Infrastructure

Start by creating the infrastructure using Terraform on AWS environment. First we will run the terraform script to create the Kubernetes cluster (Master & 2 Worker Nodes) in addition to the Monitoring instance.

Terraform

```
#### Providers
   terraform {
   required_providers {
      aws = {
      source  = "hashicorp/aws"
      version = "~> 5.0"  # Use the latest version that suits your needs
      }
   }
   }

   provider "aws" {
      region = "us-east-1"

   }

#### Use the default VPC:
   data "aws_vpc" "Default-VPC" {
      default = true
   }

#### Generate a Key Pair and exporting the DEPI-KeyPair.pem file:
   resource "tls_private_key" "DEPI-Key" {
      algorithm = "RSA"
      rsa_bits = 4096
   }

   # Create the key pair using the public key generated above
   resource "aws_key_pair" "DEPI-KeyPair" {
      key_name = "DEPI-KeyPair"
      public_key = tls_private_key.DEPI-Key.public_key_openssh
   }

   # Create a local file to save the private key
   resource "local_file" "KeyPair" {
      content = tls_private_key.DEPI-Key.private_key_pem
      filename = "DEPI-KeyPair.pem"
   }

   # Output the private key path
   output "private_key_path" {
      value = local_file.KeyPair.filename
```

```
    }

#### Configure the Security Group for the instances:
    resource "aws_security_group" "DEPI-SecurityGroup" {
        vpc_id = data.aws_vpc.Default-VPC.id
        name = "DEPI-SecurityGroup"
        ingress { #To access the VMs via SSH
            from_port = 22
            to_port = 22
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }

        ingress { #For email notifications (will not be used in the project)
            from_port = 25
            to_port = 25
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }

        ingress { #Range used for most of the applications
            from_port = 3000
            to_port = 10000
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }

        ingress { #HTTP
            from_port = 80
            to_port = 80
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }

        ingress { #HTTPS
            from_port = 443
            to_port = 443
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }

        ingress { #Required when setting up Kubernetes cluster
            from_port = 6443
            to_port = 6443
            protocol = "tcp"
            cidr_blocks = ["0.0.0.0/0"]
        }
```

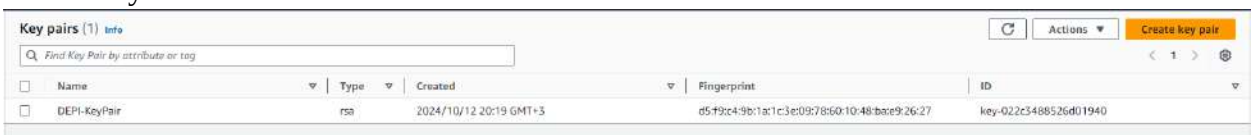
```
    ingress { #Range used to send mail notification from our Jenkins pipeline to our gmail address
        from_port = 465
        to_port = 465
        protocol = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }

    ingress { #Range used for deployment of applications
        from_port = 30000
        to_port = 32767
        protocol = "tcp"
        cidr_blocks = ["0.0.0.0/0"]
    }

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

}

#### Kubernetes Cluster

# Create the EC2 instances
resource "aws_instance" "Master" {
    ami           = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name = aws_key_pair.DEPI-KeyPair.key_name
    tags = {
        Name = "Master"
    }
}

resource "aws_instance" "Slave-01" {
    ami           = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name = aws_key_pair.DEPI-KeyPair.key_name
    tags = {
        Name = "Slave-01"
    }
}

resource "aws_instance" "Slave-02" {
    ami           = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name = aws_key_pair.DEPI-KeyPair.key_name
```

```
    tags = {
      Name = "Slave-02"
    }
  }

#### Monitoring Server
  resource "aws_instance" "Monitoring" {
    ami           = "ami-0866a3c8686eaeeba"
    instance_type = "t2.micro"
    key_name = aws_key_pair.DEPI-KeyPair.key_name
    tags = {
      Name = "Monitoring"
    }
  }
```

