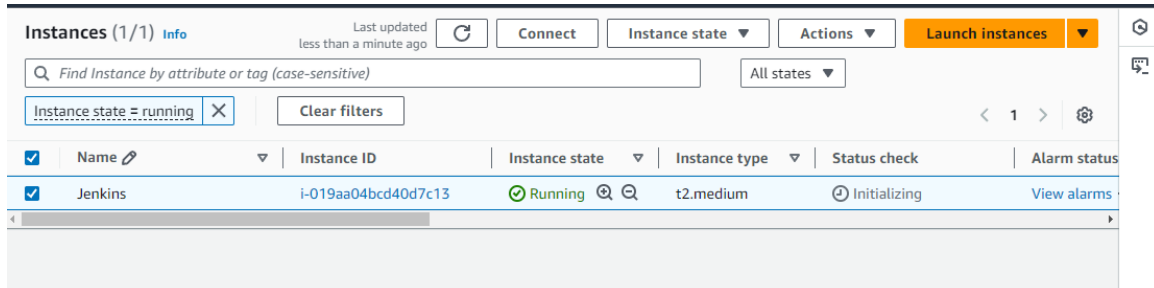


Banking and Finance Domain Project

Step1:

Create an instance named Jenkins



- Install Java and Maven which we need in the Jenkins tools part

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-172-31-35-56:~$ java -version
openjdk version "17.0.12" 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
ubuntu@ip-172-31-35-56:~$
```

i-019aa04bcd40d7c13 (Jenkins)
PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

- Maven install

```
root@ip-172-31-35-56:/home/ubuntu# mvn --version
Apache Maven 3.9.9 (8e8579a9e76f7d015ee5ec7bfcd097d260186937)
Maven home: /opt/maven
Java version: 17.0.12, vendor: Ubuntu, runtime: /usr/lib/jvm/java-17-openjdk-amd64
Default locale: en, platform encoding: UTF-8
OS name: "linux", version: "6.8.0-1016-aws", arch: "amd64", family: "unix"
root@ip-172-31-35-56:/home/ubuntu#
```

i-019aa04bcd40d7c13 (Jenkins)
PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

- Checking the path

```

root@ip-172-31-35-56:~# nano ~/.bashrc
root@ip-172-31-35-56:~# vi .bashrc
root@ip-172-31-35-56:~# nano ~/.bashrc
root@ip-172-31-35-56:~# source ~/.bashrc
root@ip-172-31-35-56:~# echo $JAVA_HOME
/usr/lib/jvm/java-17-openjdk-amd64
root@ip-172-31-35-56:~# echo $MAVEN_HOME
/opt/maven
root@ip-172-31-35-56:~#

```

i-019aa04bcd40d7c13 (Jenkins)

PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

- To see if maven is working

```

root@ip-172-31-35-56:~# mvn
[INFO] Scanning for projects...
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 0.083 s
[INFO] Finished at: 2024-10-28T06:33:54Z
[INFO] -----
[ERROR] No goals have been specified for this build. You must specify a valid lifecycle phase or a goal
plugin-group-id>:<plugin-artifact-id>[:<plugin-version>]:<goal>. Available lifecycle phases are: pre-c
ze, generate-sources, process-sources, generate-resources, process-resources, compile, process-classes
es, generate-test-resources, process-test-resources, test-compile, process-test-classes, test, prepare
tegration-test, post-integration-test, verify, install, deploy, pre-site, site, post-site, site-deploy
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following article
[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/NoGoalSpecifiedException
root@ip-172-31-35-56:~#

```

i-019aa04bcd40d7c13 (Jenkins)

PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

Step 2

- Installing jenkins

```

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

sudo systemctl enable jenkins

sudo systemctl start jenkins

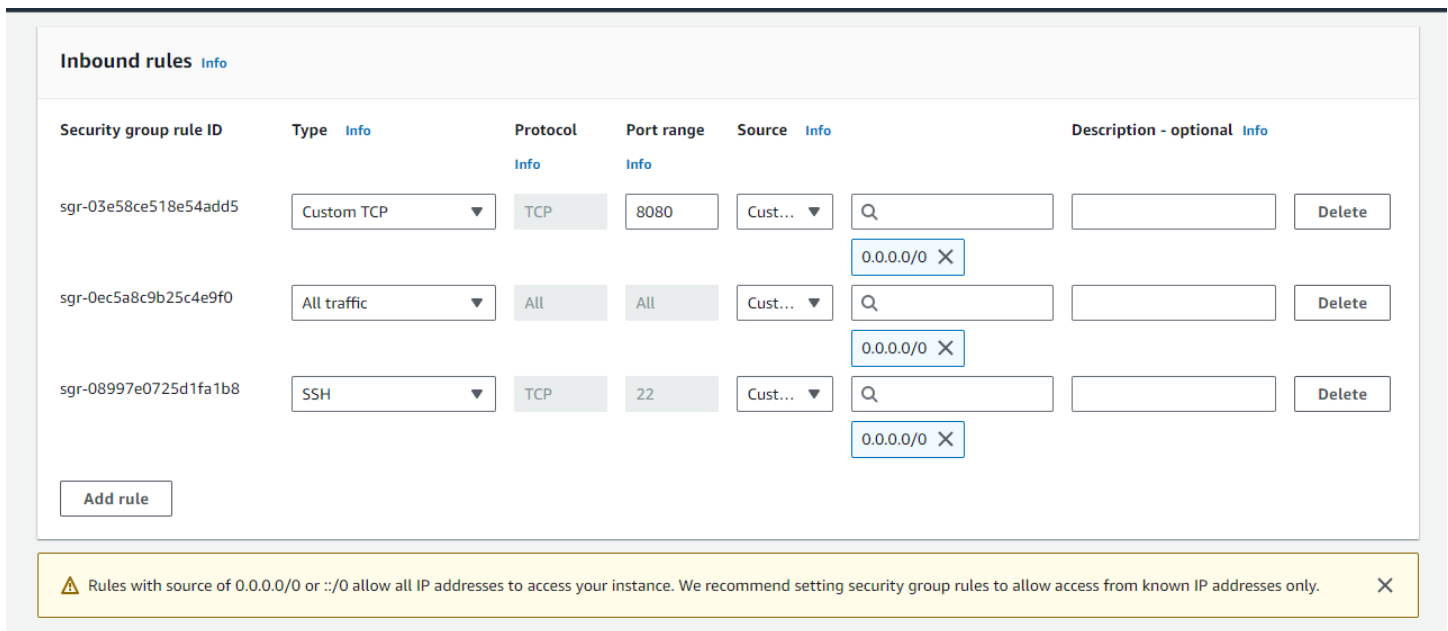
sudo systemctl status jenkins
~
~

```

- Verifying the Jenkins installed

```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
85eb092f36214b09a86b61c2399a956d
root@ip-172-31-26-213:/home/ubuntu# java --version
openjdk 17.0.12 2024-07-16
OpenJDK Runtime Environment (build 17.0.12+7-Ubuntu-1ubuntu224.04)
OpenJDK 64-Bit Server VM (build 17.0.12+7-Ubuntu-1ubuntu224.04, mixed mode, sharing)
root@ip-172-31-26-213:/home/ubuntu# jenkins --version
2.462.3
root@ip-172-31-26-213:/home/ubuntu#
```

- Allow 8080 port in the security groups

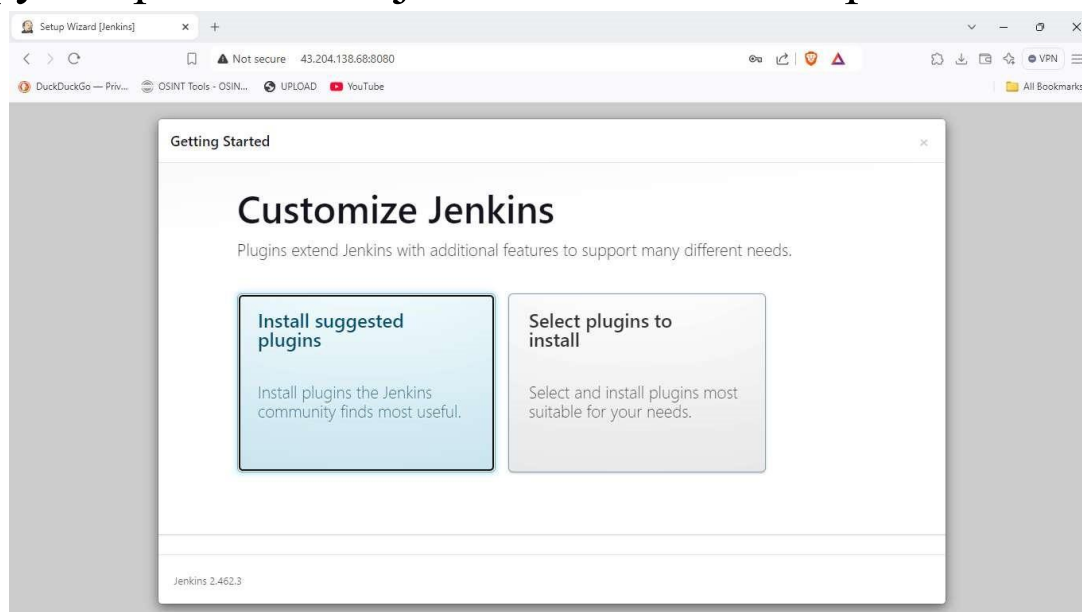


Security group rule ID	Type	Protocol	Port range	Source	Description - optional	Info
sgr-03e58ce518e54add5	Custom TCP	TCP	8080	Cust...	0.0.0.0	Delete
sgr-0ec5a8c9b25c4e9f0	All traffic	All	All	Cust...	0.0.0.0	Delete
sgr-08997e0725d1fa1b8	SSH	TCP	22	Cust...	0.0.0.0	Delete

Add rule

Rules with source of 0.0.0.0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

- Copy the public IP of jenkins machine with port 8080



Setup Wizard [Jenkins] x +

Not secure 43.204.138.68:8080

DuckDuckGo — Priv... OSINT Tools - OSIN... UPLOAD YouTube

Getting Started

Customize Jenkins

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

Install suggested plugins

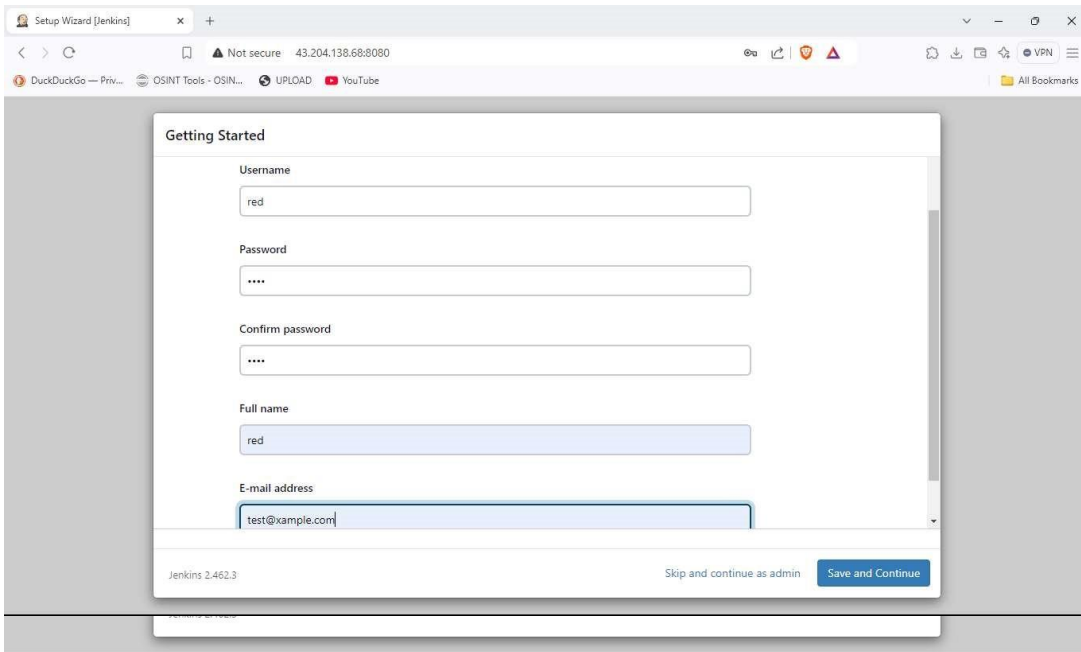
Install plugins the Jenkins community finds most useful.

Select plugins to install

Select and install plugins most suitable for your needs.

Jenkins 2.462.3

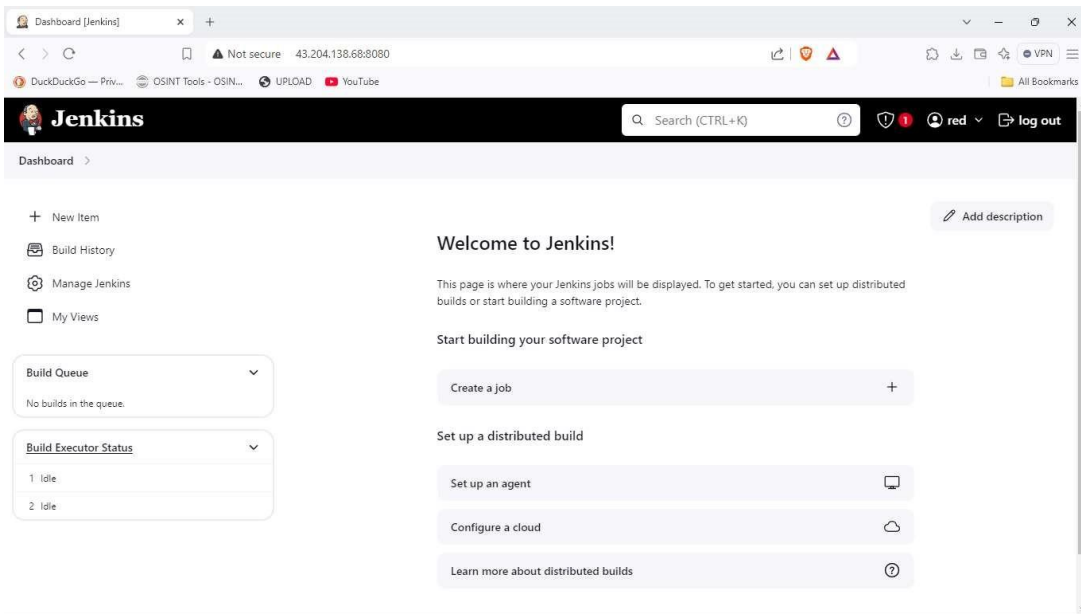
- Give the username and password to use jenkins



The screenshot shows the 'Getting Started' configuration page of the Jenkins Setup Wizard. The form contains the following fields:

- Username:** red
- Password:** ****
- Confirm password:** ****
- Full name:** red
- E-mail address:** test@example.com

At the bottom of the form, there are two buttons: 'Skip and continue as admin' and 'Save and Continue'. The version 'Jenkins 2.462.3' is displayed in the bottom left corner of the wizard window.



Step 4:

- Before starting using jenkins pipeline we need to allow the Jenkins user to execute any command without being prompted for a password

