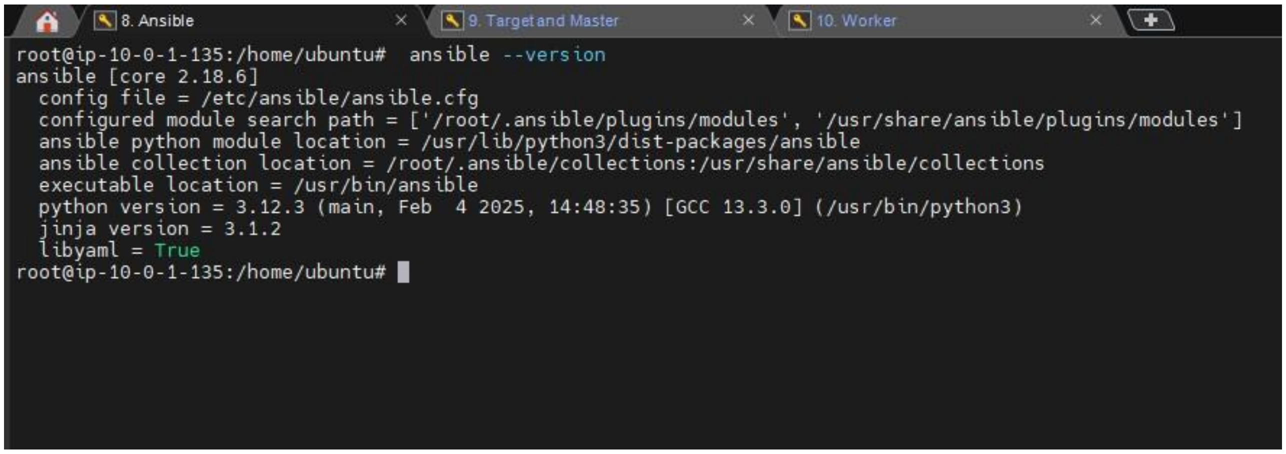
	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availabilit
☐	Ansible	i-01f4e3de53663035b	Running	t3.large	3/3 checks passec	View alarms +	ap-south-2
☐	Target and Ma...	i-0298418f78c66ba3	Running	t3.large	3/3 checks passec	View alarms +	ap-south-2
✓	Grafana	i-0c63edc696b9c4947	Running	t3.large	3/3 checks passec	View alarms +	ap-south-2

Step17:- Now connect to Ansible and target and master and worker servers using Mobaxterm agent and launch an instance

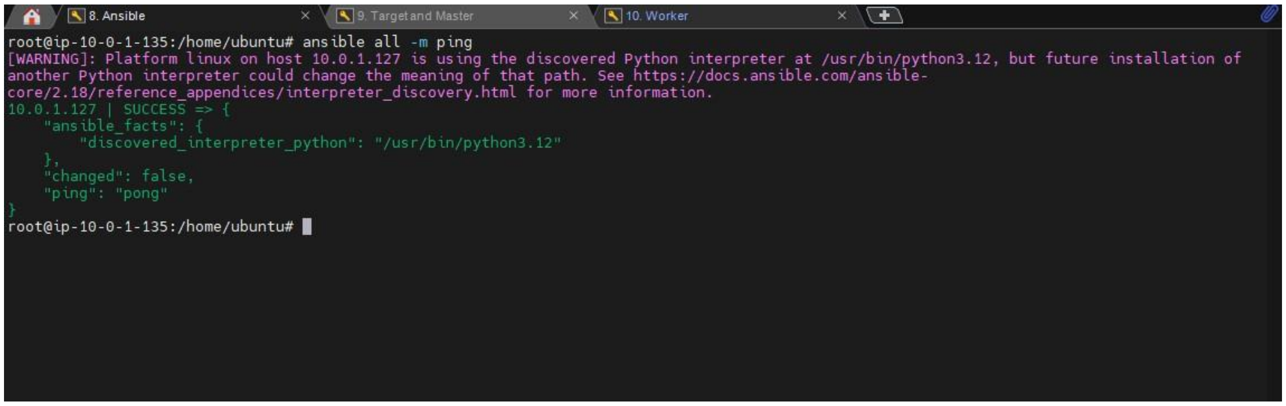


```
root@ip-10-0-1-135:/home/ubuntu#
```

Step18:- Now install Ansible in Ansible server and connect this server with the Target and master sever and enable All traffic in the security group of this server

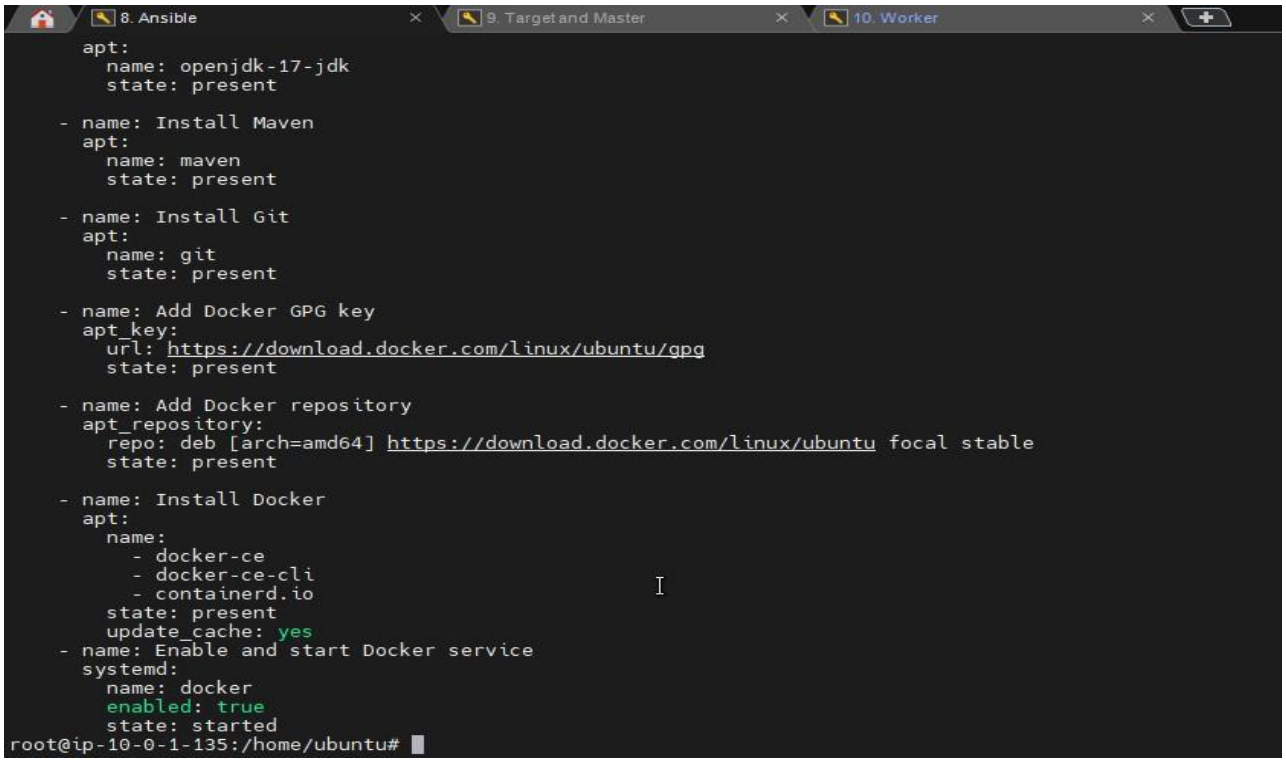


```
root@ip-10-0-1-135:/home/ubuntu#  ansible --version
ansible [core 2.18.6]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/root/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /root/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.12.3 (main, Feb  4 2025, 14:48:35) [GCC 13.3.0] (/usr/bin/python3)
  jinja version = 3.1.2
  libyaml = True
root@ip-10-0-1-135:/home/ubuntu#
```

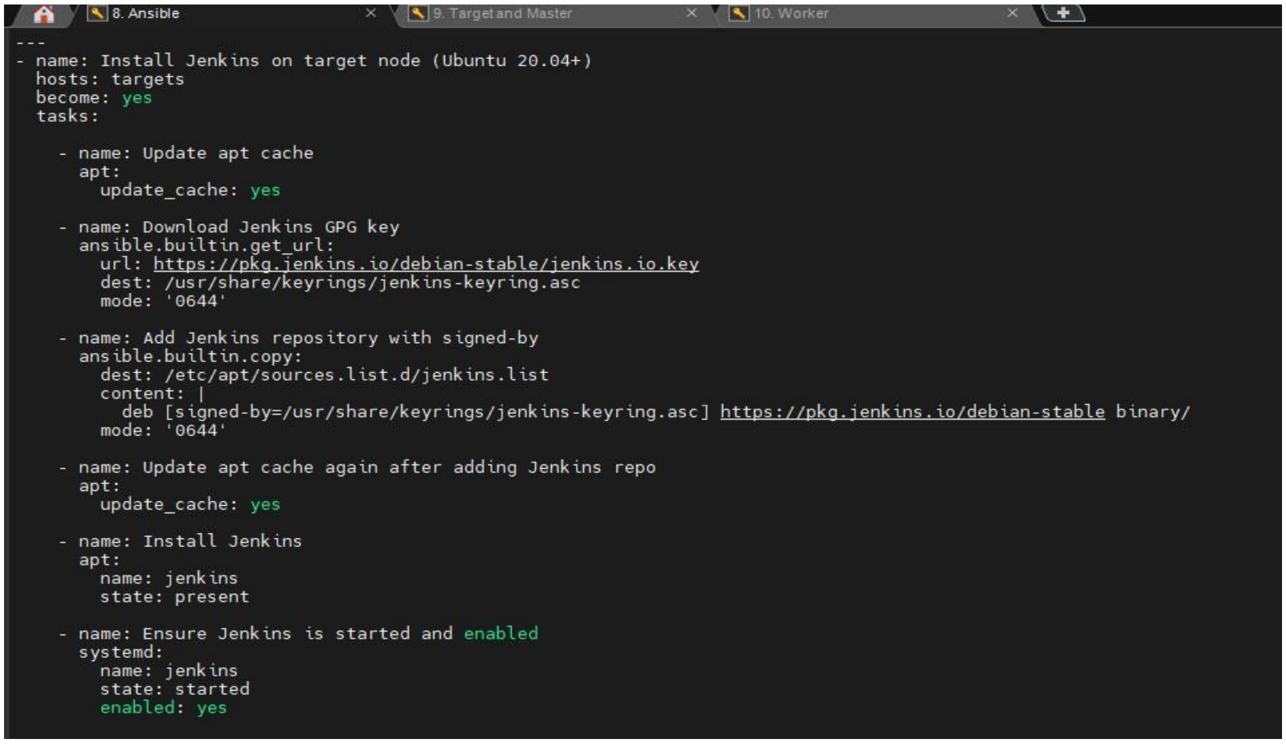


```
root@ip-10-0-1-135:/home/ubuntu# ansible all -m ping
[WARNING]: Platform linux on host 10.0.1.127 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of
another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
10.0.1.127 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3.12"
    },
    "changed": false,
    "ping": "pong"
}
root@ip-10-0-1-135:/home/ubuntu#
```