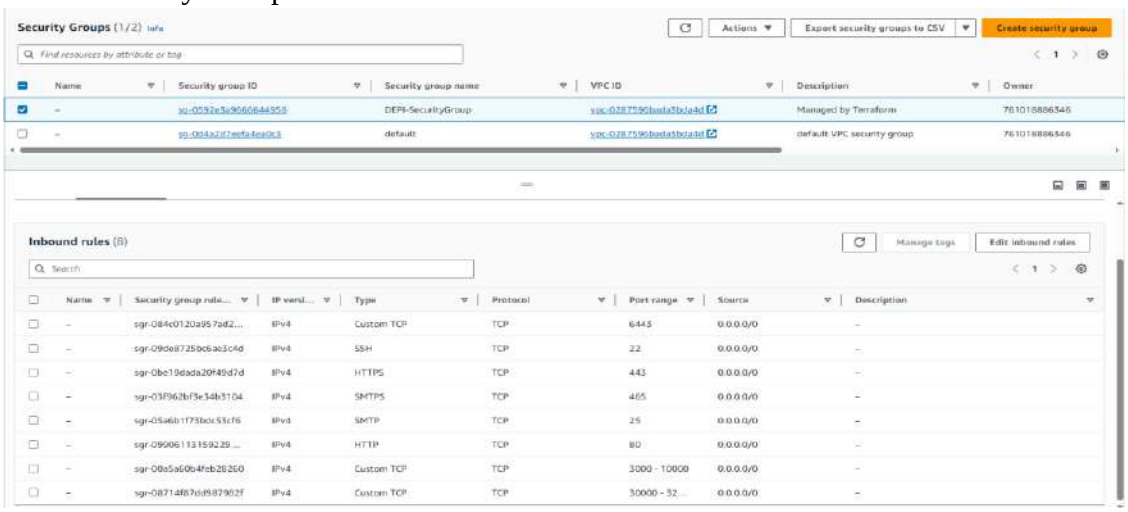
Terraform apply

1. Use the default VPC

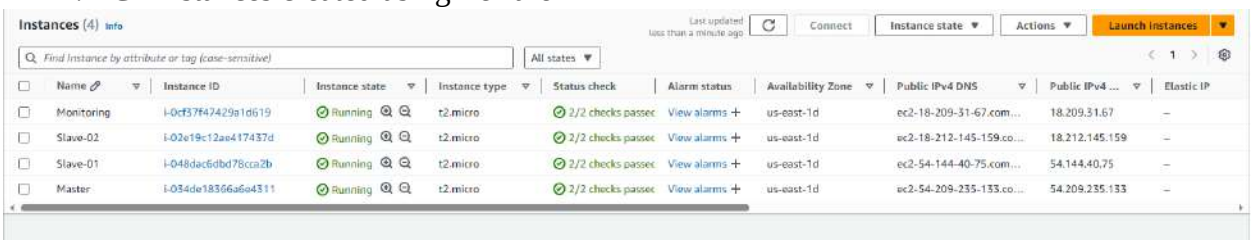
2. Key Pair



3. Security Group



4. EC2 instances created using Terraform

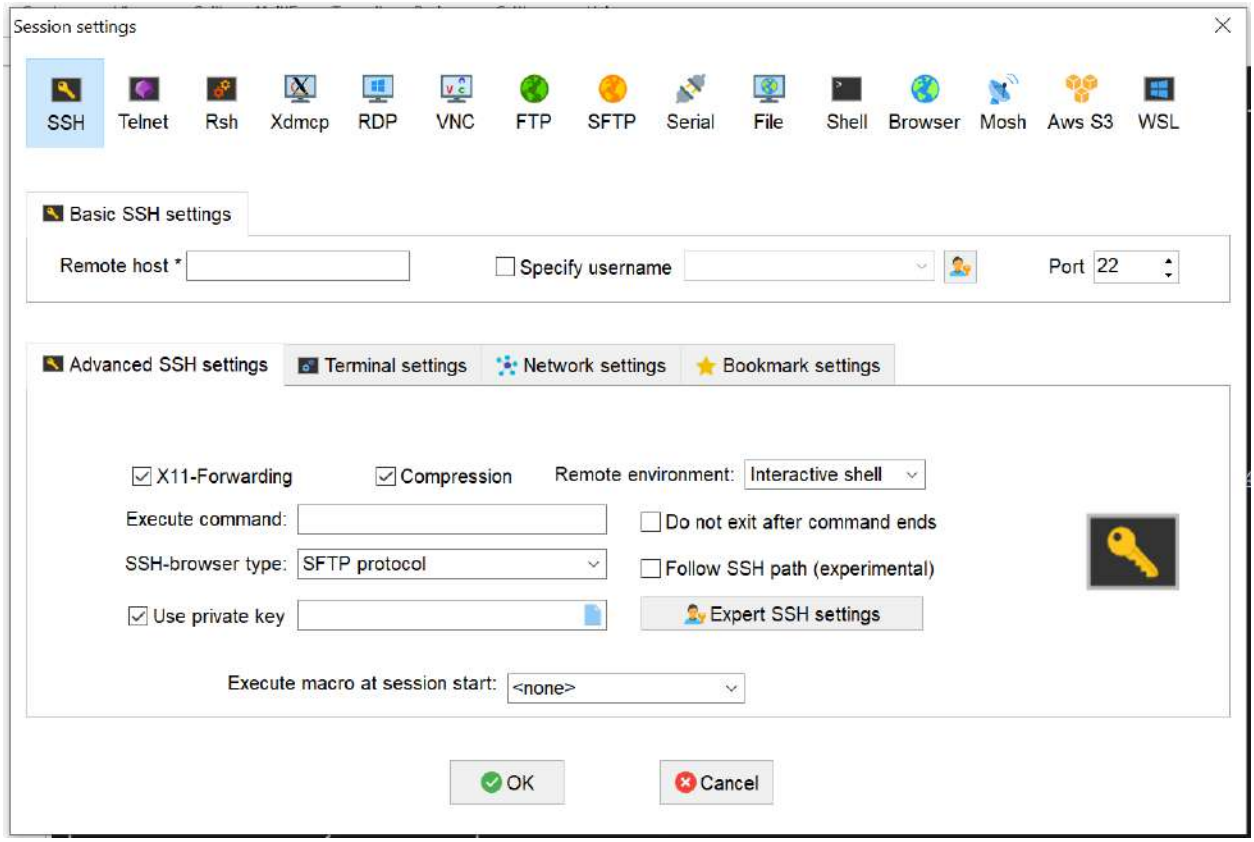


Setup the Kubernetes Cluster

1. Access the instances using MobaXterm application.

1. Create a new session.
2. Get the public IP address for each instance from AWS.

3. Copy the public IP address for each instance to the Remote host.
4. Check the Specify username box and enter “ubuntu” as the username.
5. In the Advanced SSH settings, check the Use private key box and place the .pem file.
6. Duplicate the session to create the 2 worker nodes and the Monitoring sessions as well by replacing the Remote host with each IP address.



2. Setup the Master and Worker Nodes

1. Run the below command to change to root [On Master & Worker Node]

   • sudo su

2. Create an executable file and place the following commands then run the script [On Master & Worker Node]

   # Update System Packages
   - sudo apt-get update

   # Install Docker
   - sudo apt install docker.io -y

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

Install Required Dependencies for Kubernetes

- sudo apt-get install -y apt-transport-https ca-certificates curl gnupg
- sudo mkdir -p -m 755 /etc/apt/keyrings

Add Kubernetes Repository and GPG Key

- Curl -fsSL https://pkgs.k8s.io/core:/stable:/v1.28/deb/Release.key | sudo gpg -- dearmor -o /etc/apt/keyrings/kubernetes-apt-keyring.gpg
- echo 'deb [signed-by=/etc/apt/keyrings/kubernetes-apt-keyring.gpg] https://pkgs.k8s.io/core:/stable:/v1.28/deb/ /' | sudo tee /etc/apt/sources.list.d/kubernetes.list

Update Package List

- sudo apt update

Install Kubernetes Components

- sudo apt install -y kubeadm=1.28.1-1.1 kubelet=1.28.1-1.1 kubectl=1.28.1-1.1

3. Run the following commands on the Master node only

Initialize Kubernetes Master Node

- sudo kubeadm init--pod-network-cidr=10.244.0.0/16 --ignore-preflight-errors=all

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.41.124:6443 --token d3l7q0.4mjjuwm6ux2x5vjr \
        --discovery-token-ca-cert-hash sha256:1fba7145bb3c4a7f7e0cbeea5122b4bbae9f210226b5b8de4652239c2cb73f5b
```

After running the above command then our vm will acts as master node and it will generate token to connect this with slave node-copy the token and run the command in slave machines 1 & 2