```
root@ip-172-31-26-213:/home/ubuntu# visudo
```

```
GNU nano 7.2 /etc/sudoers.tmp *
# User alias specification
# Cmnd alias specification
# User privilege specification
root    ALL=(ALL:ALL) ALL
jenkins ALL=(ALL:ALL) NOPASSWD: ALL
# Members of the admin group may gain root privileges
%admin   ALL=(ALL) ALL

# Allow members of group sudo to execute any command
%sudo   ALL=(ALL:ALL) ALL

# See sudoers(5) for more information on "@include" directives:

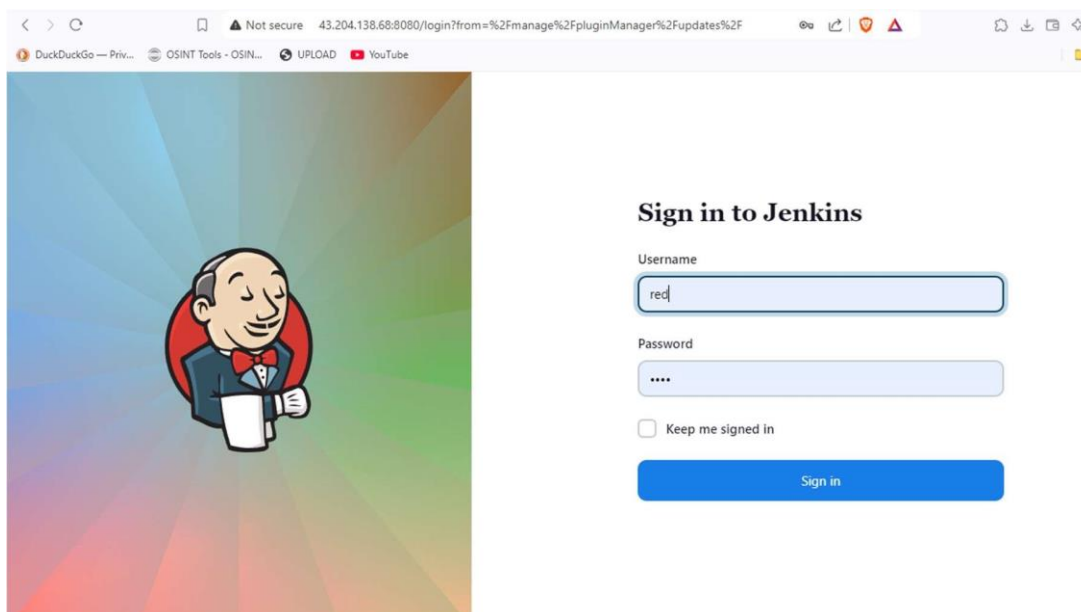
@include:: /etc/sudoers.d

^G Help      ^O Write Out  ^W Where Is   ^R Cut        ^J Execute    ^C Location   ^U Undo       ^M Set Mark
^X Exit      ^B Read File  ^N Replace    ^U Paste      ^_ Justify     ^V Go To Line  ^- Redo       ^- Copy
```

i-007c864f0ad10bf15 (Docker-Jenkins)
PublicIPs: 43.204.138.68 PrivateIPs: 172.31.26.213

- Restart the jenkins and login again

service jenkins restart



Step 5:

We need to install docker in the same machine and then we will give jenkins the permission to access docker

sudo apt install docker.io -y

sudo usermod -aG docker jenkins

service docker restart

```
root@ip-172-31-35-56:~# systemctl restart docker
root@ip-172-31-35-56:~# systemctl status docker
● docker.service - Docker Application Container Engine
   Loaded: loaded (/usr/lib/systemd/system/docker.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-10-28 06:49:40 UTC; 3s ago
     TriggeredBy: ● docker.socket
       Docs: https://docs.docker.com
    Main PID: 6678 (dockerd)
      Tasks: 9
     Memory: 24.0M (peak: 24.3M)
        CPU: 236ms
     CGroup: /system.slice/docker.service
             └─6678 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/containerd.sock
```

i-019aa04bcd40d7c13 (Jenkins)

PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

Step 6:

Install Ansible :

```
root@ip-172-31-35-56:~# ansible --version
ansible [core 2.16.3]
  config file = /root/.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-172-31-35-56:~# █
```

Step 7: Installing dependency plugins required for the project

Preparation

- Checking internet connectivity
- Checking update center connectivity
- Success

Javadoc	✓ Success
JSch dependency	✓ Success
Maven Integration	✓ Success
Pipeline Maven Plugin API	✓ Success
Config File Provider	✓ Success
Pipeline Maven Integration	✓ Success
Maven Invoker	✓ Success
Generic Webhook Trigger	✓ Success
GitHub Integration	✓ Success
Ansible	✓ Success
Cloud Statistics	✓ Success
Authentication Tokens API	✓ Success
Docker Commons	✓ Success
Apache HttpComponents Client 5.x API	✓ Success
Docker API	✓ Success
Docker	✓ Success
Docker Pipeline	✓ Success
Loading plugin extensions	✓ Success
HTML Publisher	✓ Success
Loading plugin extensions	✓ Success

Step 8:

Install Terraform

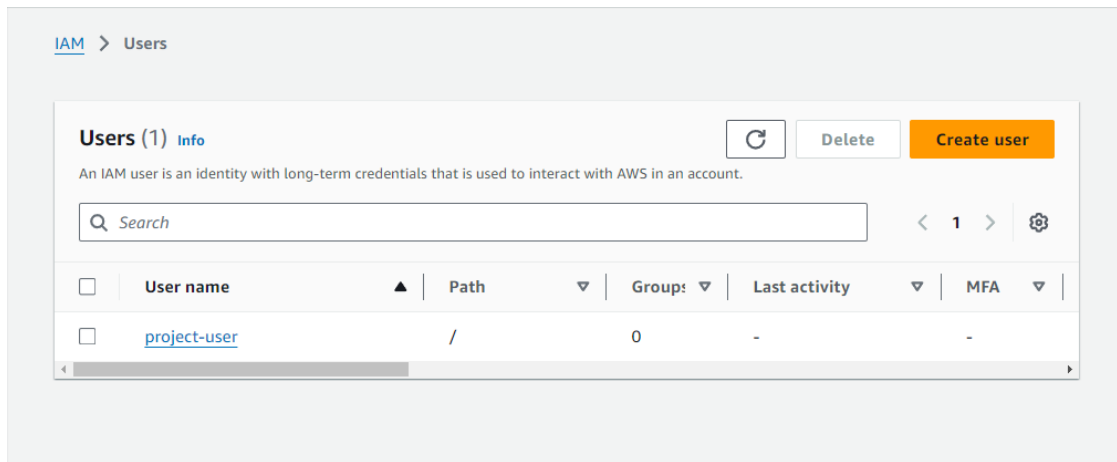
```
no VM guests are running outdated hypervisor (qemu) binaries on this platform
root@ip-172-31-35-56:~# terraform -version
Terraform v1.9.8
on linux_amd64
root@ip-172-31-35-56:~#
```

i-019aa04bcd40d7c13 (Jenkins)

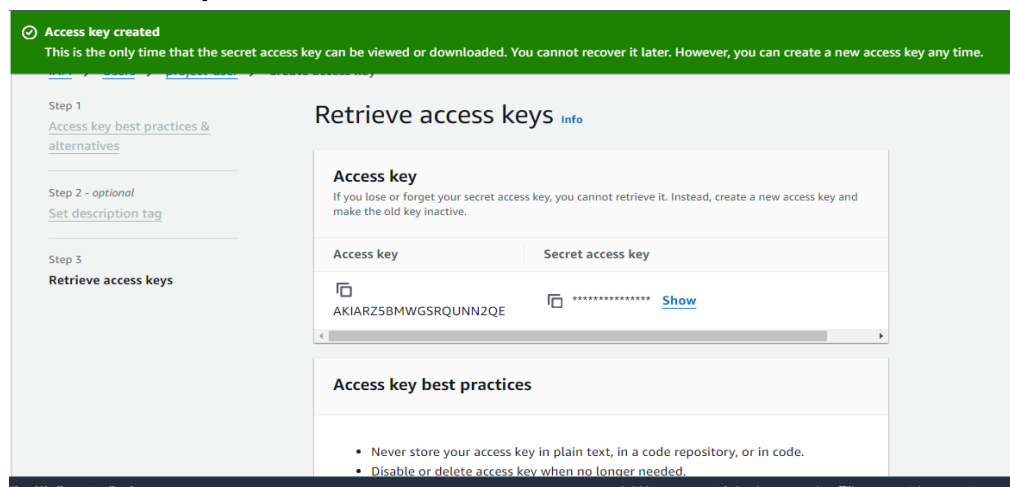
PublicIPs: 3.110.161.113 PrivateIPs: 172.31.35.56

- Once the image is pushed we need to deploy our application .we deploy in the new instances so we use Terraform to create these new instances.

- To use terraform we need to configure AWS CLI which need secret keys
- Create a user and give admin policies



- Create access keys



Step 9:

Go to tools and give the paths for the java,maven,ansible,terraform which we installed In the jenkins names instance.

JDK installations

Add JDK

≡

JDK

✕

Name

JAVA_HOME

JAVA_HOME

/usr/lib/jvm/java-17-openjdk-amd64

☐ Install automatically

?

Add JDK

Maven installations

Add Maven

≡

Maven

✕

Name

\$MAVEN_HOME

MAVEN_HOME

/opt/maven/

☐ Install automatically

?

Add Maven

Ansible installations

Add Ansible

≡

Ansible

✕

Name

ansible

Path to ansible executables directory

/usr/bin/

☐ Install automatically

?

Add Ansible

Docker installations

≡ Terraform

Name

terraform

Install directory

/usr/bin

☐ Install automatically ?

Step 10:

Go to credentials and give necessary credentials which we require to run our pipeline according to Jenkinsfile

- Docker credentials required to build image and push to hub

≡ Username and password (separated) ?

Username Variable ?

dockerlogin

Password Variable ?

dockerpassword

Credentials ?

skywalkerdarth/***** (Docker-login) ▼

+ Add

Add ▼

- To Push application to production using Terraform and Ansible we need terraform credentials and AWS ACCESS KEYS

 Update

 Delete

 Move

ubuntu (terraform-ansible)

terraform-ansible

Usage

This credential has not been recorded as used anywhere.

Note: usage tracking requires the cooperation of plugins and consequently may not track every use.

AKIARZ5BMWGSRQUNN2QE (AWS-ID)

AWS-ID

Usage

This credential has been recorded as used in the following places:

Note: usage tracking requires the cooperation of plugins and consequently may not track every use.