Step19:- Now install java, maven, docker, jenkins in the target and master server using Ansible sever

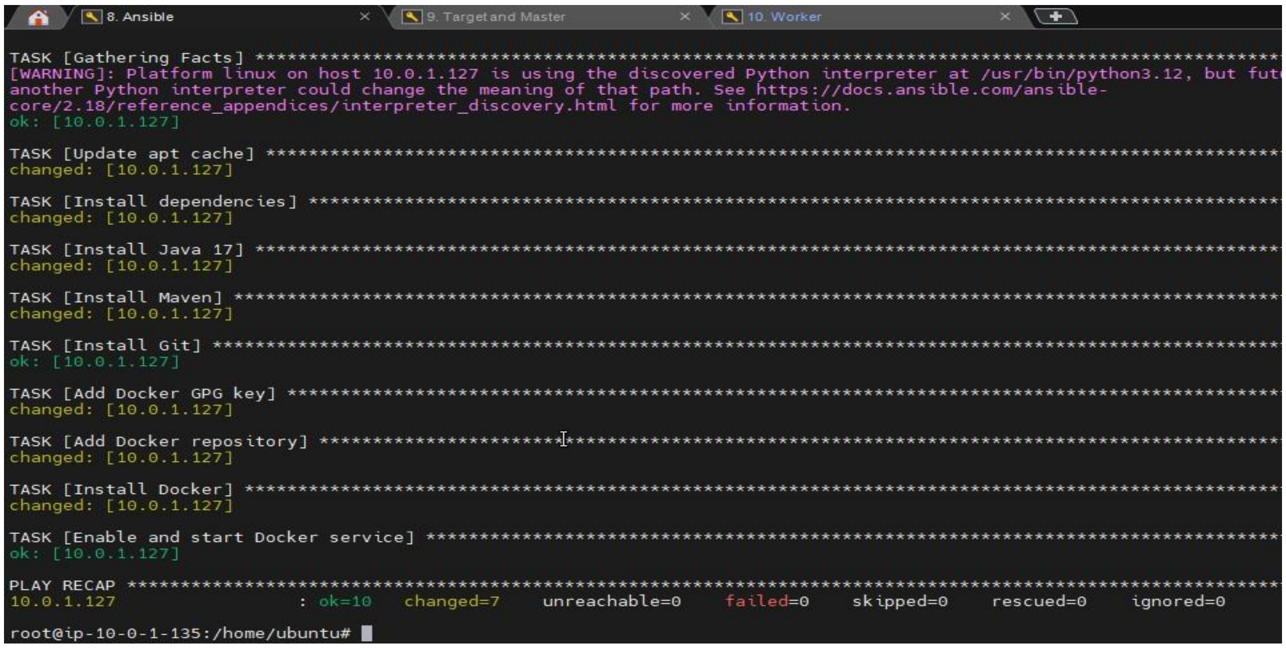


```
      apt:
        name: openjdk-17-jdk
        state: present

    - name: Install Maven
      apt:
        name: maven
        state: present

    - name: Install Git
      apt:
        name: git
        state: present

    - name: Add Docker GPG key
      apt_key:
        url: https://download.docker.com/linux/ubuntu/gpg
        state: present

    - name: Add Docker repository
      apt_repository:
        repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu focal stable
        state: present

    - name: Install Docker
      apt:
        name:
          - docker-ce
          - docker-ce-cli
          - containerd.io
        state: present
        update_cache: yes
    - name: Enable and start Docker service
      systemd:
        name: docker
        enabled: true
        state: started
root@ip-10-0-1-135:/home/ubuntu#
```

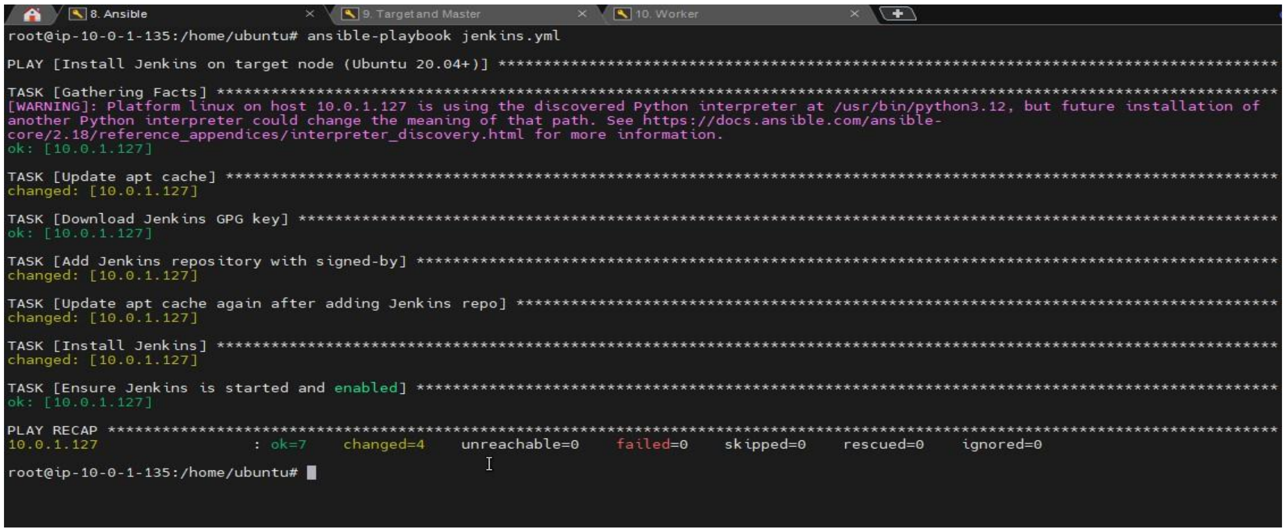


```
---
- name: Install Jenkins on target node (Ubuntu 20.04+)
  hosts: targets
  become: yes
  tasks:

    - name: Update apt cache
      apt:
        update_cache: yes

    - name: Download Jenkins GPG key
      ansible.builtin.get_url:
        url: https://pkg.jenkins.io/debian-stable/jenkins.io.key
        dest: /usr/share/keyrings/jenkins-keyring.asc
        mode: '0644'

    - name: Add Jenkins repository with signed-by
      ansible.builtin.copy:
        dest: /etc/apt/sources.list.d/jenkins.list
        content: |
          deb [signed-by=/usr/share/keyrings/jenkins-keyring.asc] https://pkg.jenkins.io/debian-stable binary/
        mode: '0644'

    - name: Update apt cache again after adding Jenkins repo
      apt:
        update_cache: yes

    - name: Install Jenkins
      apt:
        name: jenkins
        state: present

    - name: Ensure Jenkins is started and enabled
      systemd:
        name: jenkins
        state: started
        enabled: yes
```

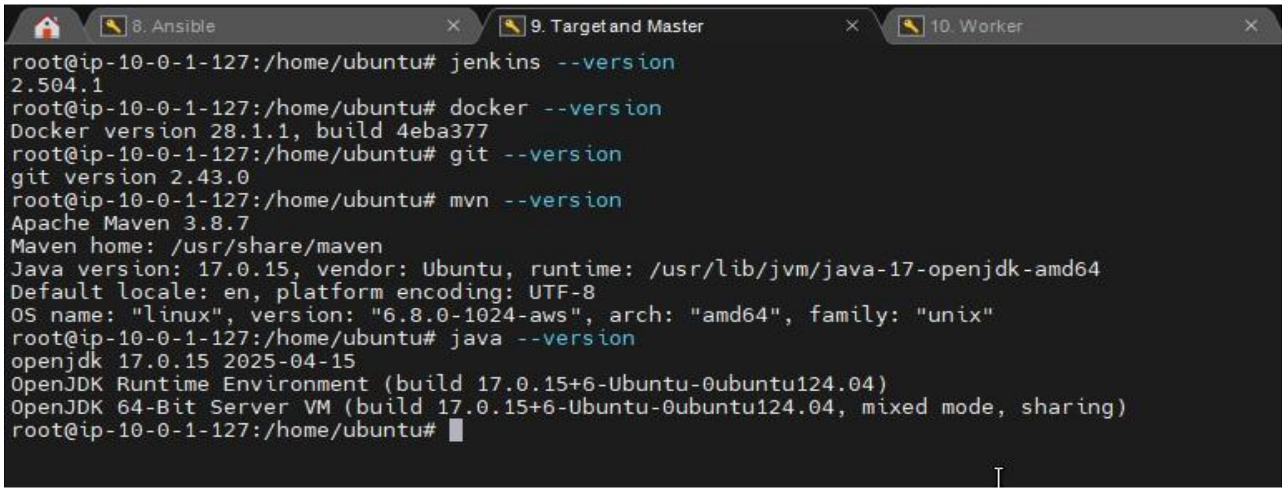
Step20:- Now run the yaml files to install packages in the target and master node and go to target and master node and verify the packages



```
TASK [Gathering Facts] ****************************************************************************************
[WARNING]: Platform linux on host 10.0.1.127 is using the discovered Python interpreter at /usr/bin/python3.12, but fut
another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [10.0.1.127]

TASK [Update apt cache] ***************************************************************************************
changed: [10.0.1.127]

TASK [Install dependencies] ***********************************************************************************
changed: [10.0.1.127]

TASK [Install Java 17] ****************************************************************************************
changed: [10.0.1.127]

TASK [Install Maven] ******************************************************************************************
changed: [10.0.1.127]

TASK [Install Git] ********************************************************************************************
ok: [10.0.1.127]

TASK [Add Docker GPG key] *************************************************************************************
changed: [10.0.1.127]

TASK [Add Docker repository] **********************************************************************************
changed: [10.0.1.127]

TASK [Install Docker] *****************************************************************************************
changed: [10.0.1.127]

TASK [Enable and start Docker service] ************************************************************************
ok: [10.0.1.127]

PLAY RECAP ****************************************************************************************************
10.0.1.127                 : ok=10   changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-10-0-1-135:/home/ubuntu# 
```