```
root@ip-172-31-41-124:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES           AGE     VERSION
ip-172-31-37-132    Ready    <none>          11s     v1.28.1
ip-172-31-38-48     Ready    <none>          15s     v1.28.1
ip-172-31-41-124    Ready    control-plane   6m18s   v1.28.1
```

Configure Kubernetes Cluster

- mkdir -p $HOME/.kube
- sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
- sudo chown $(id -u):$(id -g) $HOME/.kube/config

Deploy Networking Solution (Calico)

- kubectl apply -f https://docs.projectcalico.org/v3.20/manifests/calico.yaml

Deploy Ingress Controller (NGINX)

- kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml

4. We'll Scan Kubernetes Cluster For Any Kind Of Issues Using Cube Audit

Go To The Website & Copy The Linux_amd_64 Link

- https://github.com/shopify/kubeaudit/releases

Paste It Using wget Command

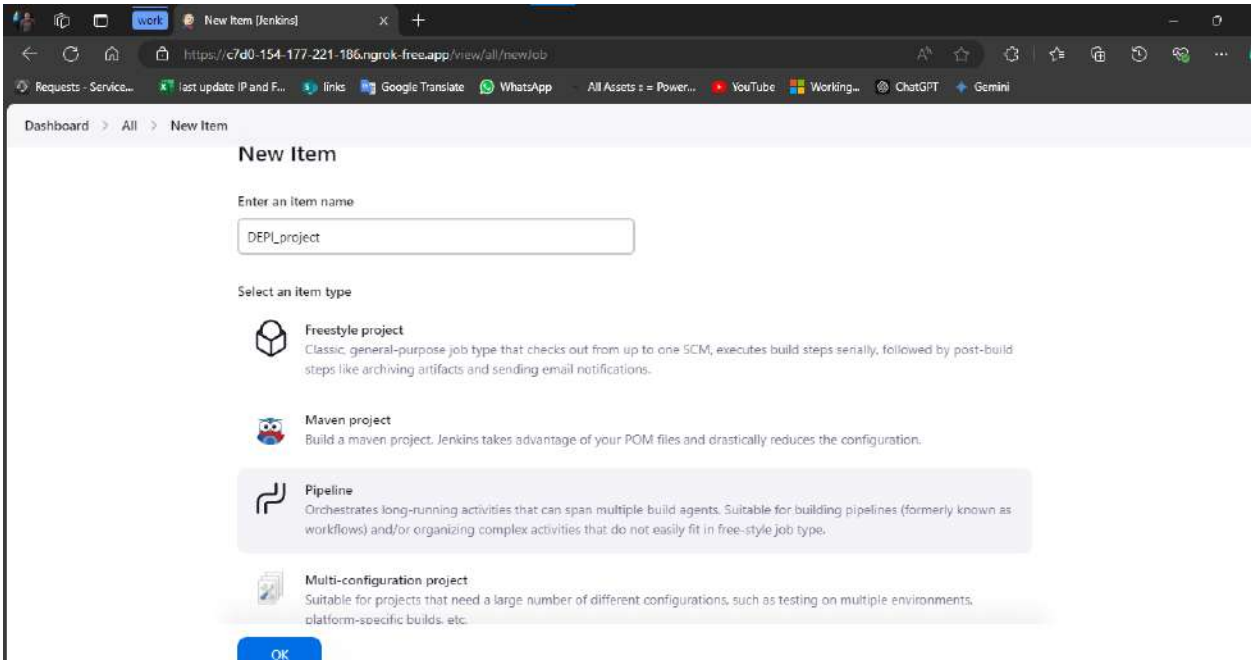
Now Untar The File Using tar-xvf File Name

sudo mv kubeaudit /usr/local/bin/->kubeaudit all

```
root@ip-172-31-41-124:/home/ubuntu# wget https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux
_amd64.tar.gz
--2024-10-12 17:26:37--  https://github.com/Shopify/kubeaudit/releases/download/v0.22.2/kubeaudit_0.22.2_linux_amd64.tar.gz
Resolving github.com (github.com)... 140.82.113.3
Connecting to github.com (github.com)|140.82.113.3|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://objects.githubusercontent.com/github-production-release-asset-2e65be/103579225/b7d91b50-ef71-4085-9d2a-5db1f
f0f7779?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241012%2Fus-east-1%2Fs3%2Faws4_request&X-
Amz-Date=20241012T172638Z&X-Amz-Expires=300&X-Amz-Signature=03d7733cef32ee942b47b30c929fe0dafa6c057257403e98eb8863a5d378516d&X
-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkubeaudit_0.22.2_linux_amd64.tar.gz&response-c
ontent-type=application%2Foctet-stream [following]
--2024-10-12 17:26:38--  https://objects.githubusercontent.com/github-production-release-asset-2e65be/103579225/b7d91b50-ef71-
4085-9d2a-5db1ff0f7779?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=releaseassetproduction%2F20241012%2Fus-east-1%2Fs3%2F
aws4_request&X-Amz-Date=20241012T172638Z&X-Amz-Expires=300&X-Amz-Signature=03d7733cef32ee942b47b30c929fe0dafa6c057257403e98eb8
863a5d378516d&X-Amz-SignedHeaders=host&response-content-disposition=attachment%3B%20filename%3Dkubeaudit_0.22.2_linux_amd64.ta
r.gz&response-content-type=application%2Foctet-stream
Resolving objects.githubusercontent.com (objects.githubusercontent.com)... 185.199.110.133, 185.199.111.133, 185.199.108.133,
...
Connecting to objects.githubusercontent.com (objects.githubusercontent.com)|185.199.110.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10181094 (9.7M) [application/octet-stream]
Saving to: ‘kubeaudit_0.22.2_linux_amd64.tar.gz’

kubeaudit_0.22.2_linux_amd64.ta 100%[===================================================>]   9.71M  18.0MB/s    in 0.5s