```
stage('Config & Deployment') {  
  steps {  
  
    withCredentials([aws(accessKeyVariable: 'AWS_ACCESS_KEY_ID', credentialsId: 'AWS-ID', secretKeyVariable: 'AWS_SECRET_ACCESS_KEY')]) {  
      dir('terraform-files') {  
        sh 'sudo chmod 600 docker.pem'  
        sh 'terraform init'  
        sh 'terraform validate'  
        sh 'terraform apply --auto-approve'  
      }  
    }  
  }  
}
```

Step 11:

Create the terraform main.tf file to create an . To Push application to production using Terraform .

Edit

Preview

 Code 55% faster with GitHub Copilot

```
1  resource "aws_instance" "test-server" {
2      ami = "ami-0dee22c13ea7a9a67"
3      instance_type = "t2.micro"
4      key_name = "docker"
5      vpc_security_group_ids = ["sg-0fc8b1c023e7b4e38"]
6      connection {
7          type = "ssh"
8          user = "ubuntu"
9          private_key = file("./docker.pem")
10         host = self.public_ip
11     }
12     provisioner "remote-exec" {
13         inline = ["echo 'wait to start the instance' "]
14     }
15     tags = {
16         Name = "test-server"
17     }
18     provisioner "local-exec" {
19         command = "echo ${aws_instance.test-server.public_ip} > inventory"
20     }
21     provisioner "local-exec" {
22         command = "ansible-playbook /var/lib/jenkins/workspace/Banking and Finance Domain Project
23 /terraform-files/ansibleplaybook.yml"
24     }
25 }
```

- Ansible playbook.yml file to install necessary package tools in the server into the production server.

```
- name: Configure Docker on EC2 Instances
hosts: all
become: true
connection: ssh
tasks:
  - name: Updating apt
    command: sudo apt update

  - name: Install required packages
    command: sudo apt install -y apt-transport-https ca-certificates curl software-properties-common

  - name: Update apt again
    command: sudo apt update

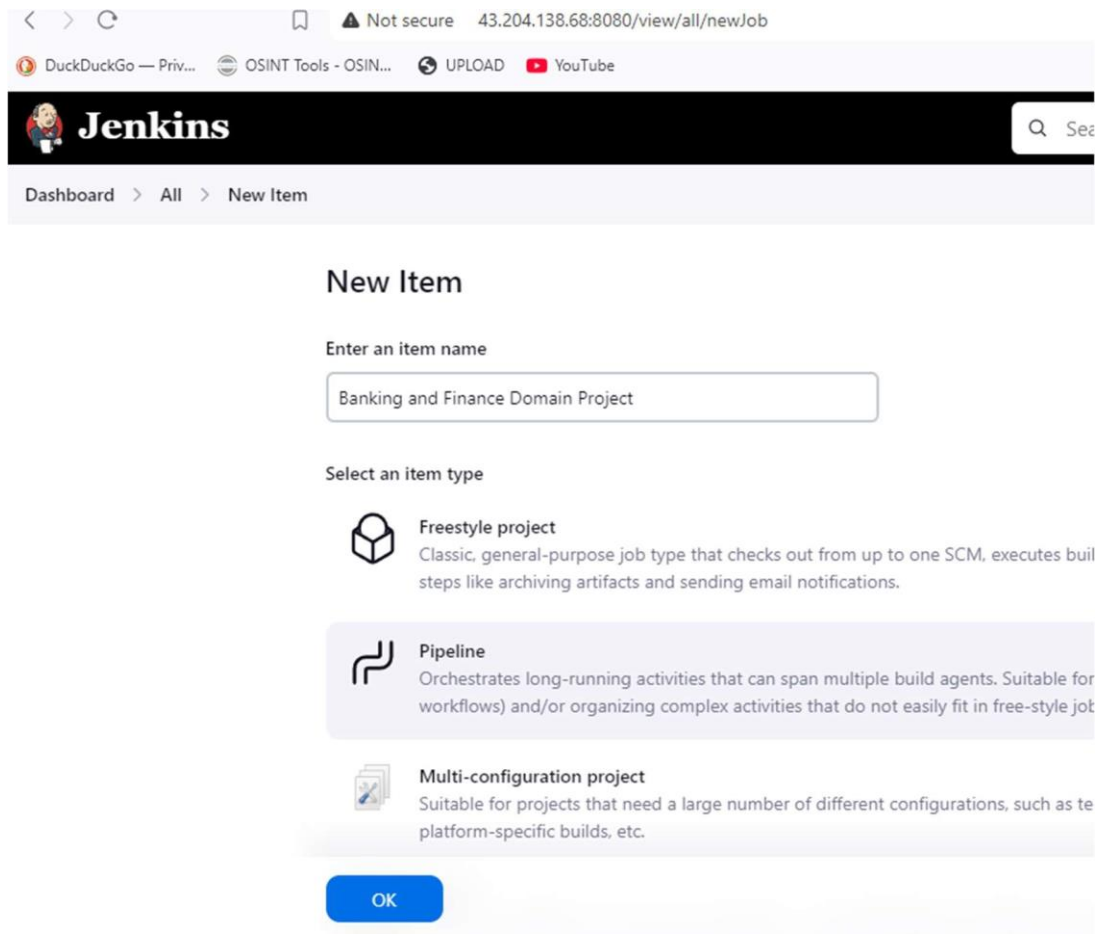
  - name: Install Docker
    command: sudo apt install docker.io -y

  - name: Start Docker Service
    command: sudo systemctl start docker

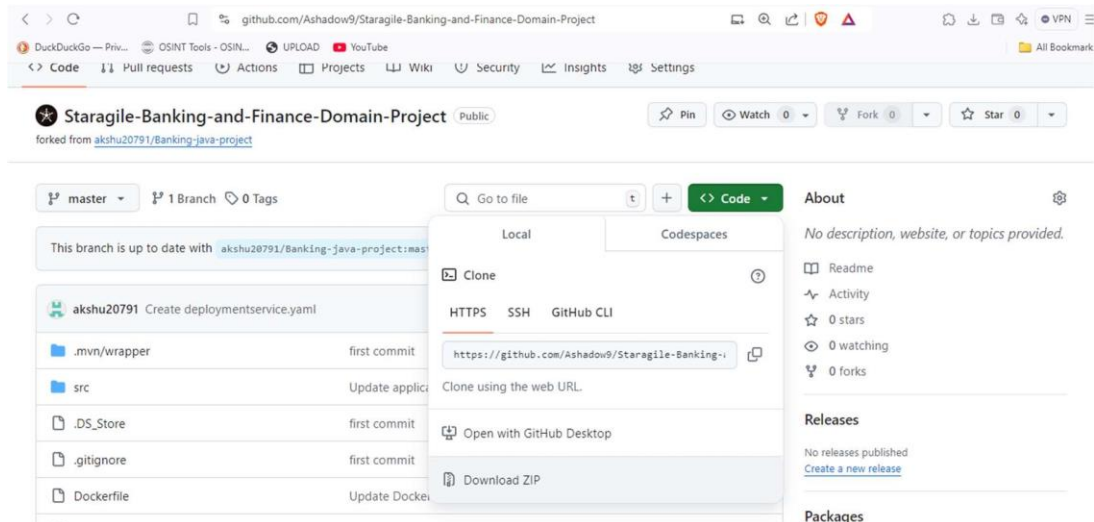
  - name: Deploy Docker Container
    command: docker run -itd -p 8084:8081 skywalkerdarth/banking-project-demo:3.0
```

Step 12:

- Now, Select a new item from the Jenkins dashboard to create the job and select the pipeline



- Copy the GitHub repo in which we have the Jenkin file



- paste it in scm in the pipeline configure option

Dashboard > Banking and Finance Domain Project > Configuration

Configure

General
Advanced Project Options
Pipeline

Pipeline

Definition
Pipeline script from SCM

SCM ?
Git

Repositories ?

Repository URL ?
`https://github.com/Ashadow9/Staragile-Banking-and-Finance-Domain-Project.git`

Credentials ?
- none -

+ Add

Save Apply

- Now, start the build and we can see the process:

Dashboard > Banking and Finance Domain Project > Changes

Build Now
Configure
Delete Pipeline
Full Stage View
HTML Report
Stages
Rename
Pipeline Syntax

Stage View

Average stage times:
(Average full run time: ~3min 47s)

	Declarative: Checkout SCM	Declarative: Tool Install	Build	Generate Test Reports	Create Docker Image	Docker-Login	Push-Image	Config & Deployment
Average	876ms	122ms	13s	223ms	5s	2s	11s	59s
#10 Oct 28 15:22 1 commit	874ms	125ms	12s	229ms	7s	2s	13s	3min 8s
#9 Oct 28 15:17 1 commit	1s	118ms	12s	203ms	7s	2s	14s	2min 30s failed
#8 Oct 28 15:12 1 commit	843ms	107ms	12s	197ms	7s	2s	14s	2min 6s failed
#7 Push 76								

Build History trend
Filter... /

- The build is success :

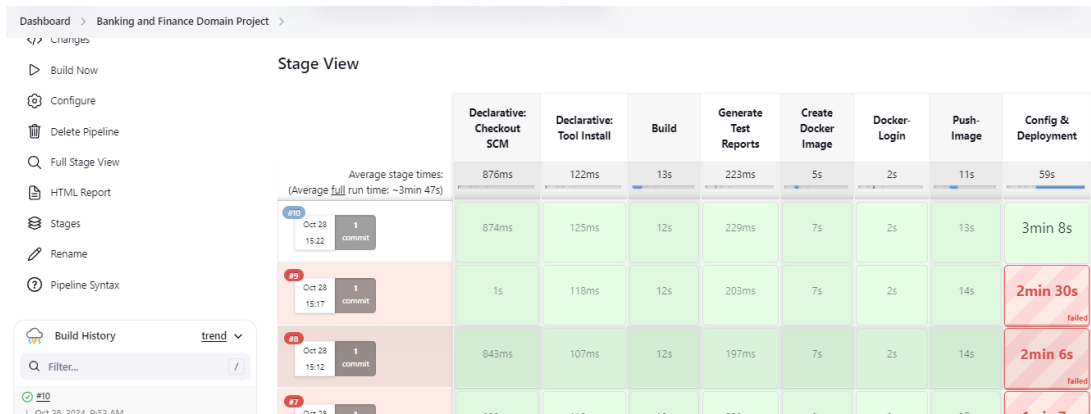
Dashboard > Banking and Finance Domain Project > #10

Status
Changes
Console Output
Edit Build Information
Delete build '#10'
Timings
Git Build Data
Pipeline Overview
Pipeline Console
Restart from Stage
Replay
Pipeline Steps
Workspaces
Previous Build

Console Output

Download Copy View as plain text