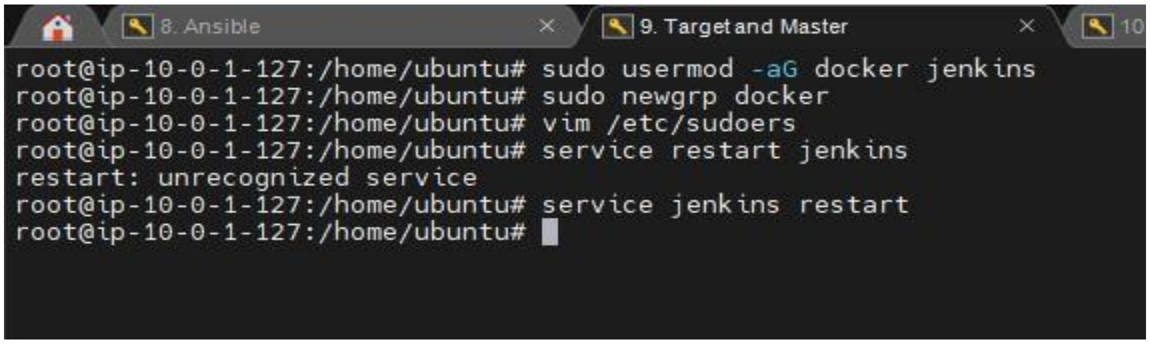


```
root@ip-10-0-1-135:/home/ubuntu# ansible-playbook jenkins.yml

PLAY [Install Jenkins on target node (Ubuntu 20.04+)] *****************************************************************************

TASK [Gathering Facts] ************************************************************************************************************
[WARNING]: Platform linux on host 10.0.1.127 is using the discovered Python interpreter at /usr/bin/python3.12, but future installation of
another Python interpreter could change the meaning of that path. See https://docs.ansible.com/ansible-
core/2.18/reference_appendices/interpreter_discovery.html for more information.
ok: [10.0.1.127]

TASK [Update apt cache] ***********************************************************************************************************
changed: [10.0.1.127]

TASK [Download Jenkins GPG key] ***************************************************************************************************
ok: [10.0.1.127]

TASK [Add Jenkins repository with signed-by] **************************************************************************************
changed: [10.0.1.127]

TASK [Update apt cache again after adding Jenkins repo] ***************************************************************************
changed: [10.0.1.127]

TASK [Install Jenkins] ************************************************************************************************************
changed: [10.0.1.127]

TASK [Ensure Jenkins is started and enabled] **************************************************************************************
ok: [10.0.1.127]

PLAY RECAP ************************************************************************************************************************
10.0.1.127                 : ok=7    changed=4    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0

root@ip-10-0-1-135:/home/ubuntu# 
```



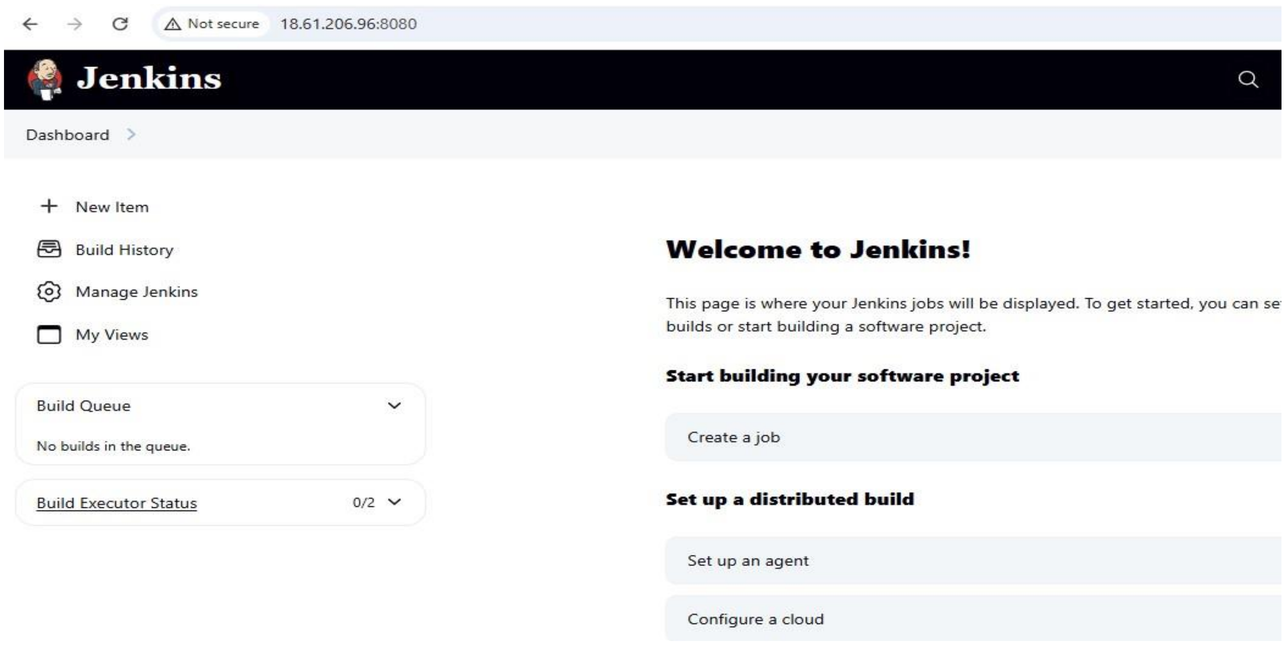
```
root@ip-10-0-1-127:/home/ubuntu# jenkins --version
2.504.1
root@ip-10-0-1-127:/home/ubuntu# docker --version
Docker version 28.1.1, build 4eba377
root@ip-10-0-1-127:/home/ubuntu# git --version
git version 2.43.0
root@ip-10-0-1-127:/home/ubuntu# mvn --version
Apache Maven 3.8.7
Maven home: /usr/share/maven
Java version: 17.0.15, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1024-aws", arch: "amd64", family: "unix"
root@ip-10-0-1-127:/home/ubuntu# java --version
openjdk 17.0.15 2025-04-15
OpenJDK Runtime Environment (build 17.0.15+6-Ubuntu-0ubuntu124.04)
OpenJDK 64-Bit Server VM (build 17.0.15+6-Ubuntu-0ubuntu124.04, mixed mode, sharing)
root@ip-10-0-1-127:/home/ubuntu# 
```

Step21:- Now add Jenkins group to docker and give root permissions to the Jenkins user in the sudoers file as under root give jenkins ALL=(ALL:ALL) NOAPSSWD: ALL restart the jenkins



```
root@ip-10-0-1-127:/home/ubuntu# sudo usermod -aG docker jenkins
root@ip-10-0-1-127:/home/ubuntu# sudo newgrp docker
root@ip-10-0-1-127:/home/ubuntu# vim /etc/sudoers
root@ip-10-0-1-127:/home/ubuntu# service restart jenkins
restart: unrecognized service
root@ip-10-0-1-127:/home/ubuntu# service jenkins restart
root@ip-10-0-1-127:/home/ubuntu#
```

Step22:- Go to the any browser and give the details and click on recommended plugins and login to the Jenkins



No builds in the queue.

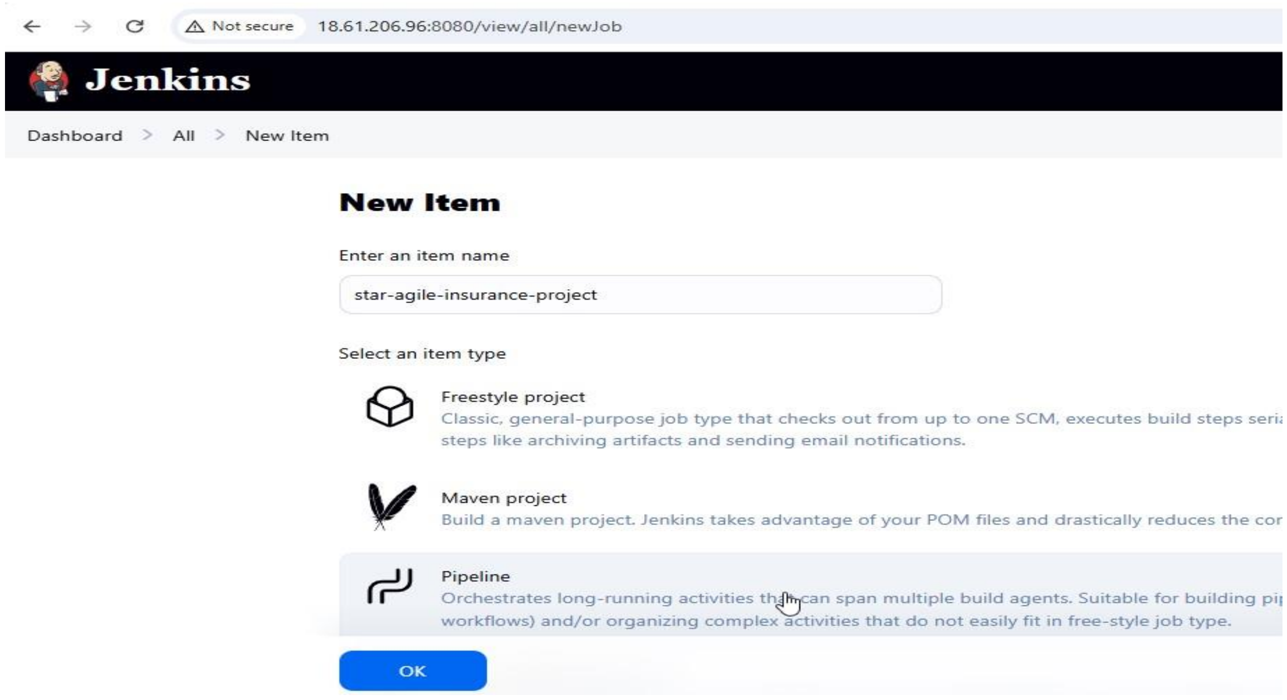
Welcome to Jenkins!

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

Start building your software project

Set up a distributed build

Step23:- Now in the Jenkins dashboard click on new item an give any name and select pipeline project as type and click on ok


Not secure 18.61.206.96:8080/view/all/newJob
Jenkins
Dashboard > All > New Item

New Item

Enter an item name

star-agile-insurance-project

Select an item type

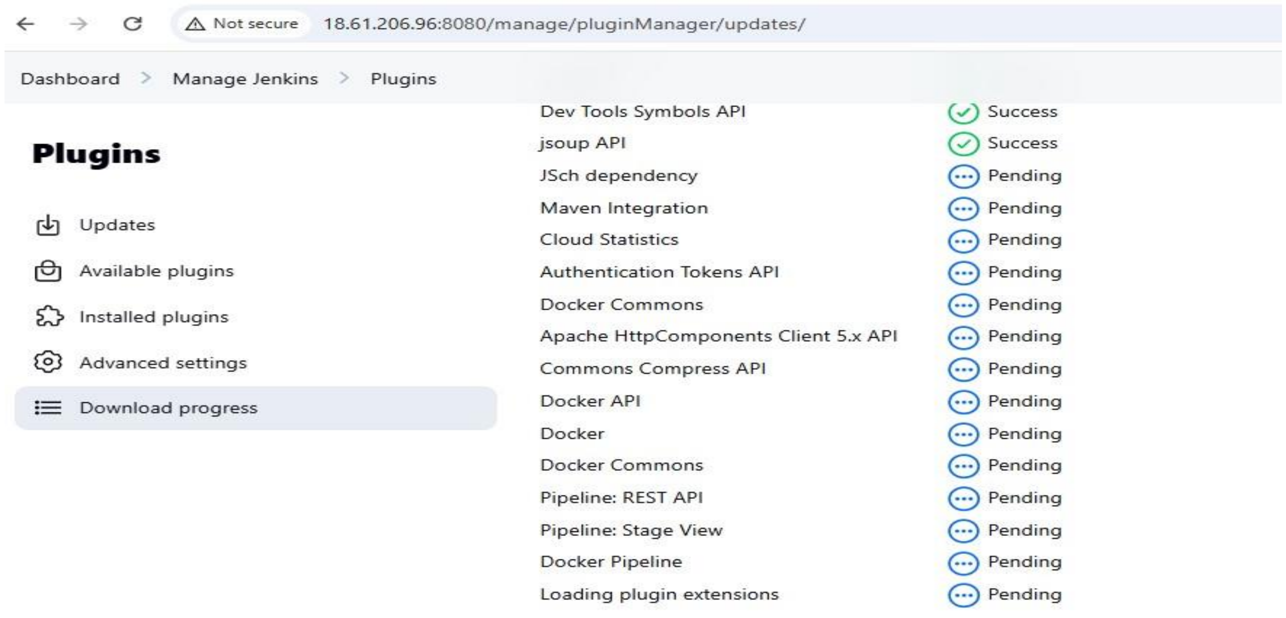
Freestyle project
Classic, general-purpose job type that checks out from up to one SCM, executes build steps seri
steps like archiving artifacts and sending email notifications.