2024-10-12 17:26:38 (18.0 MB/s) - ‘kubeaudit_0.22.2_linux_amd64.tar.gz’ saved [10181094/10181094]

root@ip-172-31-41-124:/home/ubuntu# tar -xvf kubeaudit_0.22.2_linux_amd64.tar.gz
README.md
kubeaudit
root@ip-172-31-41-124:/home/ubuntu# sudo mv kubeaudit /usr/local/bin/
root@ip-172-31-41-124:/home/ubuntu# kubeaudit all
```

CI / CD Pipeline

Open Jenkins then Start new project



Jenkins Plugin Installation Steps

1. **Eclipse Temurin Installer**
   - Navigate to **Jenkins Dashboard** -> **Manage Jenkins** -> **Manage Plugins**.
   - Go to the **Available** tab, search for **Eclipse Temurin Installer**.
   - Select it and click **Install without restart**.
2. **Pipeline Maven Integration**
   - Repeat the above steps, searching for **Pipeline Maven Integration**.
   - Select and click **Install without restart**.
3. **Config File Provider**
   - Follow the same procedure, searching for **Config File Provider**.
   - Select and click **Install without restart**.
4. **SonarQube Scanner**
   - Search for **SonarQube Scanner** in the **Available** tab.
   - Select and click **Install without restart**.
5. **Kubernetes CLI**
   - Search for **Kubernetes CLI** and select it.

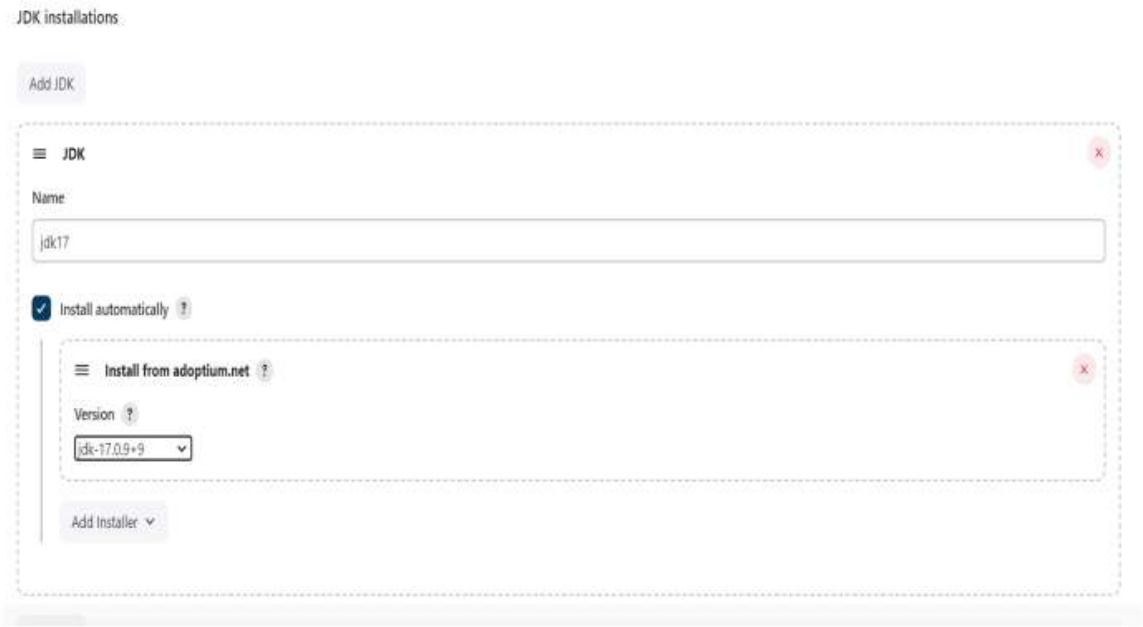