```
Started by user red
Obtained Jenkinsfile from git https://github.com/Ashadow9/star-agile-banking-finance.git
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Banking and Finance Domain Project
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Checkout SCM)
[Pipeline] checkout
Selected Git installation does not exist. Using Default
The recommended git tool is: NONE
No credentials specified
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/Banking and Finance Domain Project/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/Ashadow9/star-agile-banking-finance.git # timeout=10
Fetching upstream changes from https://github.com/Ashadow9/star-agile-banking-finance.git
> git --version # timeout=10
> git --version # 'git version 2.43.0'
> git fetch --tags --force --progress -- https://github.com/Ashadow9/star-agile-banking-finance.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/master^(commit) # timeout=10
Checking out Revision 2766a560b0562f77305bc7352be2418454005ef (refs/remotes/origin/master)
> git config core.sparsecheckout # timeout=10
```



Step 13:

- The instance is created

Instances (2) Info						
Find Instance by attribute or tag (case-sensitive)		Running				
<input type="checkbox"/>	Name	Instance ID	Instance state	Instance type	Status check	Alarm status
<input type="checkbox"/>	Jenkins	i-019aa04bcd40d7c13	Running	t2.medium	2/2 checks passed	View alarms
<input type="checkbox"/>	test-server	i-01735c5c26bb77005	Running	t2.micro	2/2 checks passed	View alarms

- We can see the docker images and docker containers build from the pipeline

```

stage('Create Docker Image') {
  steps {
    sh 'docker build -t skywalkerdarth/banking-project-demo:3.0 .'
  }
}
stage('Docker-Login') {
  steps {
    withCredentials([usernamePassword(credentialsId: 'Docker-login', passwordVariable: 'dockerpassword', usernameVariable: 'dockerlogin')]) {
      sh 'docker login -u ${dockerlogin} -p ${dockerpassword}'
    }
  }
}
stage('Push-Image') {
  steps {
    sh 'docker push skywalkerdarth/banking-project-demo:3.0'
  }
}

```

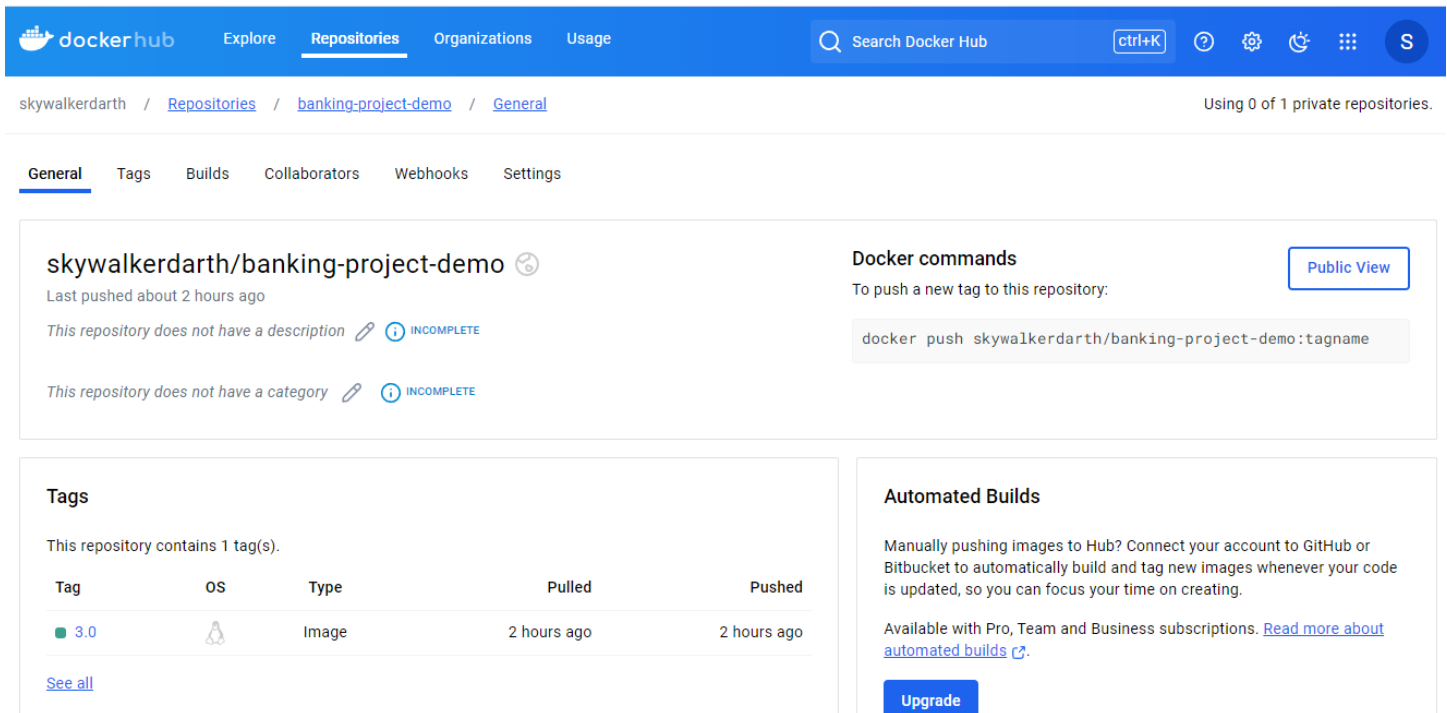


```

root@ip-172-31-26-213:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS                               NAMES
a95601fba189   myimg    "java -jar /app.jar"    8 minutes ago Up 8 minutes   0.0.0.0:8091->8091/tcp, :::8091->8091/tcp   c000
root@ip-172-31-26-213:/home/ubuntu# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
myimg         latest   8cb3f8b3a13c   8 minutes ago   696MB
openjdk       11       47a932d998b7   2 years ago     654MB
root@ip-172-31-26-213:/home/ubuntu#

```

- The image is pushed into the docker hub



The screenshot shows the Docker Hub interface for the repository 'skywalkerdarth/banking-project-demo'. The page includes a navigation bar with 'dockerhub', 'Explore', 'Repositories' (selected), 'Organizations', and 'Usage'. A search bar and utility icons are on the right. The breadcrumb trail is 'skywalkerdarth / Repositories / banking-project-demo / General'. The 'General' tab is active, showing repository details: 'Last pushed about 2 hours ago', 'This repository does not have a description' (incomplete), and 'This repository does not have a category' (incomplete). A 'Public View' button is present. The 'Docker commands' section shows the command 'docker push skywalkerdarth/banking-project-demo:tagname'. The 'Tags' section lists one tag: '3.0' (Image, pulled 2 hours ago, pushed 2 hours ago). The 'Automated Builds' section explains how to connect GitHub or Bitbucket for automatic builds, with an 'Upgrade' button for Pro, Team, or Business subscriptions.

Step 14:

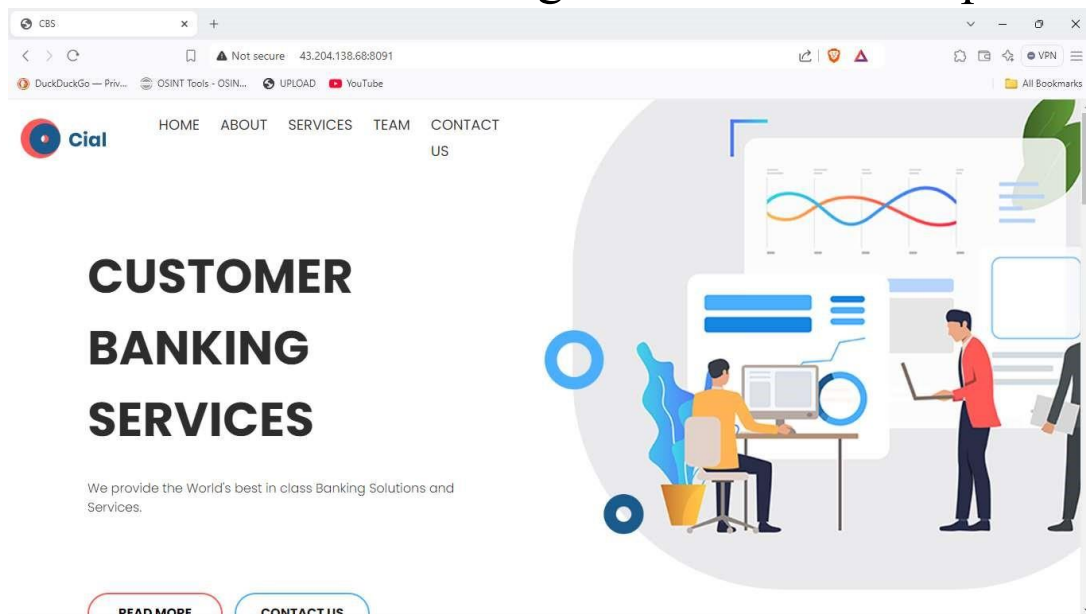
Allow the port 8084 in the security group of the Instance

Inbound rules Info						
Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info	
sgr-07b01c55d01318d74	Custom TCP ▼	TCP	8084	Cust... ▼	Q	Delete
					0.0.0.0/0 ✕	
sgr-03e58ce518e54add5	Custom TCP ▼	TCP	8080	Cust... ▼	Q	Delete
					0.0.0.0/0 ✕	
sgr-0ec5a8c9b25c4e9f0	All traffic ▼	All	All	Cust... ▼	Q	Delete
					0.0.0.0/0 ✕	
sgr-08997e0725d1fa1b8	SSH ▼	TCP	22	Cust... ▼	Q	Delete
					0.0.0.0/0 ✕	
Add rule						

Step 15:

Now run the public IP address of the server with the port mentioned in the Jenkins file .

We can see the website is working on the mentioned port




```
sudo mv prometheus-2.43.0.linux-amd64 /usr/local/Prometheus
```

```
cd /usr/local/Prometheus
```

- vi prometheus.yml

```
- job_name: 'node_exporter'
```

```
static_configs:
```

```
- targets: [publicip:9100']
```

```
- static_configs:
  - targets:
    # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
  - job_name: "prometheus"