Maven project
Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the cor

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

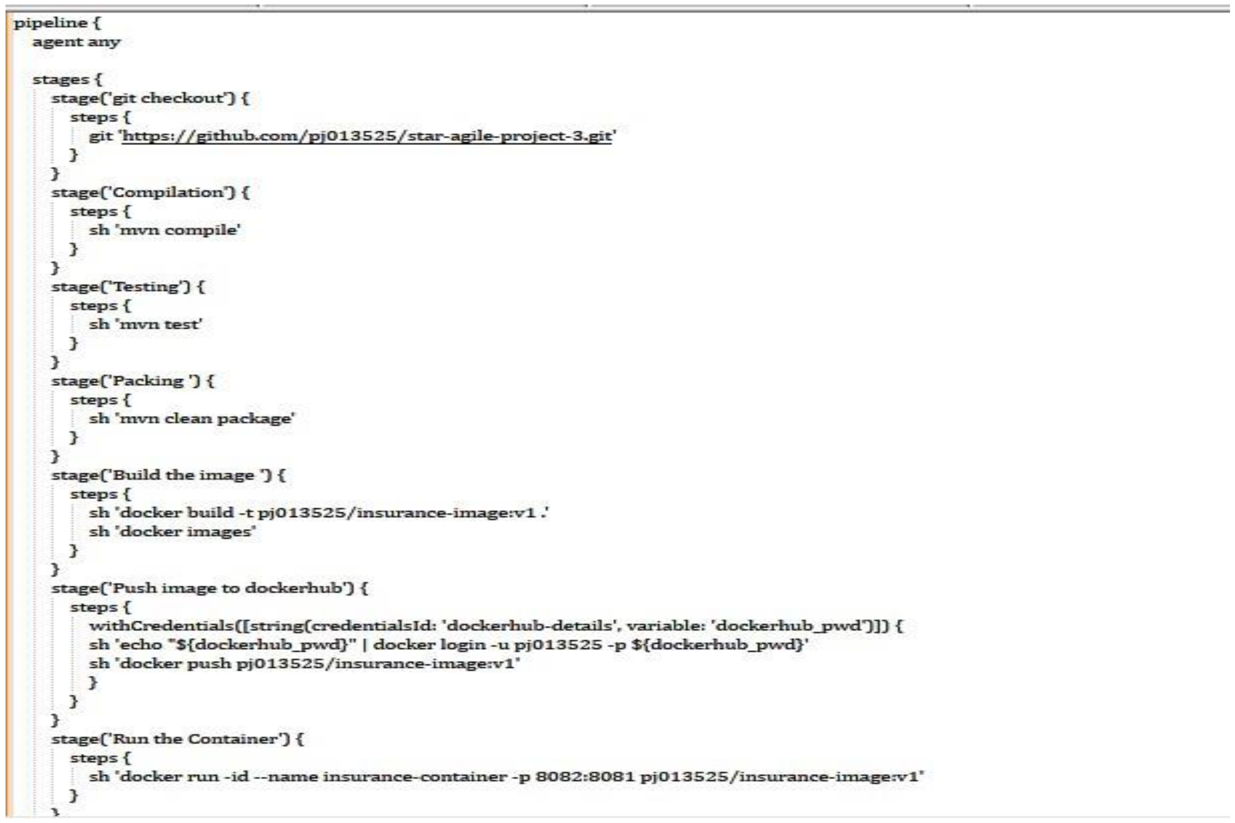
Step24:- Now install docker and other required plugins in the Jenkins

Pipeline stage view	Maven Integration Plugin
Git Plugin	Docker Commons Plugin
Docker Pipeline Plugin	Pipeline: GitHub
Credentials Binding Plugin	


Not secure 18.61.206.96:8080/manage/pluginManager/updates/
Dashboard > Manage Jenkins > Plugins
Plugins
Updates
Available plugins
Installed plugins
Advanced settings
Download progress

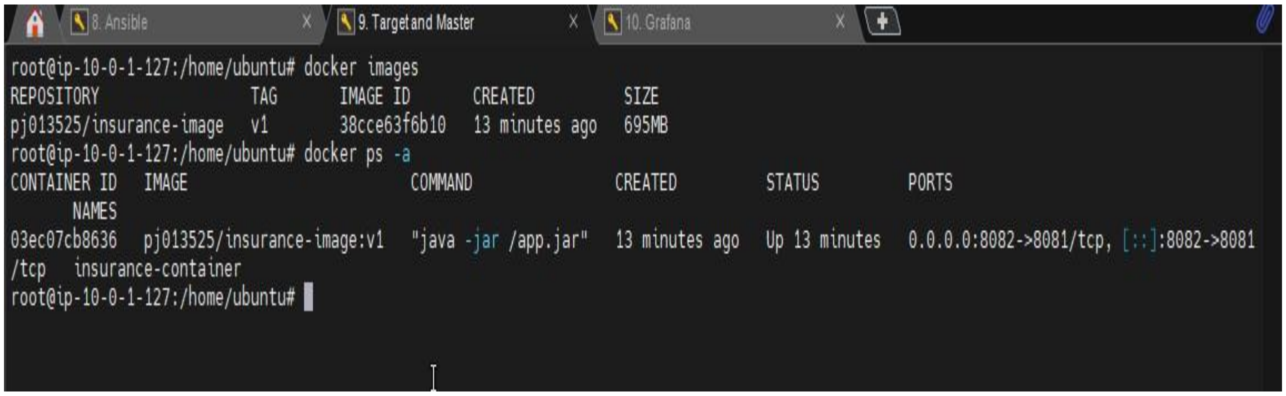
Dev Tools Symbols API	Success
jsoup API	Success
JSch dependency	Pending
Maven Integration	Pending
Cloud Statistics	Pending
Authentication Tokens API	Pending
Docker Commons	Pending
Apache HttpComponents Client 5.x API	Pending
Commons Compress API	Pending
Docker API	Pending
Docker	Pending
Docker Commons	Pending
Pipeline: REST API	Pending
Pipeline: Stage View	Pending
Docker Pipeline	Pending
Loading plugin extensions	Pending

Step25:- Now to perform the pipeline in the Jenkins go to Project==> Pipeline ==> pipeline script and write pipeline using groovy script

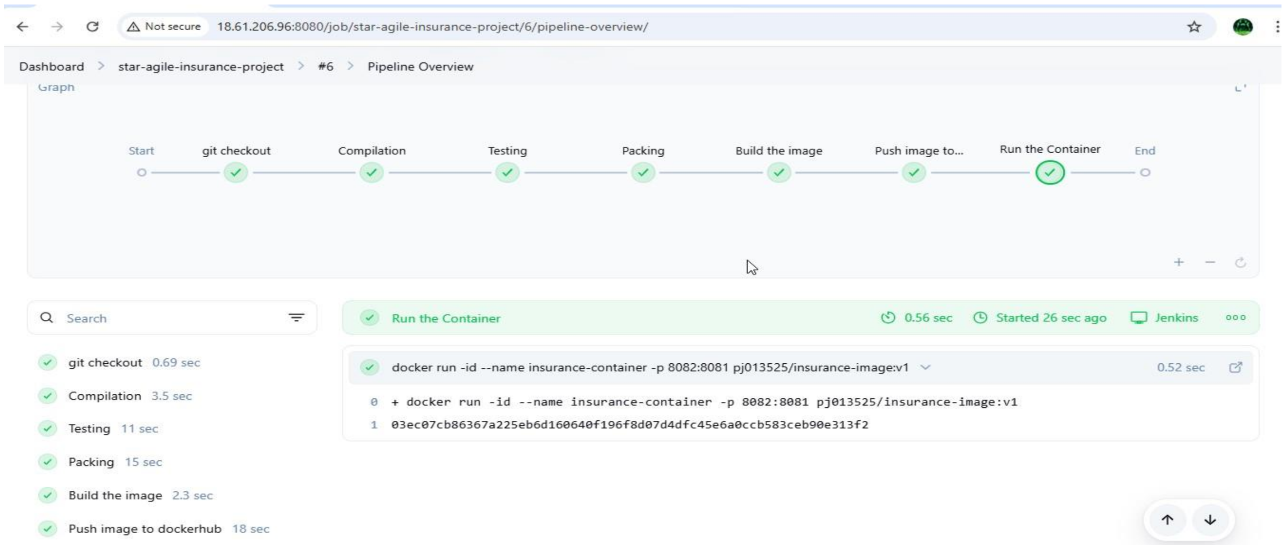


```
pipeline {
  agent any

  stages {
    stage('git checkout') {
      steps {
        git 'https://github.com/pj013525/star-agile-project-3.git'
      }
    }
    stage('Compilation') {
      steps {
        sh 'mvn compile'
      }
    }
    stage('Testing') {
      steps {
        sh 'mvn test'
      }
    }
    stage('Packing ') {
      steps {
        sh 'mvn clean package'
      }
    }
    stage('Build the image ') {
      steps {
        sh 'docker build -t pj013525/insurance-image:v1 .'
        sh 'docker images'
      }
    }
    stage('Push image to dockerhub') {
      steps {
        withCredentials([string(credentialsId: 'dockerhub-details', variable: 'dockerhub_pwd')]) {
        sh 'echo "${dockerhub_pwd}" | docker login -u pj013525 -p ${dockerhub_pwd}'
        sh 'docker push pj013525/insurance-image:v1'
        }
      }
    }
    stage('Run the Container') {
      steps {
        sh 'docker run -id --name insurance-container -p 8082:8081 pj013525/insurance-image:v1'
      }
    }
```

Step26:- Now again go back to Jenkins project and click on Build now to check the status of the build and as you can see that the build is successful and a container also created in the ec2 Target and Master node.