- Click **Install without restart**.

6. **Kubernetes**
   - Find **Kubernetes** in the **Available** tab.
   - Select and click **Install without restart**.
7. **Docker**
   - Search for **Docker** and select it.
   - Click **Install without restart**.
8. **Docker Pipeline Step**
   - Search for **Docker Pipeline Step**.
   - Select and click **Install without restart**.
9. **Prometheus metrics**
   - Search for **Prometheus metrics** and select it.
   - Select and click **Install with restart**.
10. **Slack**
   - Finally, search for Slack Notification
   - Select and click **Install with restart**.

Configure Above Plugins in Jenkins Pipeline

Configure the tools choose manage jenkins → Tools

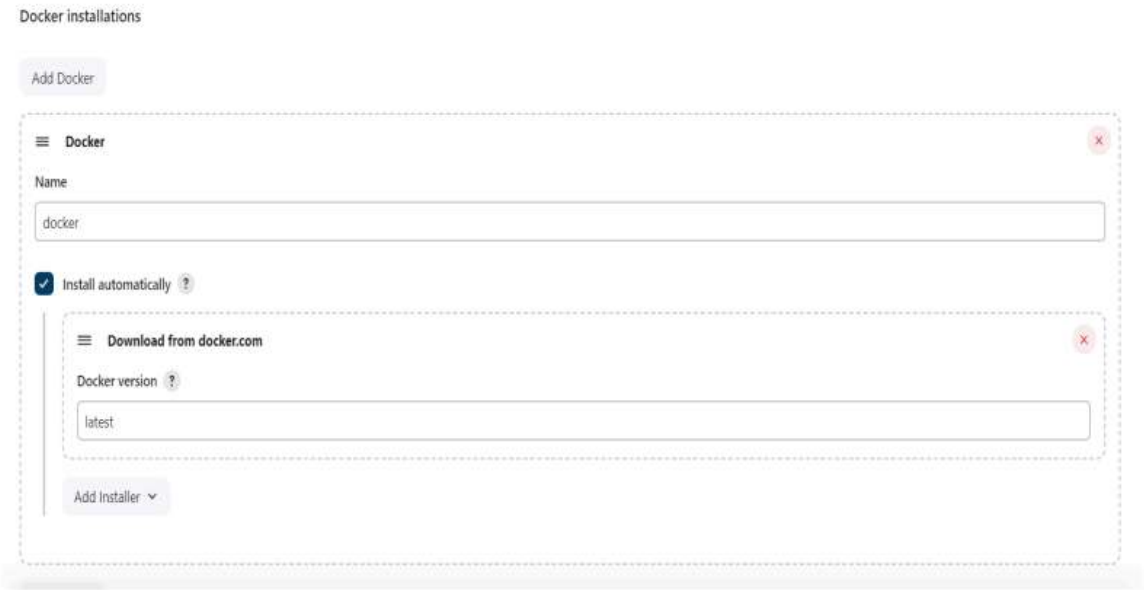
1. Choose jdk and fill as given below



2. choose sonarqube scanner and configure

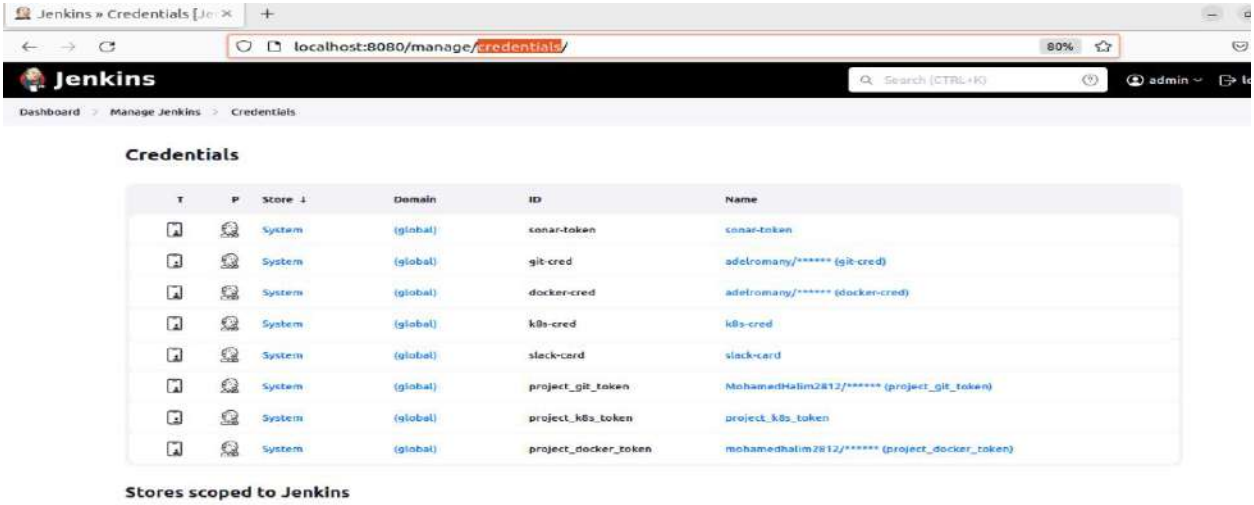
3. choose maven and Configure

4. choose Docker and Configure

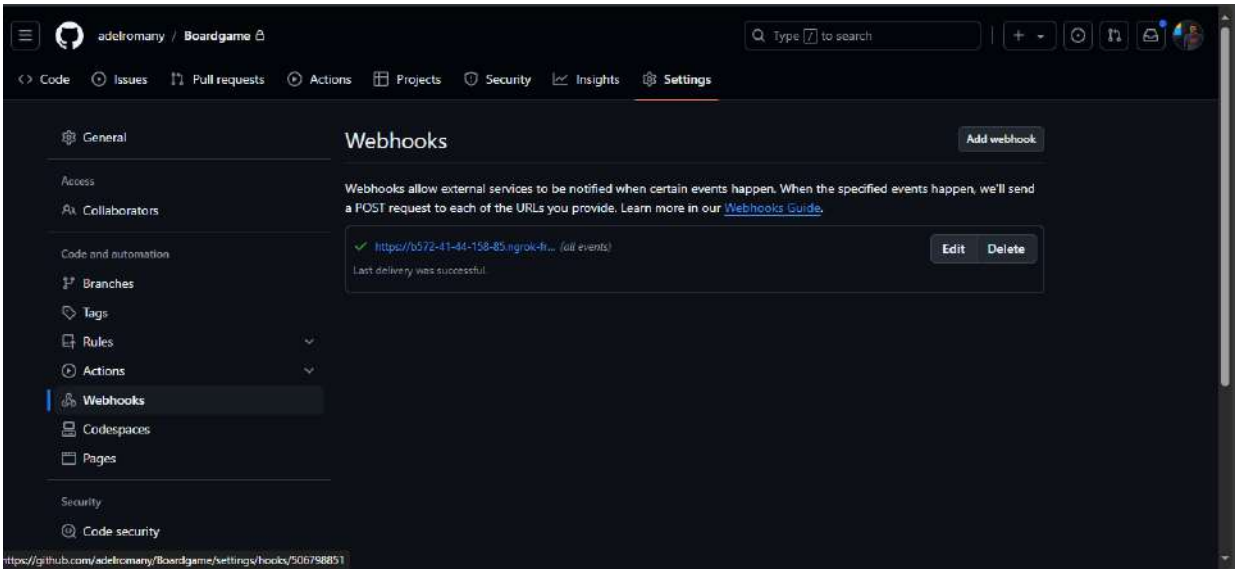


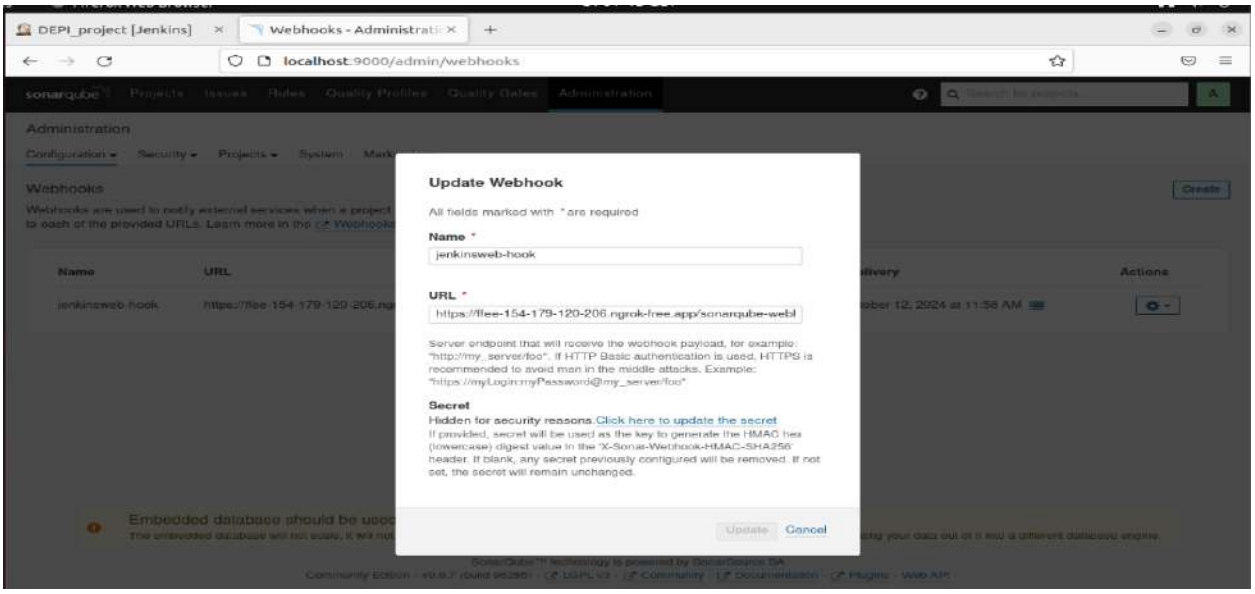
1. choose Slack and Configure
    a. **follow steps from https://plugins.jenkins.io/slack/**

Create Credentials



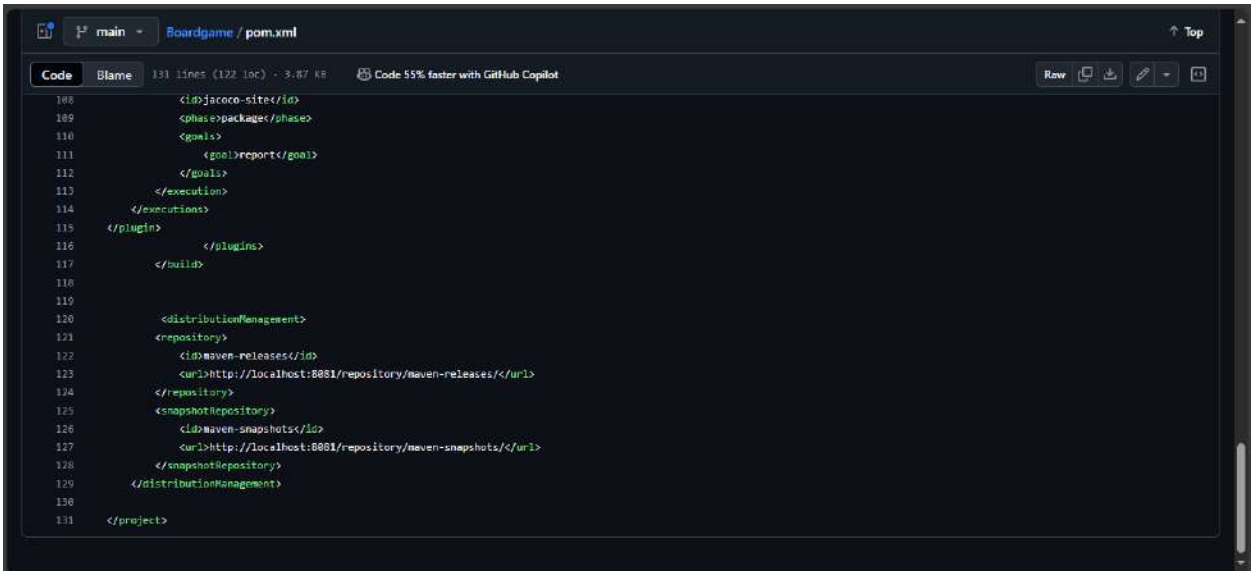
configure webhooks in GitHub and SonarQube





Configure nexus on the pom.xml