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

    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'node_exporter'

    static_configs:
      - targets: ['35.154.59.115:9100']
"prometheus.yml" [readonly] 33L, 1029B
```

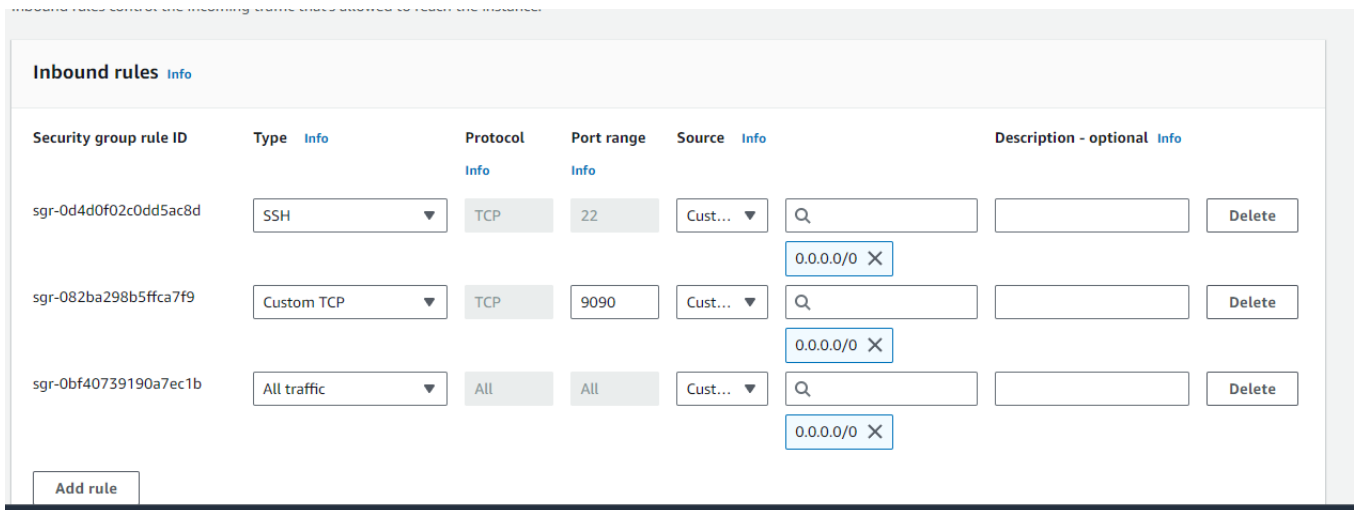
- Start Prometheus by running:

```
./prometheus --config.file=prometheus.yml
```

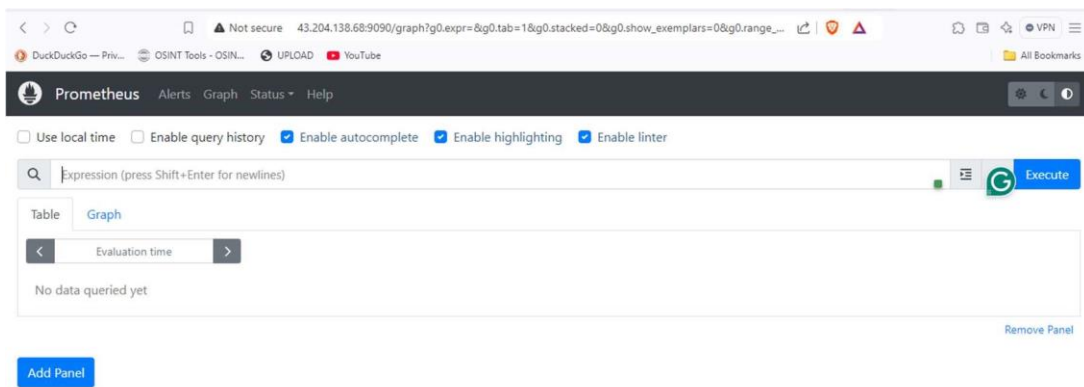
```
root@ip-172-31-26-213:/usr/local/prometheus# vi prometheus.yml
root@ip-172-31-26-213:/usr/local/prometheus# ./prometheus --config.file=prometheus.yml
ts=2024-10-05T06:41:03.428Z caller=main.go:520 level=info msg="No time or size retention was set so using the default of 30 days"
ts=2024-10-05T06:41:03.428Z caller=main.go:564 level=info msg="Starting Prometheus Server" mode=server version=2.43.0 build_info="(go=go1.19.7, platform=linux/amd64, ts=2024-10-05T06:41:03.428Z) caller=main.go:569 level=info build_context="(go=go1.19.7, platform=linux/amd64, ts=2024-10-05T06:41:03.428Z) caller=main.go:570 level=info host_details="(Linux 6.8.0-1016-aws #17-Ubuntu SMP Thu Aug 1 12:56:07, tags=netgo,builtinsassets) ip-172-31-26-213 (none)"
ts=2024-10-05T06:41:03.429Z caller=main.go:571 level=info fd_limits="(soft=1048576, hard=1048576)"
ts=2024-10-05T06:41:03.429Z caller=main.go:572 level=info vm_limits="(soft=unlimited, hard=unlimited)"
ts=2024-10-05T06:41:03.435Z caller=web.go:561 level=info component=web msg="Start listening for connections"
ts=2024-10-05T06:41:03.436Z caller=main.go:1005 level=info msg="Starting TSDB ..."
ts=2024-10-05T06:41:03.444Z caller=tls_config.go:232 level=info component=web msg="Listening on" address=[::]:9090
ts=2024-10-05T06:41:03.444Z caller=tls_config.go:235 level=info component=web msg="TLS is disabled." http2=false
ts=2024-10-05T06:41:03.451Z caller=head.go:587 level=info component=tsdb msg="Replaying on-disk memory mappable chunks if any"
ts=2024-10-05T06:41:03.451Z caller=head.go:658 level=info component=tsdb msg="On-disk memory mappable chunks replayed"
ts=2024-10-05T06:41:03.451Z caller=head.go:664 level=info component=tsdb msg="Replaying WAL, this may take a while"
ts=2024-10-05T06:41:03.466Z caller=head.go:735 level=info component=tsdb msg="WAL segment loaded" segment=0
ts=2024-10-05T06:41:03.467Z caller=head.go:735 level=info component=tsdb msg="WAL segment loaded" segment=1
ts=2024-10-05T06:41:03.467Z caller=head.go:772 level=info component=tsdb msg="WAL replay completed" checkpoint=15.42779ms wbl_replay_duration=154ns total_replay_duration=15.601857ms
ts=2024-10-05T06:41:03.468Z caller=main.go:1026 level=info fs_type=EXT4_SUPER_MAGIC
```

Prometheus will now be accessible via public ip :9090.

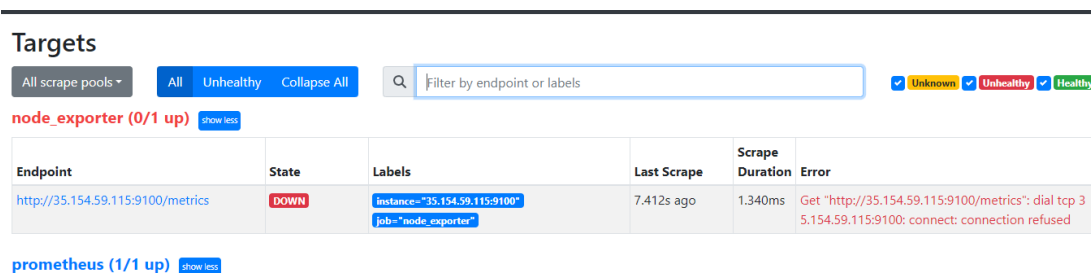
- Allow port 9090 in the security group of the Prometheus server



- Paste the public ip with port 9090 to access prometheus



- Node Exporter is down so we need to install node exporter



- Install Node Exporter (For Server Metrics)

wget ps://github.com/prometheus/node_exporter/releases/download/v1.6.0/node_exporter-1.6.0.linux-amd64.tar.gz

```
root@ip-172-31-26-213:/usr/local/prometheus# wget https://github.com/prometheus/node_exporter/releases/download/v1.6.0/node_exporter-1.6.0.linux-amd64.tar.gz
--2024-10-05 06:42:13-- https://github.com/prometheus/node_exporter/releases/download/v1.6.0/node_exporter-1.6.0.linux-amd64.tar.gz
Resolving github.com (github.com)... 20.207.73.82
Connecting to github.com (github.com)|20.207.73.82|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/0596919f-7be9-4f77-88af-1bfc173d40c77X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241005%2Fus-east-1%2F%3Faws4_request%26X-Amz-Expires=300&X-Amz-Signature=526e86be4673dfbf604d08d61e549ecc16178ed2f4f94b70c04cb98c726bb116X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream [following]
--2024-10-05 06:42:13-- https://objects.githubusercontent.com/github-production-release-asset-2e65be/9524057/0596919f-7be9-4f77-88af-1bfc173d40c77X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241005%2Fus-east-1%2F%3Faws4_request%26X-Amz-Expires=300&X-Amz-Signature=526e86be4673dfbf604d08d61e549ecc16178ed2f4f94b70c04cb98c726bb116X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dnode_exporter-1.6.0.linux-amd64.tar.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.111.133, 185.199.110.133, 185.199.108.133, ...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.111.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10367009 (9.9M) [application/octet-stream]
Saving to: 'node_exporter-1.6.0.linux-amd64.tar.gz'

node_exporter-1.6.0.linux-amd64. 100%[=====] 9.89M --.-KB/s in 0.04s
```

tar -xvf node_exporter-1.6.0.linux-amd64.tar.gz sudo mv node_exporter-1.6.0.linux-amd64
/usr/local/node_exporter
cd /usr/local/node_exporter

```
root@ip-172-31-26-213:/usr/local/prometheus# tar -xvf node_exporter-1.6.0.linux-amd64.tar.gz
node_exporter-1.6.0.linux-amd64/
node_exporter-1.6.0.linux-amd64/NOTICE
node_exporter-1.6.0.linux-amd64/node_exporter
node_exporter-1.6.0.linux-amd64/LICENSE
root@ip-172-31-26-213:/usr/local/prometheus# sudo mv node_exporter-1.6.0.linux-amd64 /usr/local/node_exporter
root@ip-172-31-26-213:/usr/local/prometheus# cd /usr/local/node_exporter
root@ip-172-31-26-213:/usr/local/node_exporter#
```

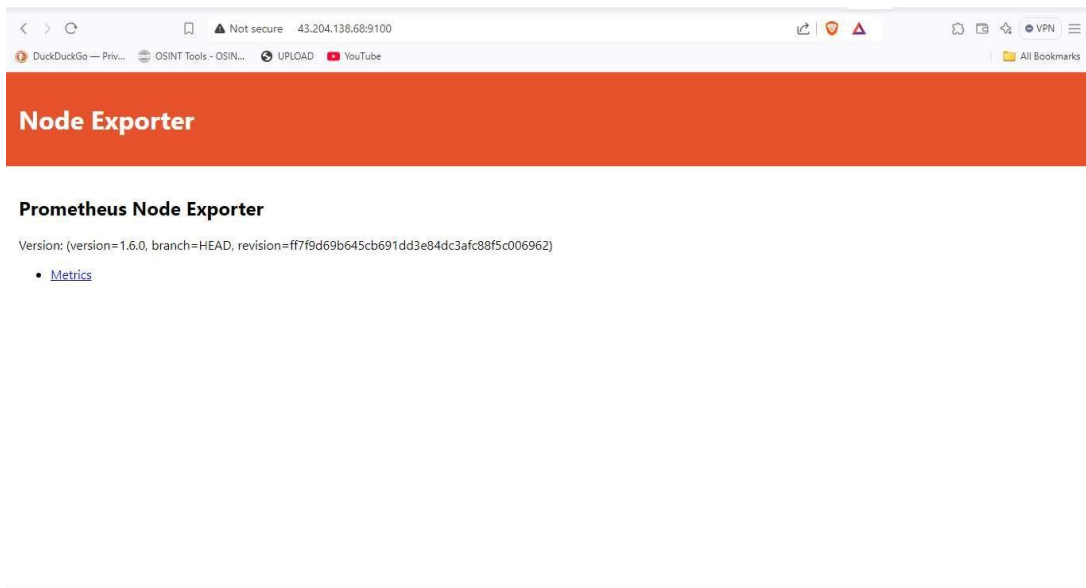
Start Node Exporter

./node_exporter