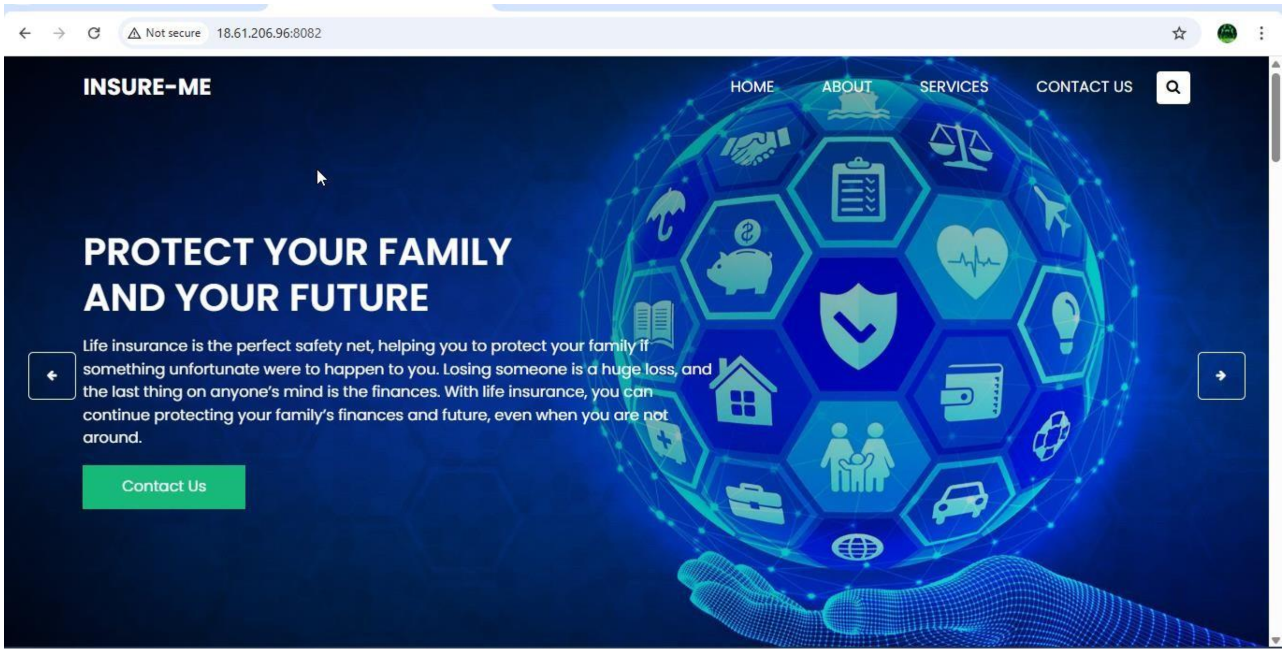


```
root@ip-10-0-1-127:/home/ubuntu# docker images
REPOSITORY                TAG       IMAGE ID       CREATED          SIZE
pj013525/insurance-image   v1       38cce63f6b10   13 minutes ago   695MB
root@ip-10-0-1-127:/home/ubuntu# docker ps -a
CONTAINER ID   IMAGE                         COMMAND               CREATED          STATUS          PORTS
      NAMES
03ec07cb8636   pj013525/insurance-image:v1   "java -jar /app.jar"   13 minutes ago   Up 13 minutes   0.0.0.0:8082->8081/tcp, [::]:8082->8081
/tcp   insurance-container
root@ip-10-0-1-127:/home/ubuntu#
```


Not secure 18.61.206.96:8080/job/star-agile-insurance-project/6/pipeline-overview/
Dashboard > star-agile-insurance-project > #6 > Pipeline Overview
Graph
Start
git checkout
Compilation
Testing
Packing
Build the image
Push image to...
Run the Container
End
Search
Run the Container
0.56 sec
Started 26 sec ago
Jenkins
git checkout 0.69 sec
Compilation 3.5 sec
Testing 11 sec
Packing 15 sec
Build the image 2.3 sec
Push image to dockerhub 18 sec
docker run -id --name insurance-container -p 8082:8081 pj013525/insurance-image:v1
0.52 sec

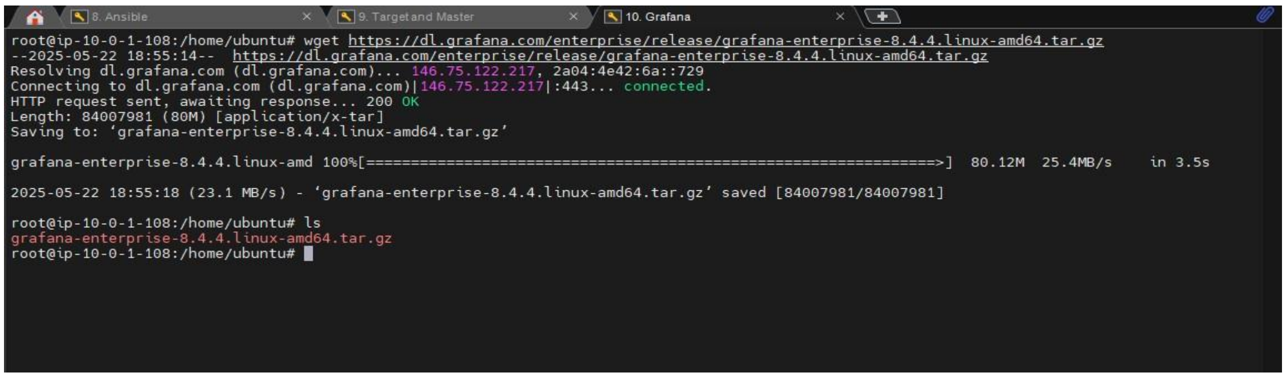
```
+ docker run -id --name insurance-container -p 8082:8081 pj013525/insurance-image:v1
03ec07cb86367a225eb6d160640f196f8d07d4dfc45e6a0ccb583ceb90e313f2
```

Step27:- Now go to any browser and give the <Target and MasternodeIP:cont-port> and click enter the you will see the home page of the project and thus the project deploy is successful using docker container.

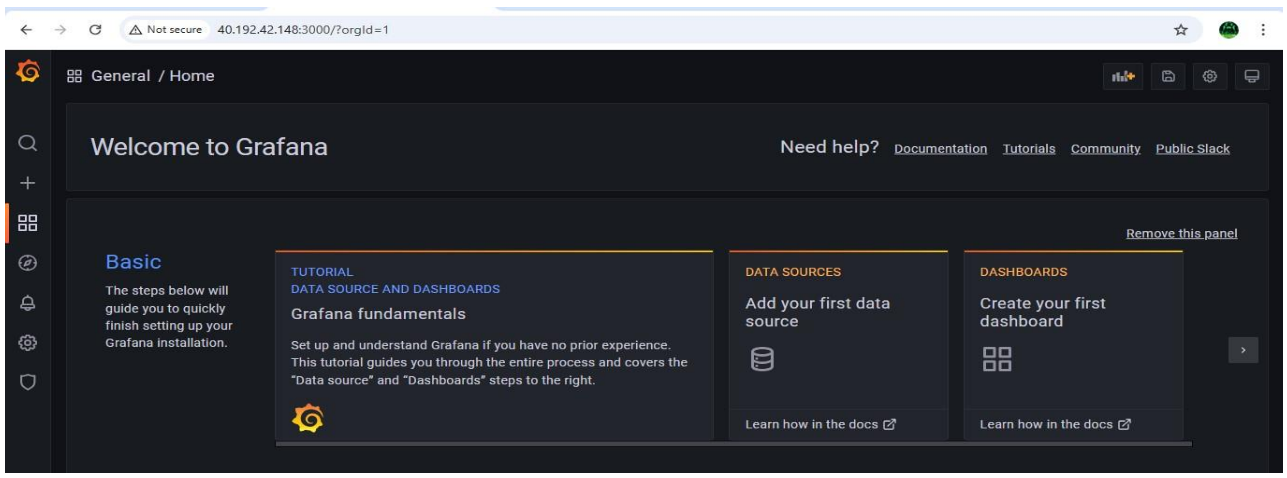

Not secure 18.61.206.96:8082
INSURE-ME
HOME
ABOUT
SERVICES
CONTACT US
PROTECT YOUR FAMILY AND YOUR FUTURE
Life insurance is the perfect safety net, helping you to protect your family if something unfortunate were to happen to you. Losing someone is a huge loss, and the last thing on anyone's mind is the finances. With life insurance, you can continue protecting your family's finances and future, even when you are not around.
Contact Us

Step28:- Now monitor the containers using Prometheus and Grafana , for that install Prometheus in Jenkins-Docker server and Grafana in another server

Step29:- Now install grfana in the Grafana server and after successful installation of Grafana, now go to browser and give grafana server ipaddress:3000 (3000 is default port number for grafana) and use admin and admin as username and password as they are default and login to the grafana home page.



```
root@ip-10-0-1-108:/home/ubuntu# wget https://dl.grafana.com/enterprise/release/grafana-enterprise-8.4.4.linux-amd64.tar.gz
--2025-05-22 18:55:14--  https://dl.grafana.com/enterprise/release/grafana-enterprise-8.4.4.linux-amd64.tar.gz
Resolving dl.grafana.com (dl.grafana.com)... 146.75.122.217, 2a04:4e42:6a::729
Connecting to dl.grafana.com (dl.grafana.com)|146.75.122.217|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 84007981 (80M) [application/x-tar]
Saving to: ‘grafana-enterprise-8.4.4.linux-amd64.tar.gz’

grafana-enterprise-8.4.4.linux-amd 100%[=================================================================>]  80.12M  25.4MB/s    in 3.5s

2025-05-22 18:55:18 (23.1 MB/s) - ‘grafana-enterprise-8.4.4.linux-amd64.tar.gz’ saved [84007981/84007981]

root@ip-10-0-1-108:/home/ubuntu# ls
grafana-enterprise-8.4.4.linux-amd64.tar.gz
root@ip-10-0-1-108:/home/ubuntu# 
```



Welcome to Grafana

Basic

The steps below will guide you to quickly finish setting up your Grafana installation.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

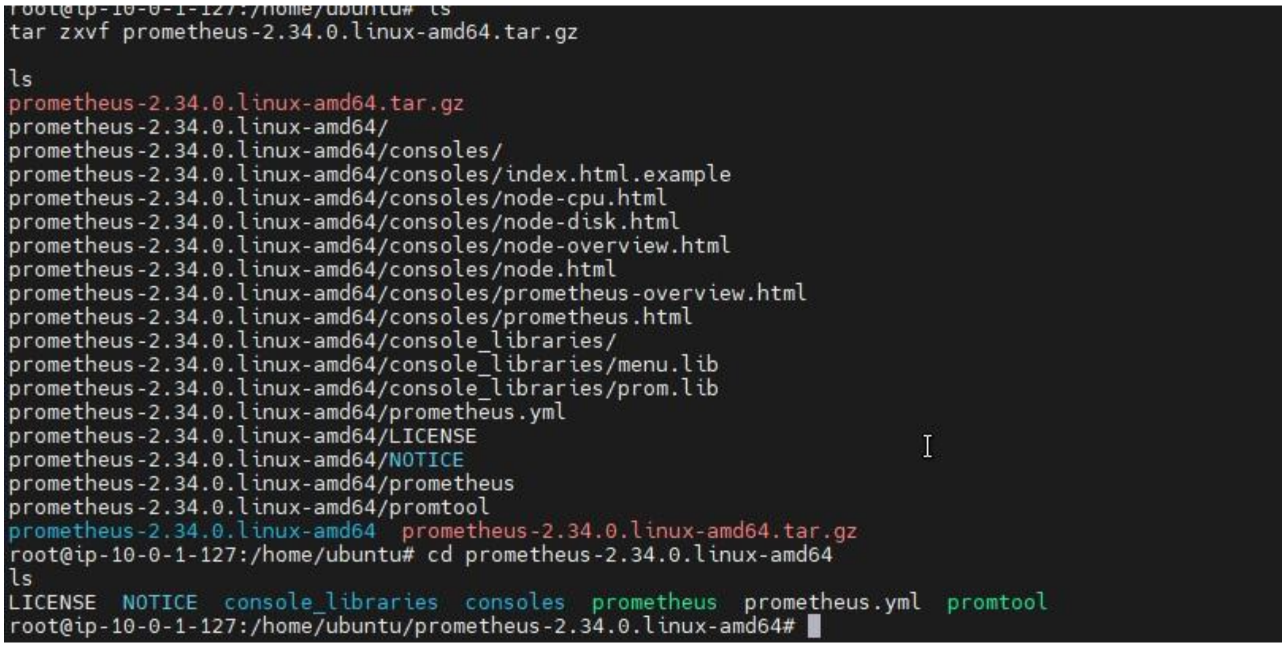
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