```
    <distributionManagement>
  <repository>
    <id>maven-releases</id>
    <url>http://localhost:8081/repository/maven-releases/</url>
  </repository>
  <snapshotRepository>
    <id>maven-snapshots</id>
    <url>http://localhost:8081/repository/maven-snapshots/</url>
  </snapshotRepository>
</distributionManagement>
```



The pipeline:

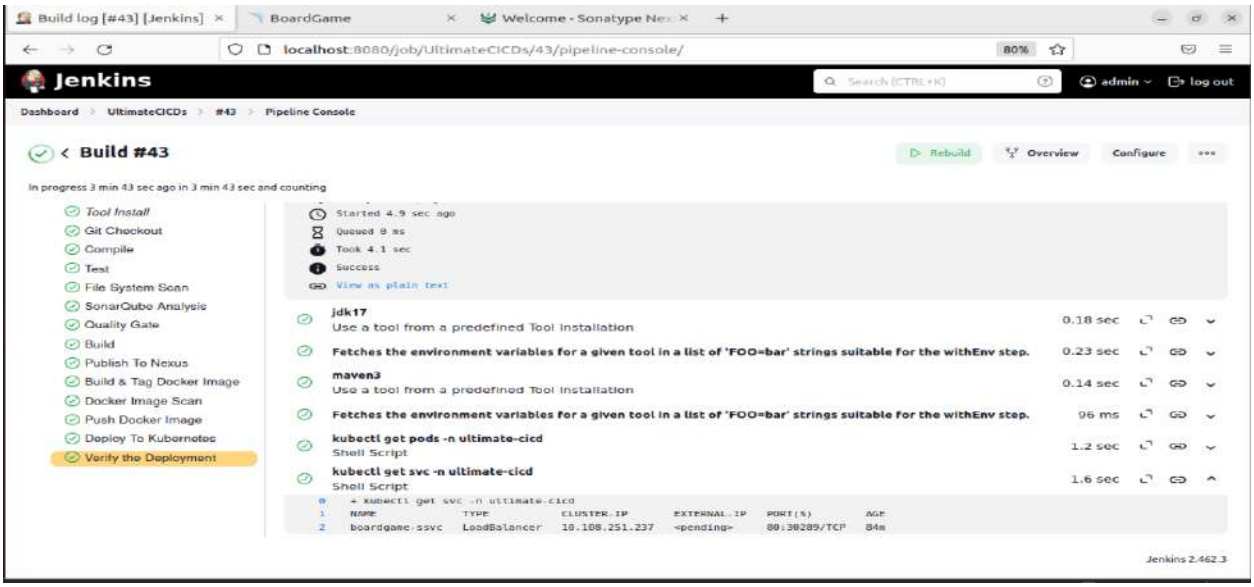
```
pipeline {
    agent any


    tools {
        jdk 'jdk17'
        maven 'maven3'
    }


    environment {
        SCANNER_HOME = tool 'sonar-scanner'
        DOCKER_BUILDKIT = "1"
    }