```
root@ip-172-31-26-213:/usr/local/node_exporter# ./node_exporter
ts=2024-10-05T06:44:13.437Z caller=node_exporter.go:180 level=info msg="Starting node_exporter" version="(ve
7f9d69b645cb691dd3e84dc3afc88f5c006962)"
ts=2024-10-05T06:44:13.437Z caller=node_exporter.go:181 level=info msg="Build context" build_context="(go=go
ot8f9c3ed0cfbd3, date=20230527-12:03:54, tags=netgo osusergo static build)"
ts=2024-10-05T06:44:13.437Z caller=node_exporter.go:183 level=warn msg="Node Exporter is running as root use
s unprivileged user, root is not required."
ts=2024-10-05T06:44:13.438Z caller=diskstats_common.go:111 level=info collector=diskstats msg="Parsed flag -
flag=^(ram|loop|fd|(h|s|v|xv)d[a-z]|nvme\d+n\d+p)\d+$
ts=2024-10-05T06:44:13.439Z caller=filesystem_common.go:111 level=info collector=filesystem msg="Parsed flag
exclude" flag=^(dev|proc|run|credentials/.+|sys|var/lib/docker/.+|var/lib/containers/storage/.+)(/|$)
ts=2024-10-05T06:44:13.439Z caller=filesystem_common.go:113 level=info collector=filesystem msg="Parsed flag
ude" flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue
_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
ts=2024-10-05T06:44:13.440Z caller=node_exporter.go:110 level=info msg="Enabled collectors"
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=arp
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=bcache
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=bonding
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=btrfs
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=conntrack
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=cpu
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=cputime
ts=2024-10-05T06:44:13.441Z caller=node_exporter.go:117 level=info collector=diskstats
```

i-007c864f0ad10bf15 (Docker-Jenkins)
PublicIPs: 43.204.138.68 PrivateIPs: 172.31.26.213

http://43.204.138.68:9100/



- Node exporter is up


Prometheus


Alerts


Graph

Status ▾

Help








Targets

All scrape pools ▾


AllUnhealthyCollapse All

Q

Filter by endpoint or labels

Unknown

☐Unhealthy

Healthy

node_exporter (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://35.154.59.115:9100/metrics	UP	instance="35.154.59.115:9100" job="node_exporter"	1.589s ago	9.862ms	

prometheus (1/1 up)

show less

Endpoint	State	Labels	Last Scrape	Scrape Duration	Error
http://localhost:9090/metrics	UP	instance="localhost:9090" job="prometheus"	13.309s ago	4.085ms	

Step 17: Install Grafana :-

sudo apt-get update

sudo apt-get install -y adduser libfontconfig1 musl

wget https://dl.grafana.com/enterprise/release/grafanaenterprise_11.2.2_amd64.deb

sudo dpkg -i grafana-enterprise_11.2.2_amd64.deb

Log in using the default credentials:

- Username: admin
- Password: admin

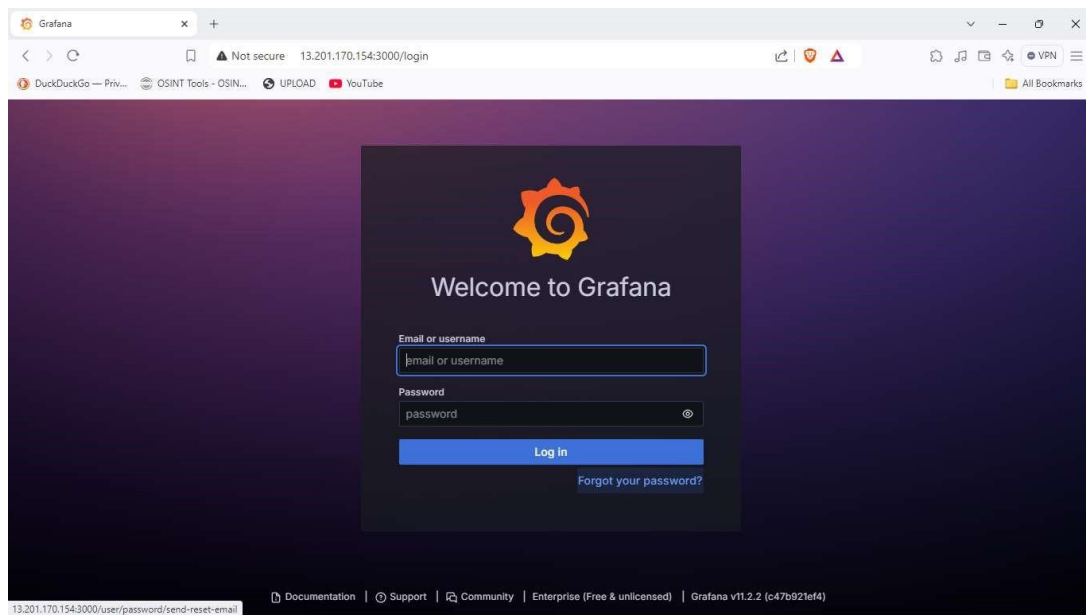

```

ubuntu@ip-172-31-8-38:~$ sudo systemctl status grafana-server
● grafana-server.service - Grafana instance
   Loaded: loaded (/usr/lib/systemd/system/grafana-server.service; enabled; preset: enabled)
   Active: active (running) since Mon 2024-10-28 11:04:30 UTC; 18s ago
     Docs: http://docs.grafana.org
    Main PID: 4736 (grafana)
      Tasks: 14 (limit: 4676)
     Memory: 76.3M (peak: 76.7M)
        CPU: 2.644s
    CGroup: /system.slice/grafana-server.service
            └─4736 /usr/share/grafana/bin/grafana server --config=/etc/grafana/grafana.ini --pid

Oct 28 11:04:35 ip-172-31-8-38 grafana[4736]: logger=plugins.update.checker t=2024-10-28T11:04:35
Oct 28 11:04:36 ip-172-31-8-38 grafana[4736]: logger=grafana.update.checker t=2024-10-28T11:04:36
Oct 28 11:04:36 ip-172-31-8-38 grafana[4736]: logger=plugin.angular detectorsprovider.dynamic t=2024-10-28T11:04:36
Oct 28 11:04:36 ip-172-31-8-38 grafana[4736]: logger=grafana-apiserver t=2024-10-28T11:04:36.0531
Oct 28 11:04:36 ip-172-31-8-38 grafana[4736]: logger=grafana-apiserver t=2024-10-28T11:04:36.0543
Oct 28 11:04:36 ip-172-31-8-38 grafana[4736]: logger=grafana-apiserver t=2024-10-28T11:04:36.0557
Oct 28 11:04:37 ip-172-31-8-38 grafana[4736]: logger=plugin.installer t=2024-10-28T11:04:37.24782
Oct 28 11:04:37 ip-172-31-8-38 grafana[4736]: logger=installer.fs t=2024-10-28T11:04:37.358806364
Oct 28 11:04:37 ip-172-31-8-38 grafana[4736]: logger=plugins.registration t=2024-10-28T11:04:37.4
Oct 28 11:04:37 ip-172-31-8-38 grafana[4736]: logger=plugin.backgroundinstaller t=2024-10-28T11:0
lines 1-21/21 (END)