DATA SOURCES
Add your first data source

DASHBOARDS
Create your first dashboard

Step30:- Now install prometheus in Target and master node and to login to the prometheus homepage first give metric address in the docker daemon.json   vi /etc/docker/daemon.json

```
{
 "metrics-addr" : "0.0.0.0:9323",
 "experimental" : true
}
```



```
root@ip-10-0-1-127:/home/ubuntu# ls
tar zxvf prometheus-2.34.0.linux-amd64.tar.gz

ls
prometheus-2.34.0.linux-amd64.tar.gz
prometheus-2.34.0.linux-amd64/
prometheus-2.34.0.linux-amd64/consoles/
prometheus-2.34.0.linux-amd64/consoles/index.html.example
prometheus-2.34.0.linux-amd64/consoles/node-cpu.html
prometheus-2.34.0.linux-amd64/consoles/node-disk.html
prometheus-2.34.0.linux-amd64/consoles/node-overview.html
prometheus-2.34.0.linux-amd64/consoles/node.html
prometheus-2.34.0.linux-amd64/consoles/prometheus-overview.html
prometheus-2.34.0.linux-amd64/consoles/prometheus.html
prometheus-2.34.0.linux-amd64/console_libraries/
prometheus-2.34.0.linux-amd64/console_libraries/menu.lib
prometheus-2.34.0.linux-amd64/console_libraries/prom.lib
prometheus-2.34.0.linux-amd64/prometheus.yml
prometheus-2.34.0.linux-amd64/LICENSE
prometheus-2.34.0.linux-amd64/NOTICE
prometheus-2.34.0.linux-amd64/prometheus
prometheus-2.34.0.linux-amd64/promtool
prometheus-2.34.0.linux-amd64  prometheus-2.34.0.linux-amd64.tar.gz
root@ip-10-0-1-127:/home/ubuntu# cd prometheus-2.34.0.linux-amd64
ls
LICENSE  NOTICE  console_libraries  consoles  prometheus  prometheus.yml  promtool
root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64#
```

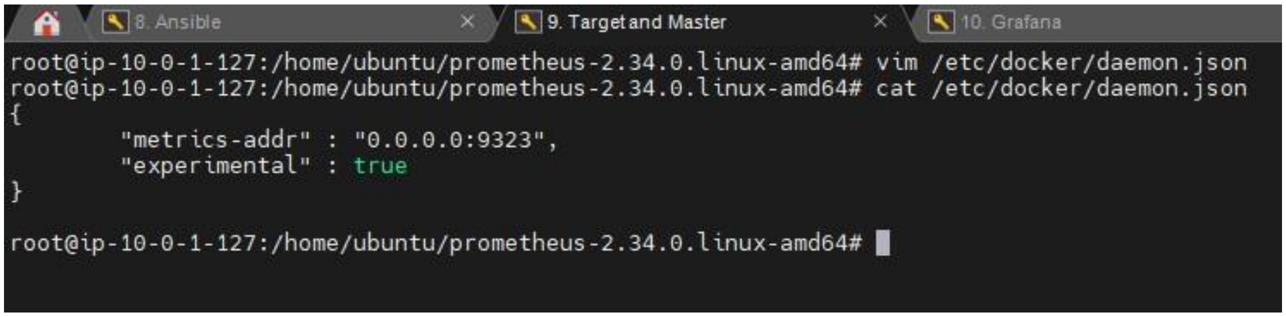
Step31:- Now setup the docker and Prometheus in another using by telling docker that Prometheus would track docker on port 9323

i.e., vi /etc/docker/daemon.json press

I to insert

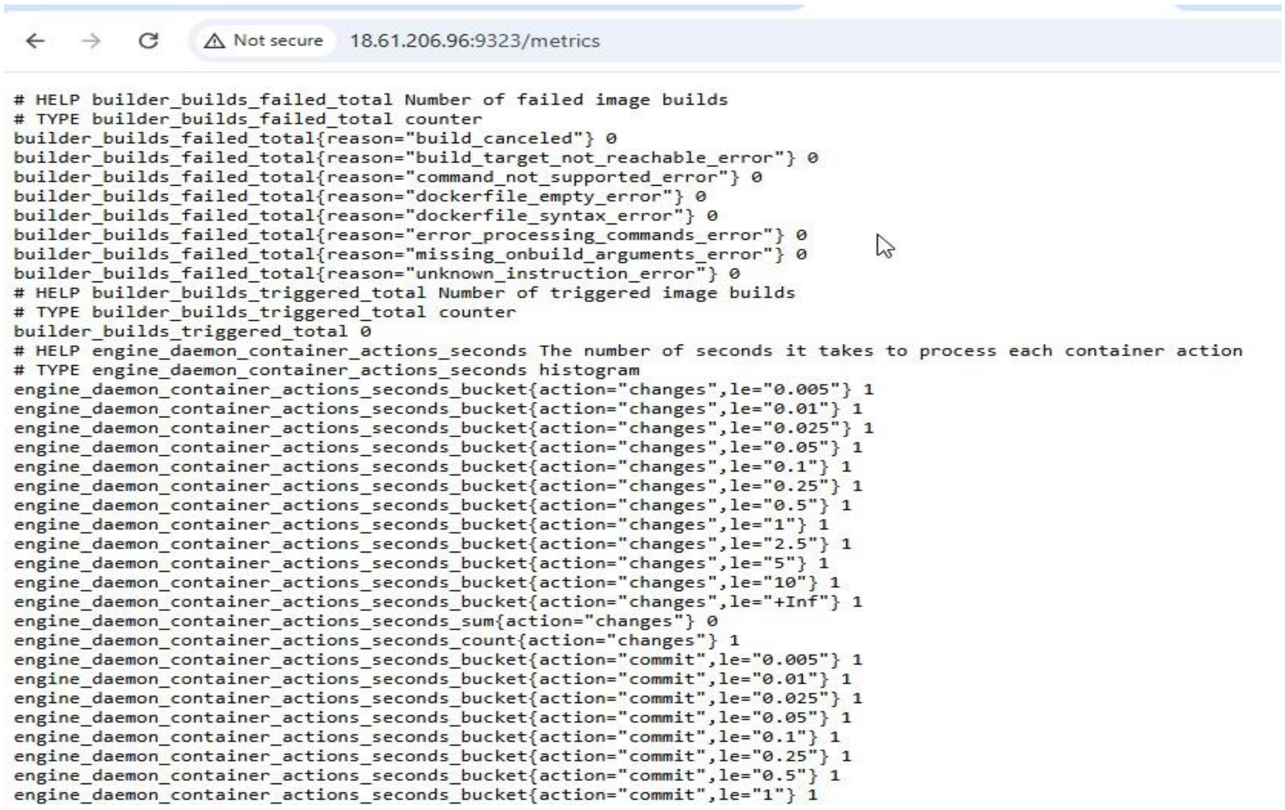
```
{
      "metrics-addr" : "0.0.0.0:9323",
      "experimental" : true
}
```

then save and exit and restart the docker



```
root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64# vim /etc/docker/daemon.json
root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64# cat /etc/docker/daemon.json
{
        "metrics-addr" : "0.0.0.0:9323",
        "experimental" : true
}

root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64#
```

Step32:- Now go to any browser and give docker ip-address:9323/metrics and in the below image you will see that the docker stats have been started successfully

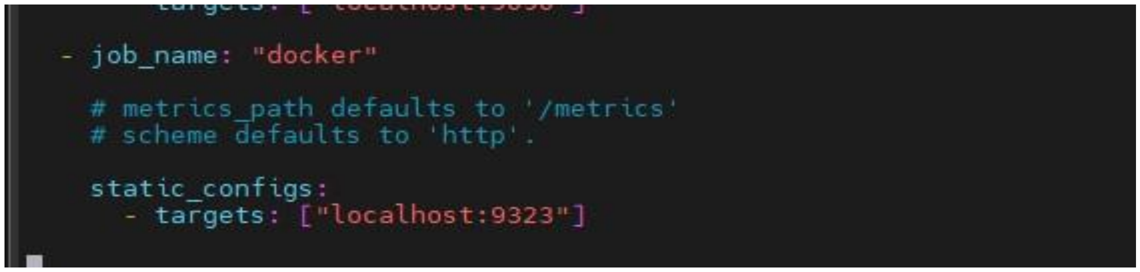


```
18.61.206.96:9323/metrics

# HELP builder_builds_failed_total Number of failed image builds
# TYPE builder_builds_failed_total counter
builder_builds_failed_total{reason="build_canceled"} 0
builder_builds_failed_total{reason="build_target_not_reachable_error"} 0
builder_builds_failed_total{reason="command_not_supported_error"} 0
builder_builds_failed_total{reason="dockerfile_empty_error"} 0
builder_builds_failed_total{reason="dockerfile_syntax_error"} 0
builder_builds_failed_total{reason="error_processing_commands_error"} 0
builder_builds_failed_total{reason="missing_onbuild_arguments_error"} 0
builder_builds_failed_total{reason="unknown_instruction_error"} 0
# HELP builder_builds_triggered_total Number of triggered image builds
# TYPE builder_builds_triggered_total counter
builder_builds_triggered_total 0
# HELP engine_daemon_container_actions_seconds The number of seconds it takes to process each container action
# TYPE engine_daemon_container_actions_seconds histogram
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="1"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="2.5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="5"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="10"} 1
engine_daemon_container_actions_seconds_bucket{action="changes",le="+Inf"} 1
engine_daemon_container_actions_seconds_sum{action="changes"} 0
engine_daemon_container_actions_seconds_count{action="changes"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.005"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.01"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.025"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.05"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.1"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.25"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="0.5"} 1
engine_daemon_container_actions_seconds_bucket{action="commit",le="1"} 1
```

Step33:- Now add docker job in the Prometheus.yml file to give this stats to Prometheus vi prometheus.yml

```
 - job_name: "docker"

  # metrics_path defaults to '/metrics'
# scheme defaults to 'http'.
static_configs:
 - targets: ["localhost:9323"]
```