    stages {
        stage('Git Checkout') {
            steps {
                git branch: 'main', credentialsId: 'project_git_token', url:
'https://github.com/MohamedHalim2812/UltimateCICD.git'  //change with your git repository
            }
        }
```

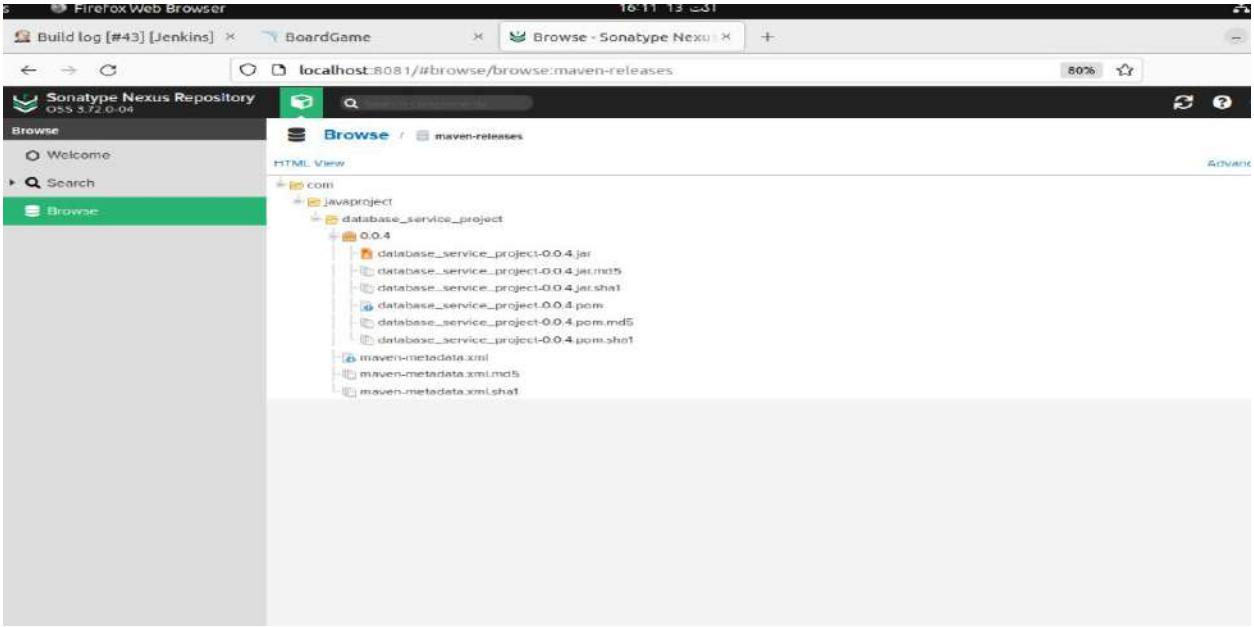
```
        stage('Compile') {
            steps {
                sh "mvn compile"
            }
        }

        stage('Test') {
            steps {
                sh "mvn test"
            }
        }

        stage('File System Scan') {
            steps {
                sh "trivy fs --format table -o trivy-fs-report.html ."
            }
        }

        stage('SonarQube Analysis') {
            steps {
                withSonarQubeEnv('sonar') {
                    sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=BoardGame -
Dsonar.projectKey=BoardGame \
                        -Dsonar.java.binaries=. '''
                }
            }
        }
        stage('Quality Gate') {
            steps {
                script {
                    waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
                }
```

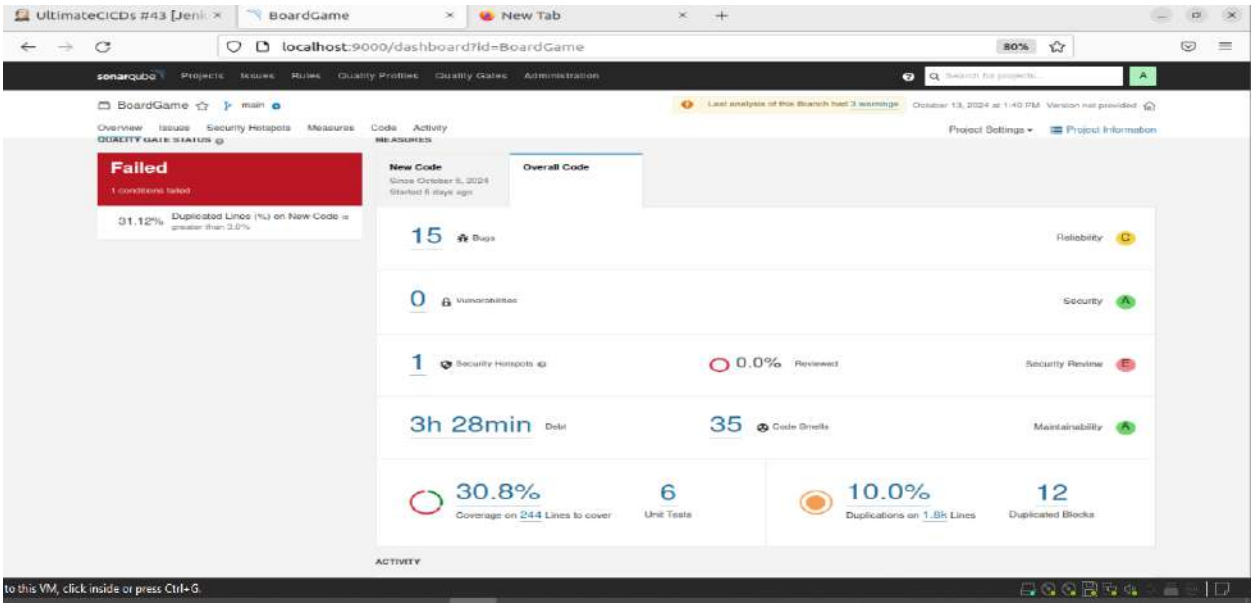
```
        }
    }

    stage('Build') {
        steps {
            sh "mvn package"
        }
    }

    stage('Publish To Nexus') {
        steps {
            withMaven(globalMavenSettingsConfig: 'global-settings', jdk: 'jdk17', maven: 'maven3',
mavenSettingsConfig: '', traceability: true) {
                sh "mvn deploy"
            }
        }
    }

    stage('Build & Tag Docker Image') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'project_docker_token', toolName: 'docker') {
                    sh "docker build -t mohamedhalim2812/boardgame:latest ."  //change to your docker account
                }
            }
        }
    }

    stage('Docker Image Scan') {
        steps {
            sh "trivy image --format table -o trivy-image-report.html mohamedhalim2812/boardgame:latest"
//change to your docker account
```