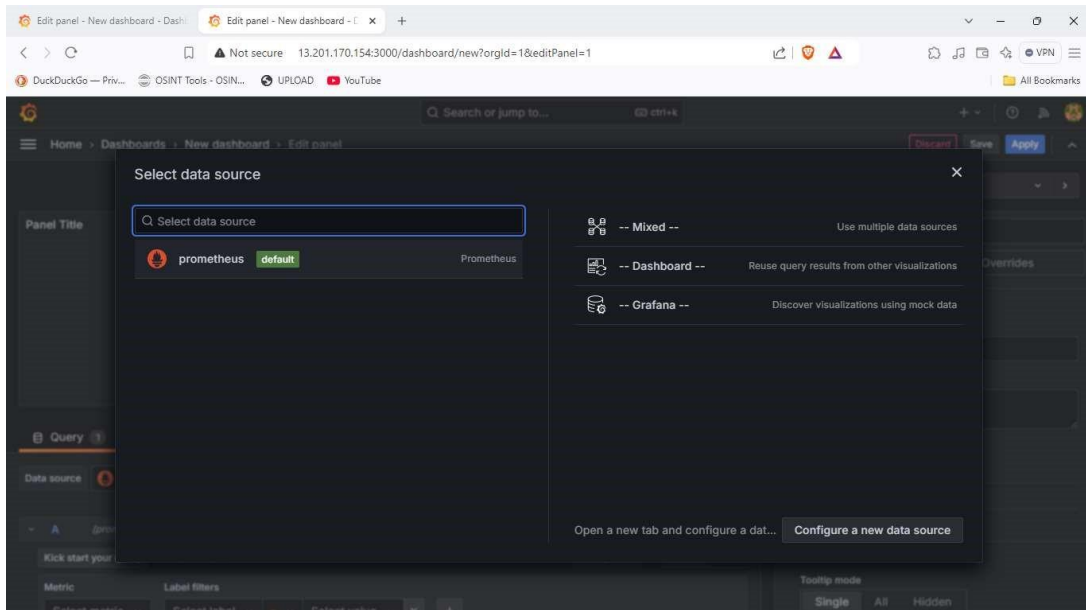
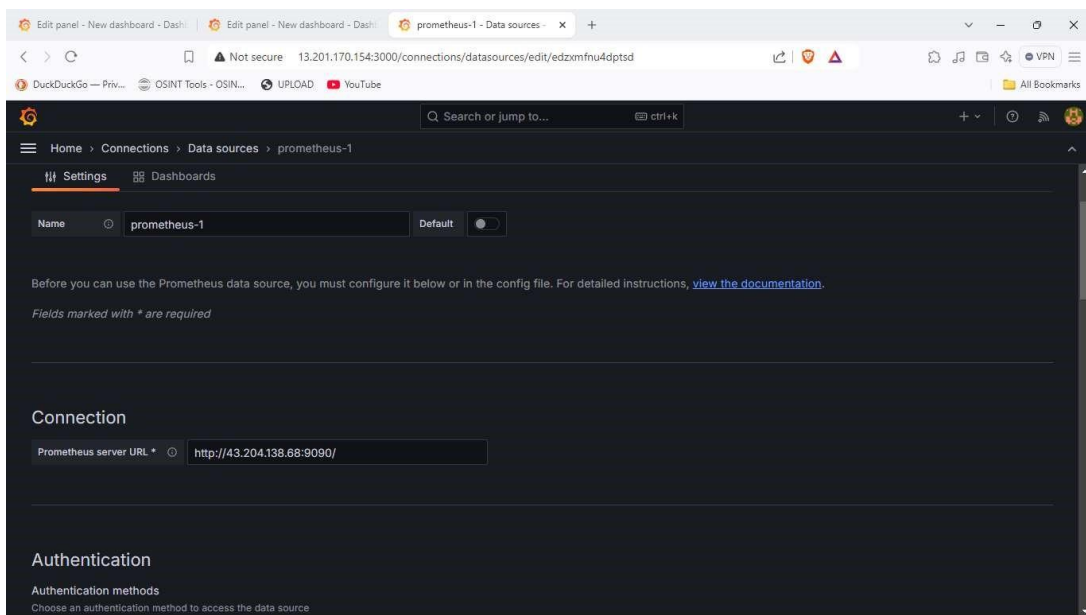
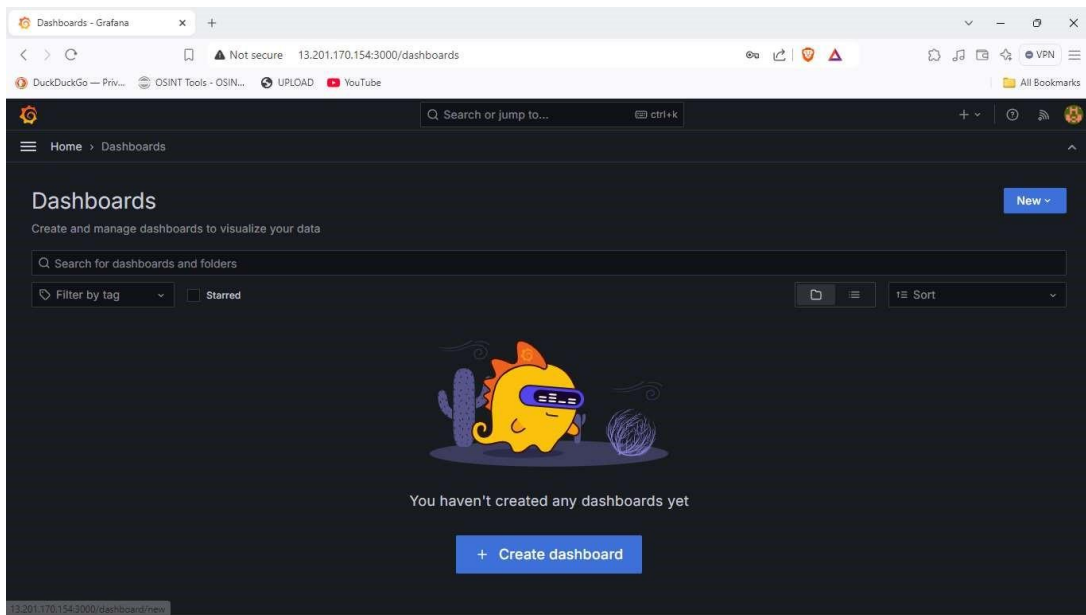
```

- Grafana will be accessible via publicip: 3000.



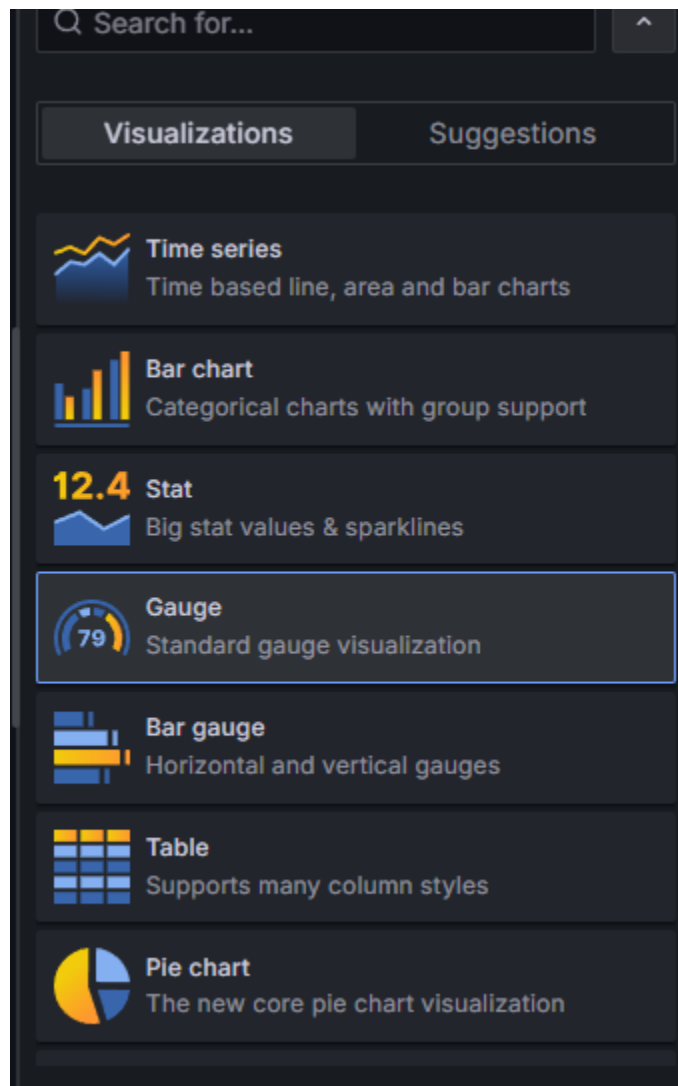
Dashboard of Grafana:

- navigate to **Configuration > Data Sources**.
- Click on **Add Data Source**, select **Prometheus**.
- Enter the URL of Prometheus: `http://<your-server-ip>:9090`.
- Click **Save & Test** to verify the connection.

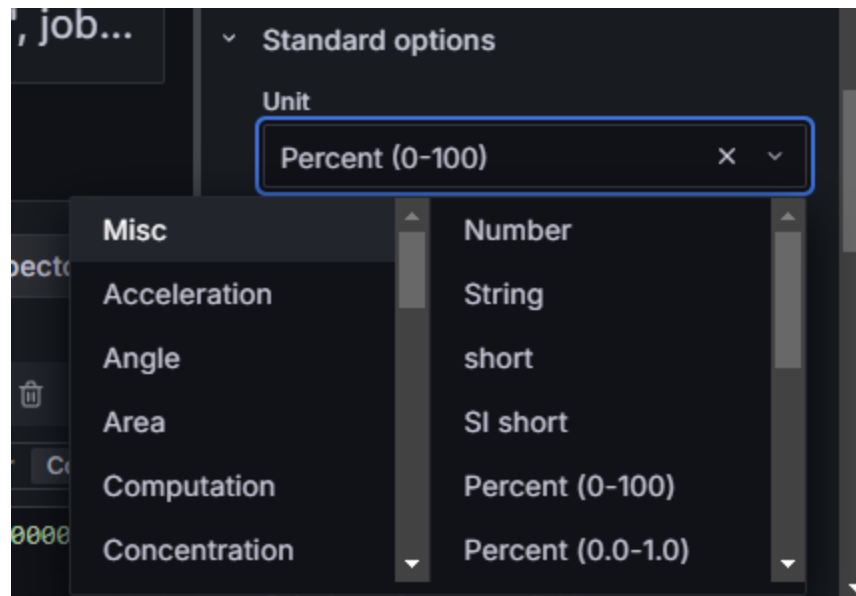


Step 18 : Metric Visualization:

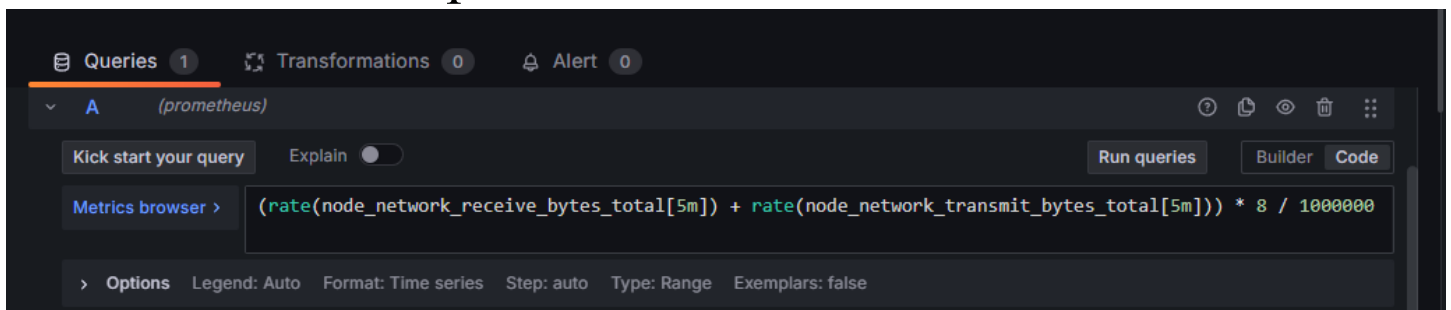
- Select gauge visualization to see the metric



- Select the Misc and percent (0-100) from standard options in unit



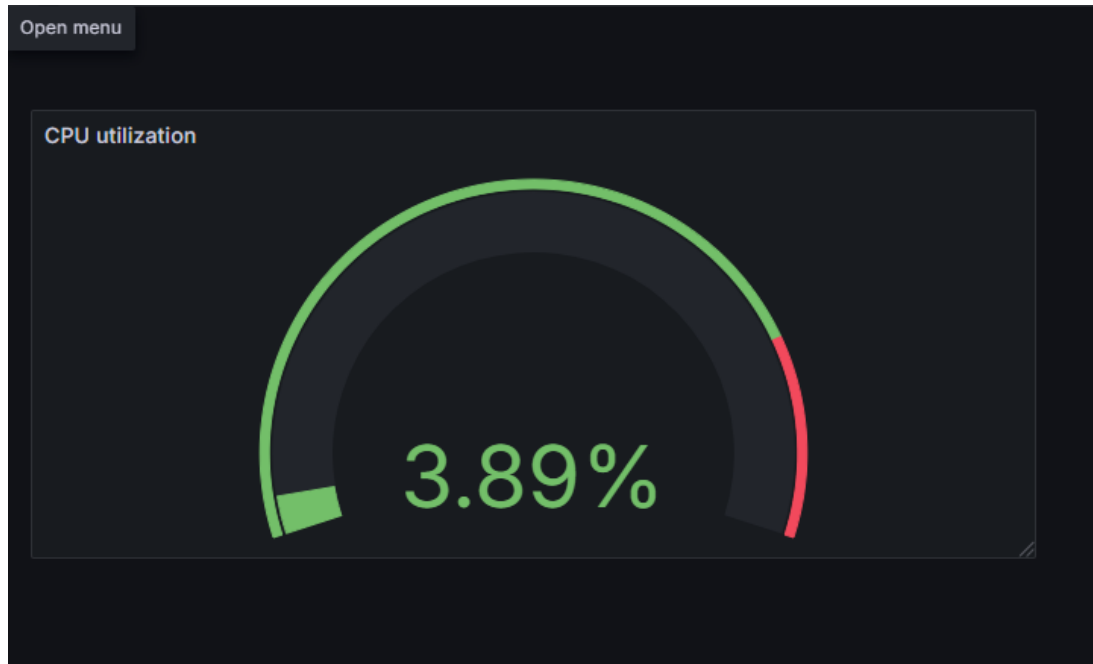
- Select the metric browser option and run queries and we can see the output visualization



Step 19:

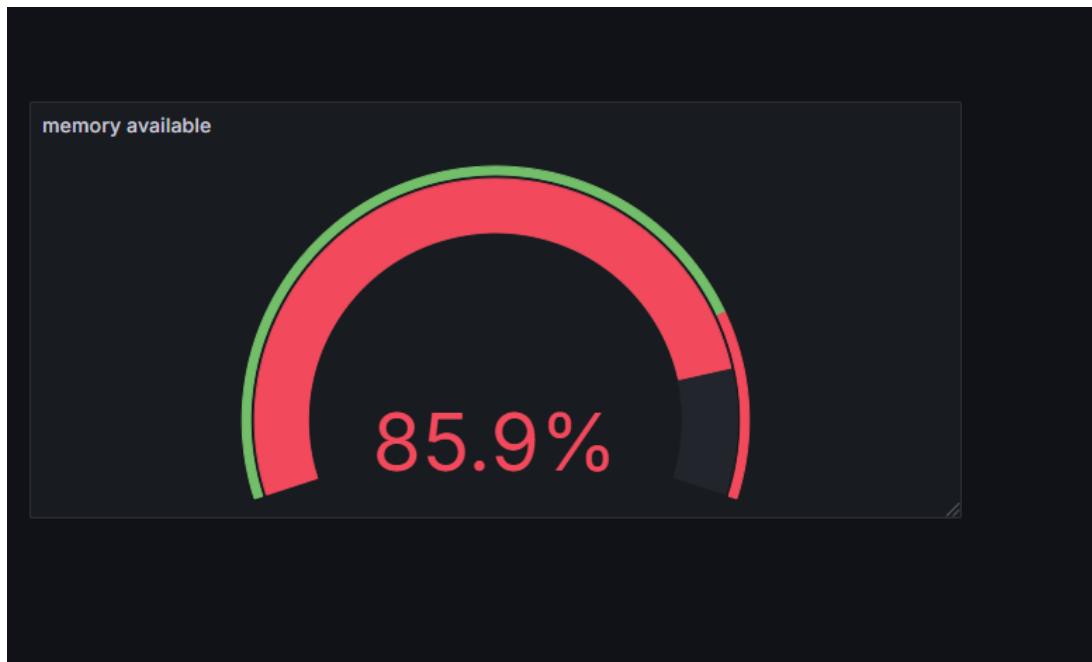
1. **CPU utilization:** To see CPU utilization of the server use gauge and select misc and run queries and paste this

$100 * (1 - \text{avg by(instance)} (\text{rate}(\text{node_cpu_seconds_total}\{\text{mode}=\text{"idle"}\}[5\text{m}])))$



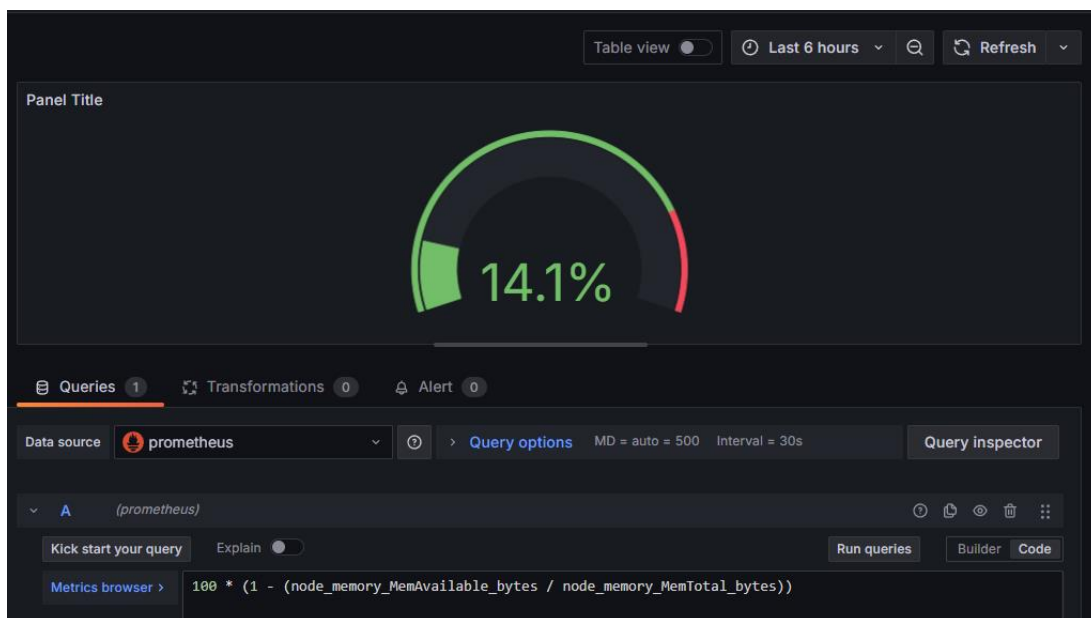
2. **Total Available Memory:-** To see available memory of the server