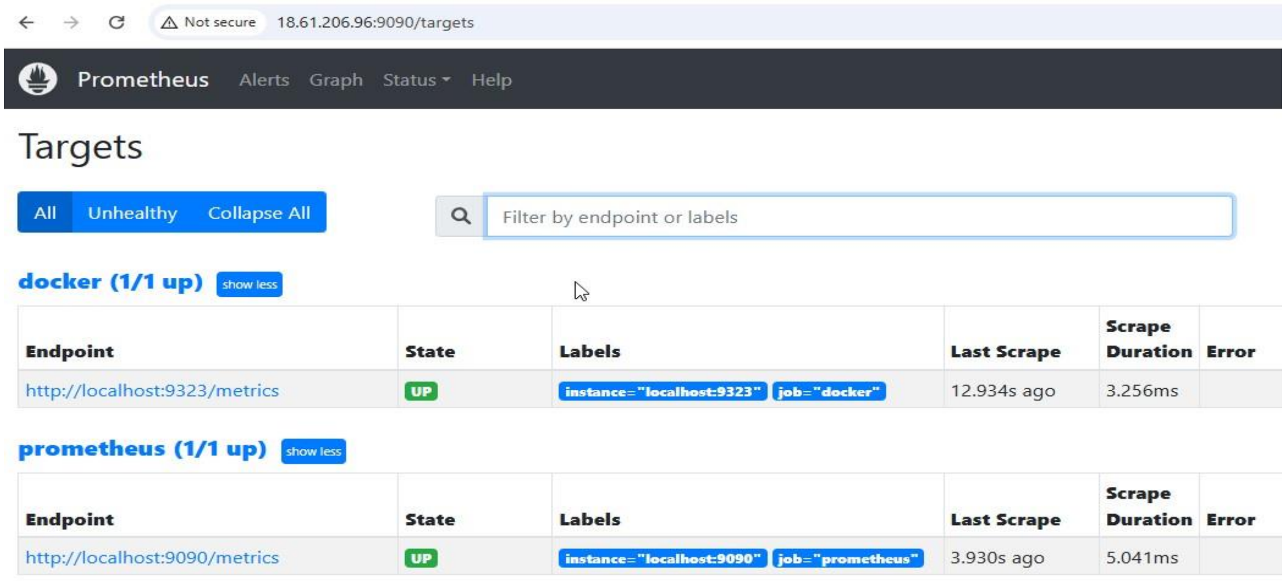
Save the file and exit and start the Prometheus using ./prometheus



```
  - job_name: "docker"

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
      - targets: ["localhost:9323"]
```

As you can see that the Prometheus have been started from the above image.

Step34:- Now go browser and give docker ip:9090 and enter , then you will be successfully enter into the Prometheus homepage and click on status ❼ targets then you will see the status of the of the docker and prometheus.


Not secure 18.61.206.96:9090/targets

Prometheus Alerts Graph Status Help

Targets

All Unhealthy Collapse All

Filter by endpoint or labels

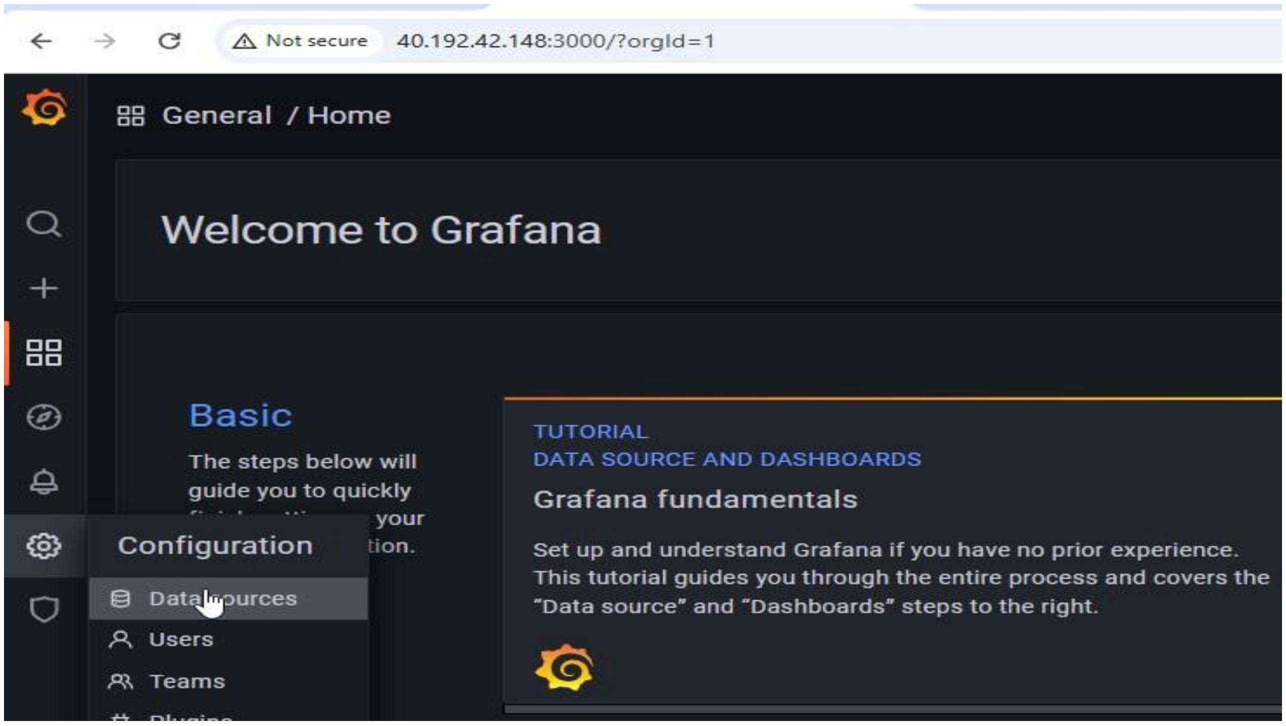
docker (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9323/metrics	UP	instance="localhost:9323" job="docker"	12.934s ago	3.256ms	

prometheus (1/1 up) show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	3.930s ago	5.041ms	

Step35:- Now go to grafana homepage ❼ configurations ❼ Data sources


Not secure 40.192.42.148:3000/?orgId=1

General / Home

Welcome to Grafana

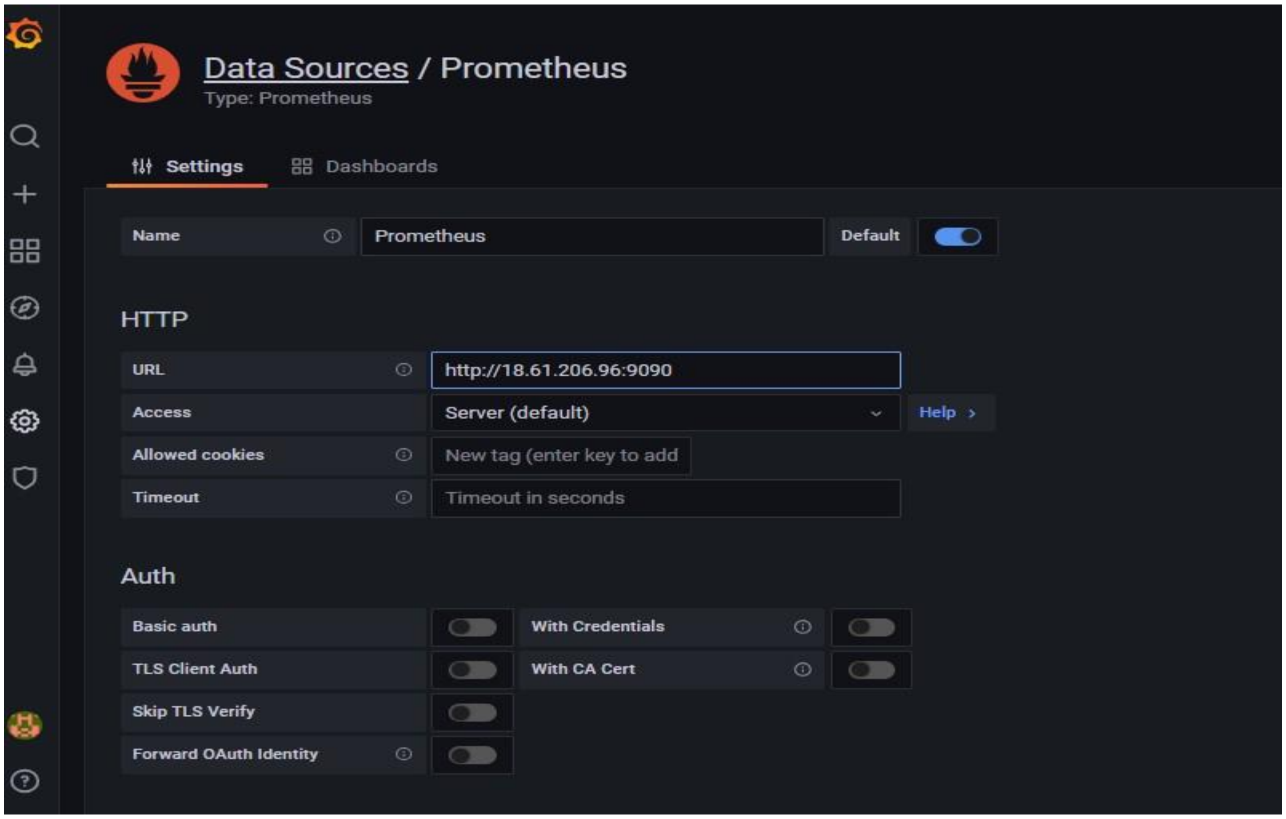
Basic

The steps below will guide you to quickly your tion.

TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals

Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.

Step36:- Now click on add Data sources ❼ Prometheus and give ipaddress:9090 and click on save and test



Data Sources / Prometheus
Type: Prometheus

Settings | Dashboards

Name: Prometheus
Default: on

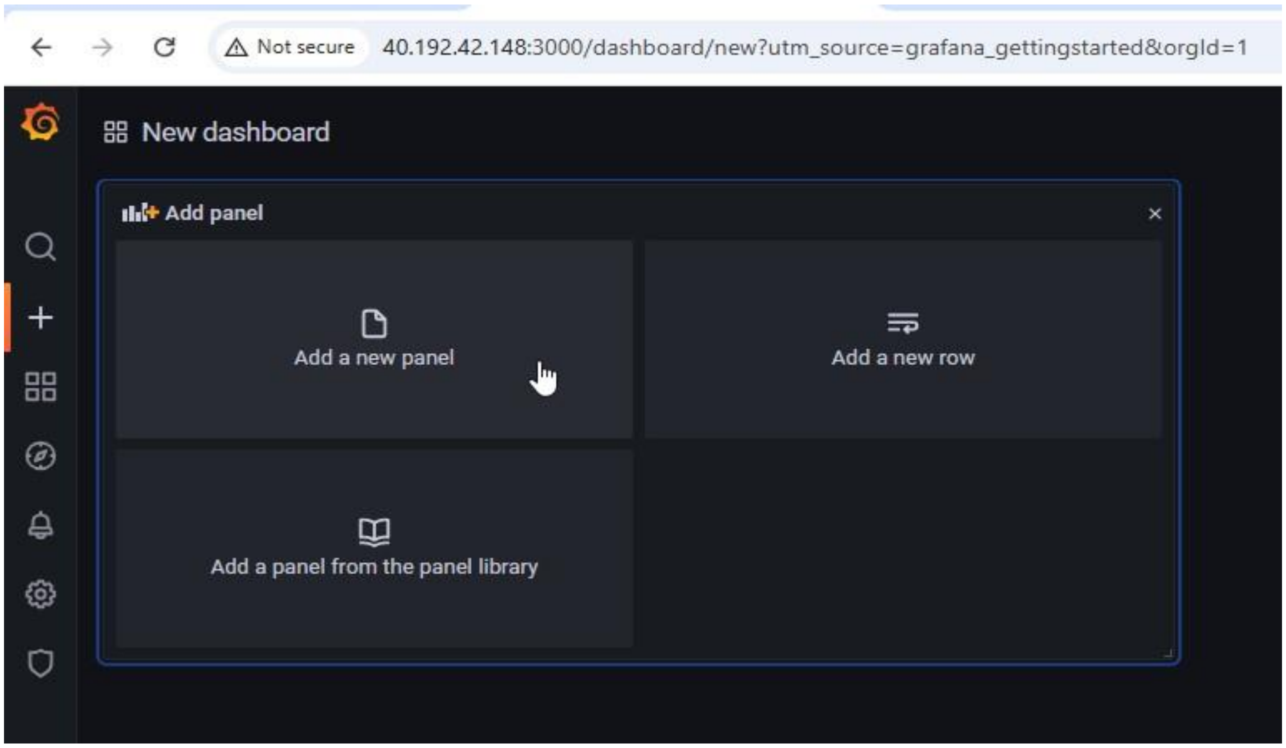
HTTP

URL: http://18.61.206.96:9090
Access: Server (default)
Allowed cookies: New tag (enter key to add
Timeout: Timeout in seconds

Auth

Basic auth: off
With Credentials: off
TLS Client Auth: off
With CA Cert: off
Skip TLS Verify: off
Forward OAuth Identity: off

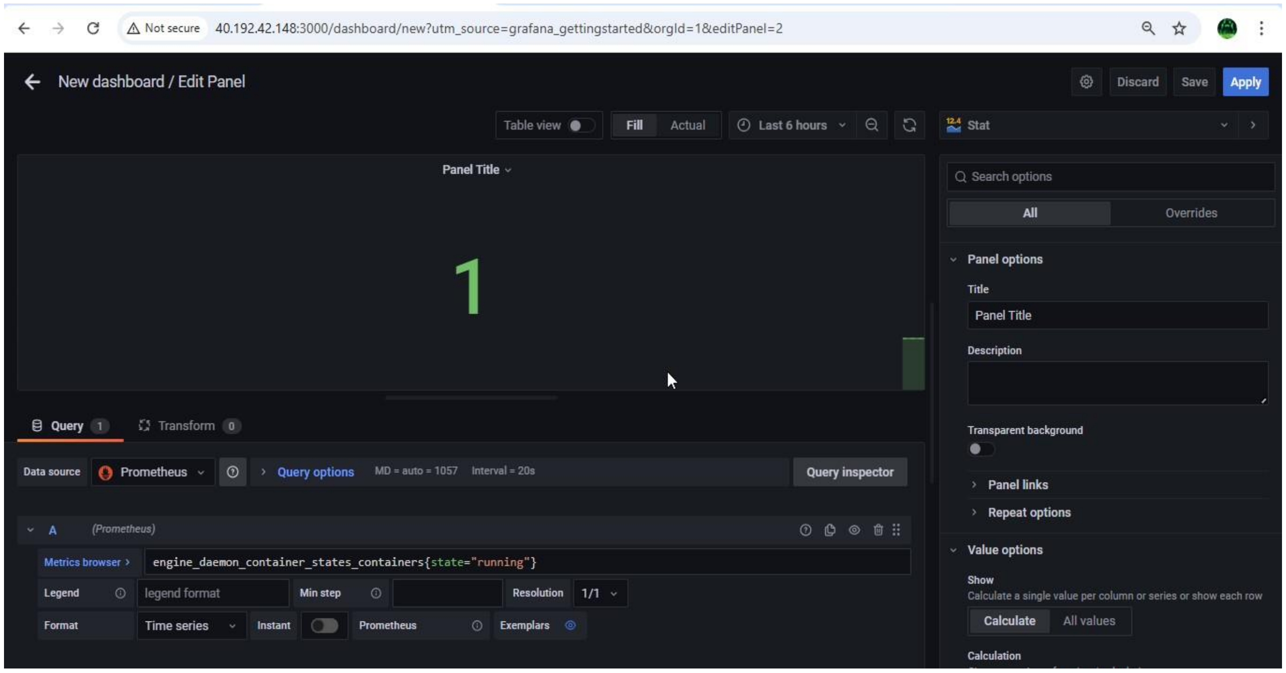
Step37:- Now click on Dash board ❼ add new panel



Not secure 40.192.42.148:3000/dashboard/new?utm_source=grafana_gettingstarted&orgId=1

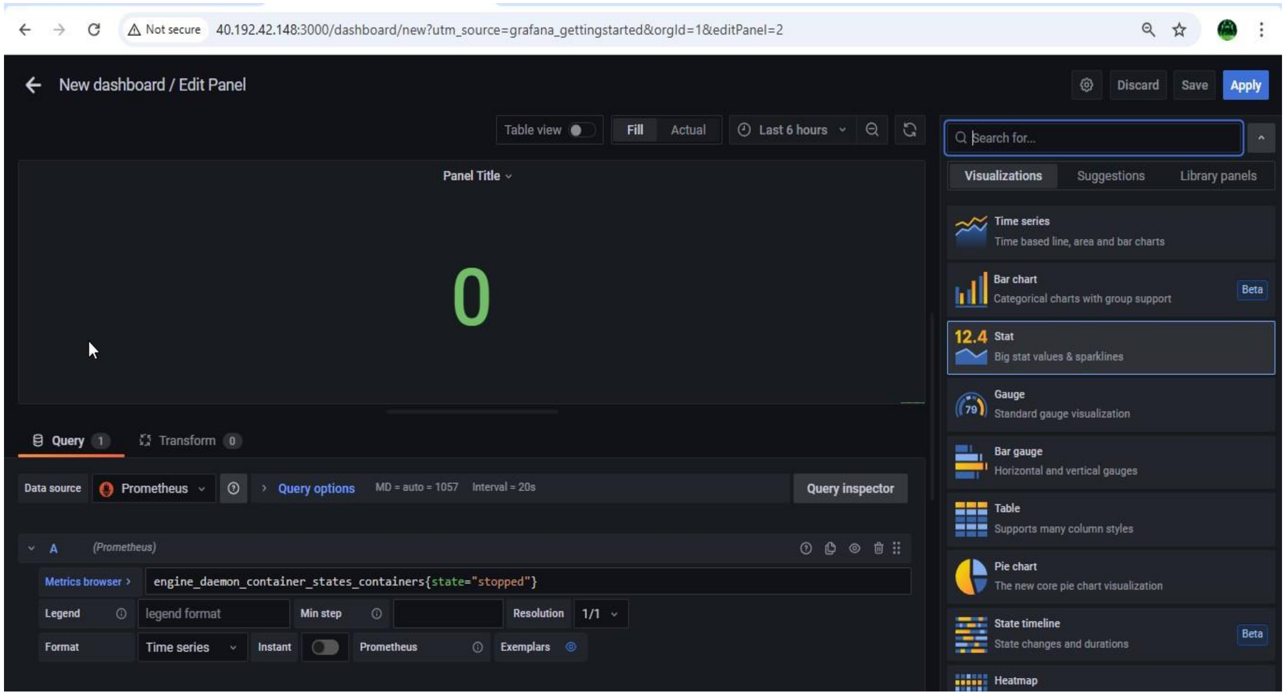
New dashboard

Step38:- Now in the metrics browser give engine daemon container states containers{state="running"} and you will see the result that same as in the metrics from the browser



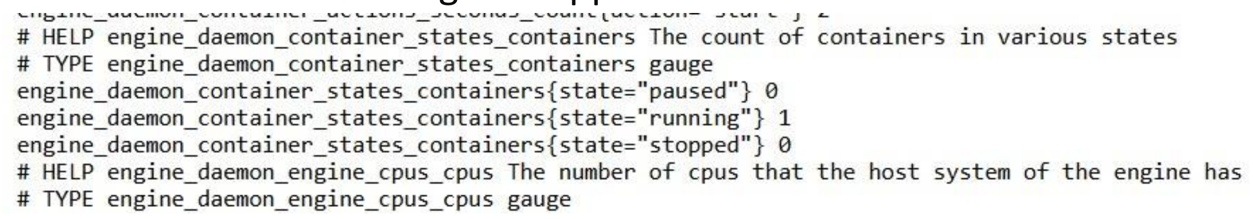
```
engine_daemon_container_states_containers{state="running"}
```

engine_daemon_container_states_containers{state="stopped"}



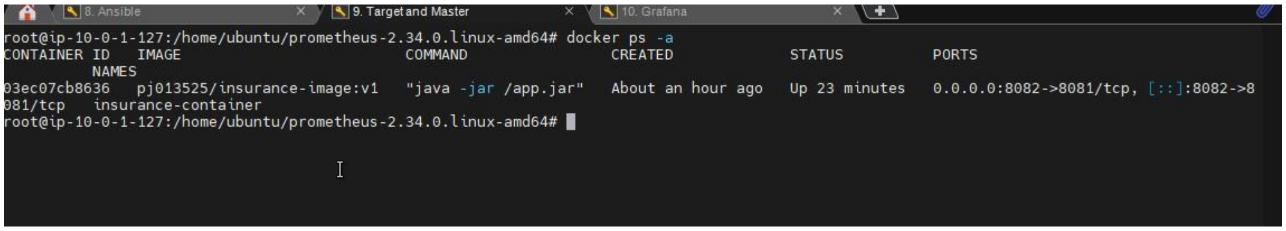
```
engine_daemon_container_states_containers{state="stopped"}
```

Step39:- The values shown in the panel must be equal to the that of shown in the docker stats, here the container which we created is in exited state so it is showing as stopped state in stats



```
# HELP engine_daemon_container_states_containers The count of containers in various states
# TYPE engine_daemon_container_states_containers gauge
engine_daemon_container_states_containers{state="paused"} 0
engine_daemon_container_states_containers{state="running"} 1
engine_daemon_container_states_containers{state="stopped"} 0
# HELP engine_daemon_engine_cpus_cpus The number of cpus that the host system of the engine has
# TYPE engine_daemon_engine_cpus_cpus gauge
```

Step40:- Now go and check the containers running or in stopped state again and check the details again in the stats



```
root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64# docker ps -a
CONTAINER ID   IMAGE                          COMMAND              CREATED             STATUS          PORTS
          NAMES
03ec07cb8636   pj013525/insurance-image:v1   "java -jar /app.jar"   About an hour ago   Up 23 minutes   0.0.0.0:8082->8081/tcp, [::]:8082->8
081/tcp   insurance-container
root@ip-10-0-1-127:/home/ubuntu/prometheus-2.34.0.linux-amd64#
```

Step41:- As you can see that the container is in running state and the stats is also shown the same. This is how we monitor the health of a container automatically and visualizing the report using Prometheus and Grafana.