$100 * (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes})$



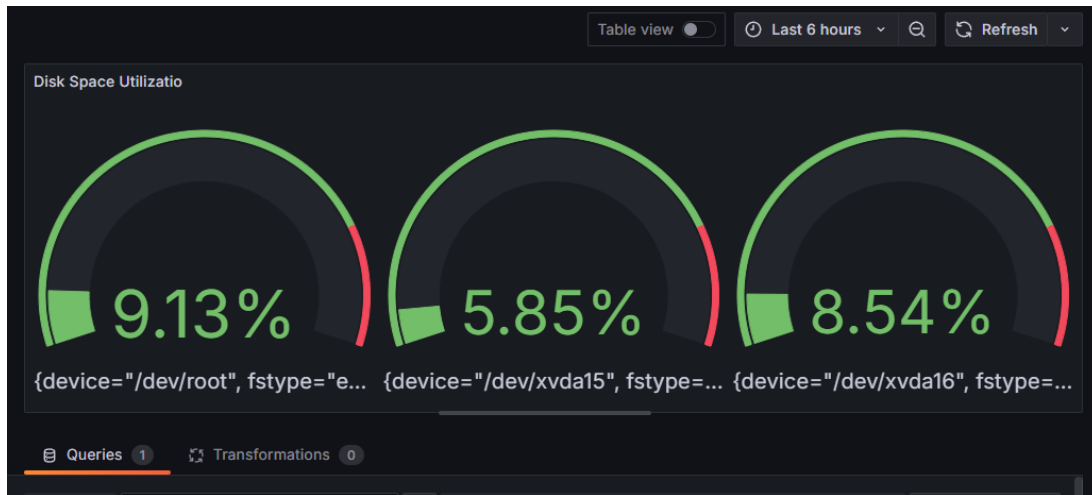
- To see the total memory

$100 * (1 - (\text{node_memory_MemAvailable_bytes} / \text{node_memory_MemTotal_bytes}))$



3. Disk Space Utilization

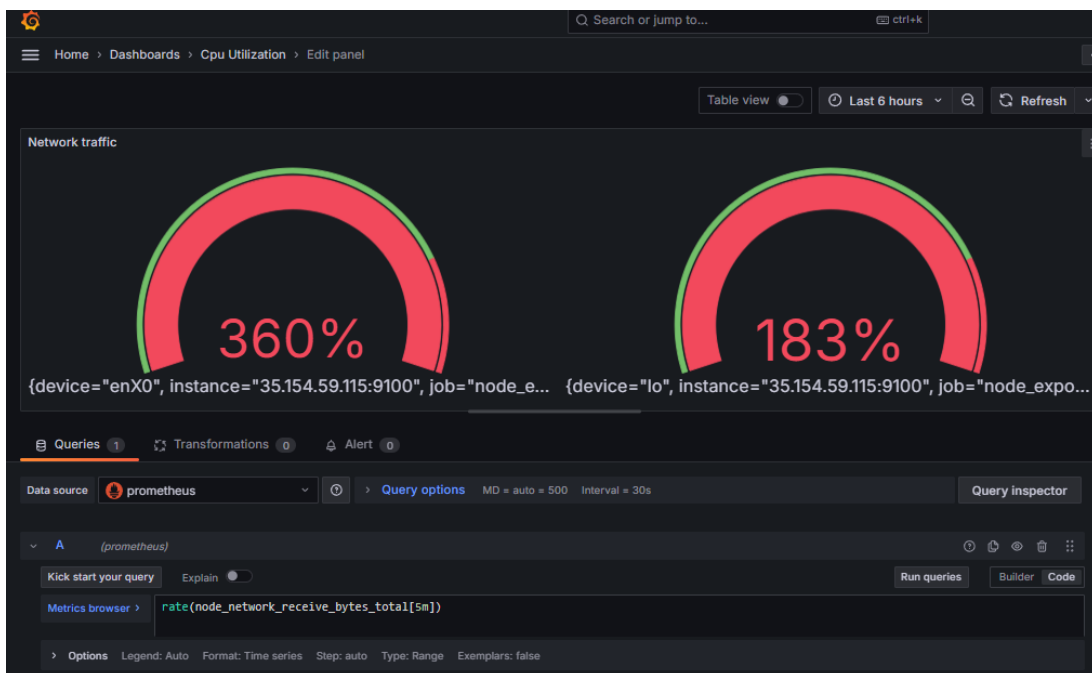
`node_filesystem_size_bytes` (Total Disk Size)



4. Network traffic

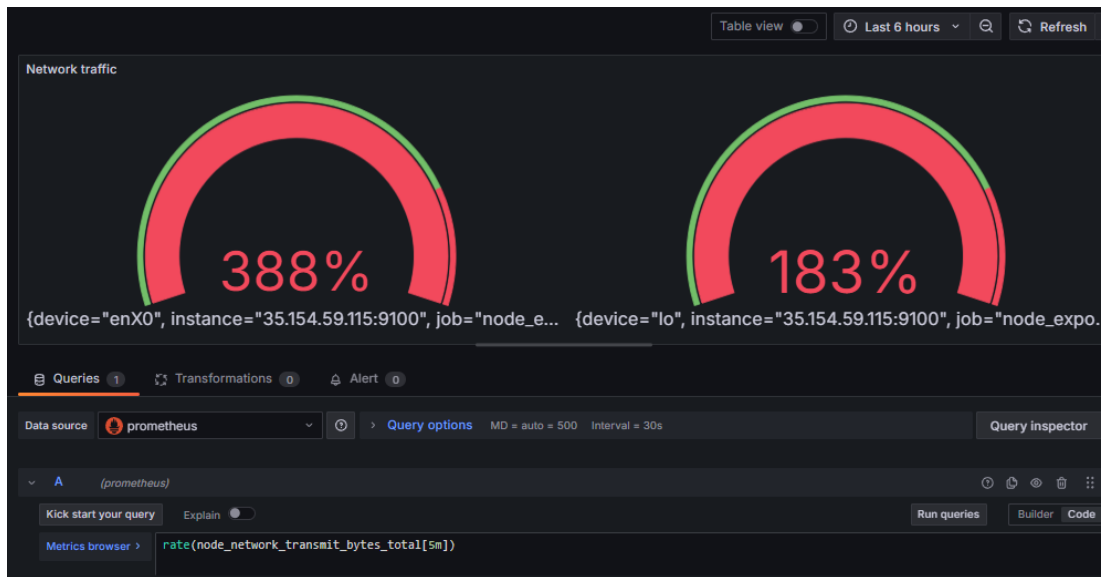
Incoming traffic:

`rate(node_network_receive_bytes_total[5m])`



Outgoing traffic

`rate(node_network_transmit_bytes_total[5m])`



Step 20: Make sure to terminate all the instances

Instances (3/3) Info Last updated 1 minute ago Connect Instance state Actions Launch instances

Find Instance by attribute or tag (case-sensitive) All states

Instance state = running Clear filters

	Name	Instance ID	Instance state	Instance type	Status check	Alarm stat
<input checked="" type="checkbox"/>	Jenkins	i-019aa04bcd40d7c13	Terminated	t2.medium	2/2 checks passed	View alarm
<input checked="" type="checkbox"/>	prometheus and grafana	i-0af7ea36fc4722b69	Terminated	t2.medium	2/2 checks passed	View alarm
<input checked="" type="checkbox"/>	test-server	i-01735c5c26bb77005	Terminated	t2.micro	2/2 checks passed	View alarm