```
        }
    }

    stage('Push Docker Image') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'project_docker_token', toolName: 'docker') {
                    sh "docker push mohamedhalim2812/boardgame:latest" //change to your docker account
                }
            }
        }
    }

    stage('Deploy To Kubernetes') {
        steps {
            withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId:
'project_k8s_token', namespace: 'ultimate-cicd', restrictKubeConfigAccess: false, serverUrl:
'https://54.209.235.133') {
                sh "kubectl apply -f deployment-service.yaml"   //change with your  credentialsId
            }
        }
    }

    stage('Verify the Deployment') {
        steps {
            withKubeConfig(caCertificate: '', clusterName: 'kubernetes', contextName: '', credentialsId:
'project_k8s_token', namespace: 'ultimate-cicd', restrictKubeConfigAccess: false, serverUrl:
'https://54.209.235.133') {
                sh "kubectl get pods -n ultimate-cicd"
                sh "kubectl get svc -n ultimate-cicd"
            }
        }
    }
```

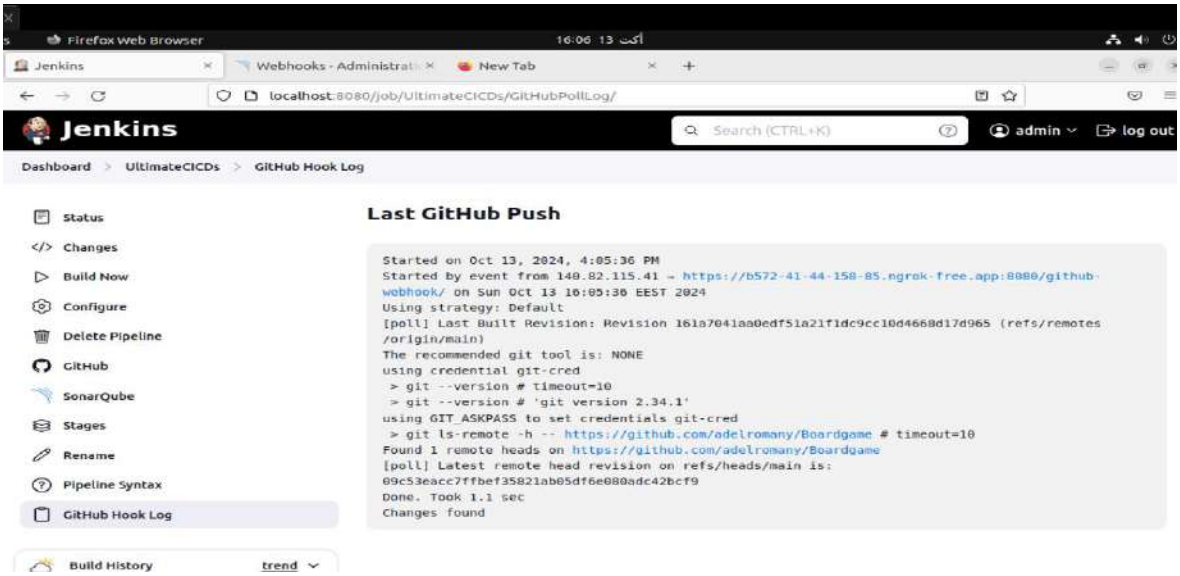
```
    }
}
```

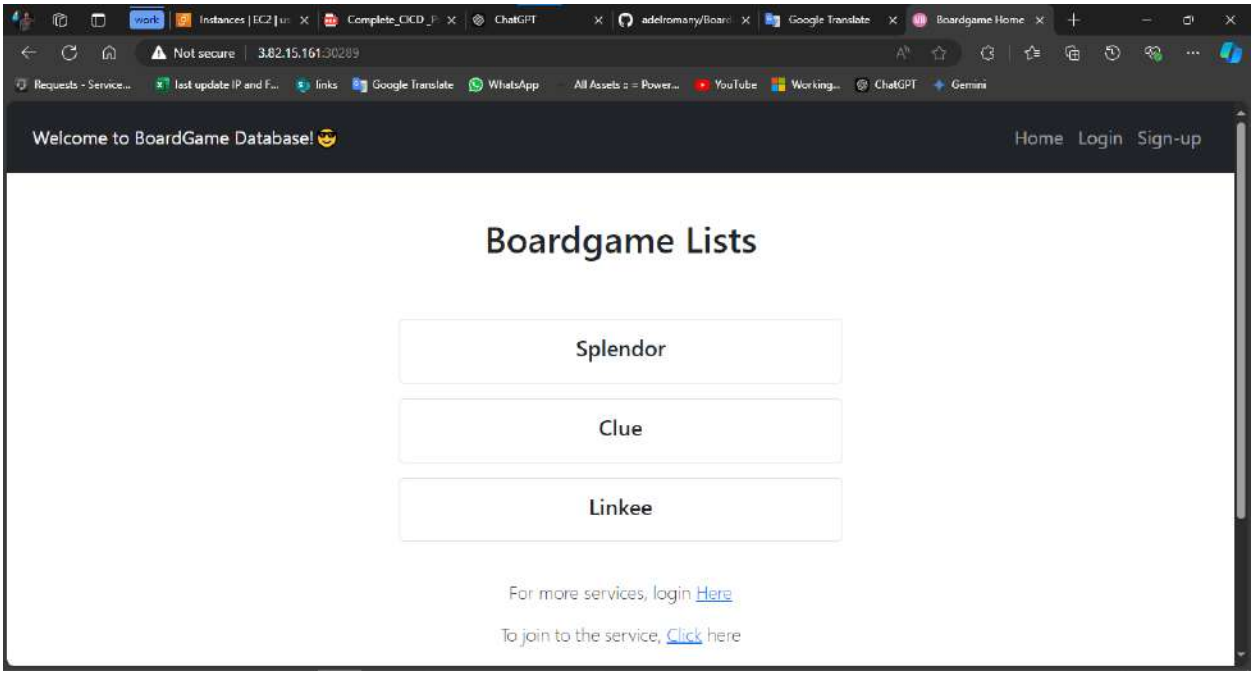

Build log [#43] [Jenkins]
BoardGame
Welcome - Sonatype Nex
localhost:8080/job/UltimateCICDs/43/pipeline-console/
Jenkins
admin
log out
Dashboard > UltimateCICDs > #43 > Pipeline Console
Build #43
Rebuild
Overview
Configure
In progress 3 min 43 sec ago in 3 min 43 sec and counting
Tool Install
Git Checkout
Compile
Test
File System Scan
SonarQube Analysis
Quality Gate
Build
Publish To Nexus
Build & Tag Docker Image
Docker Image Scan
Push Docker Image
Deploy To Kubernetes
Verify the Deployment
Started 4.9 sec ago
Queued 0 ms
Took 4.1 sec
Success
View as plain text
jdk17
Use a tool from a predefined Tool Installation
0.18 sec
Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.
0.23 sec
maven3
Use a tool from a predefined Tool Installation
0.14 sec
Fetches the environment variables for a given tool in a list of 'FOO=bar' strings suitable for the withEnv step.
96 ms
kubectl get pods -n ultimate-cicd
Shell Script
1.2 sec
kubectl get svc -n ultimate-cicd
Shell Script
1.6 sec
+ kubectl get svc -n ultimate-cicd
NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE
boardgame-ssvc LoadBalancer 10.108.251.237 <pending> 80:30289/TCP 84s
Jenkins 2.462.3

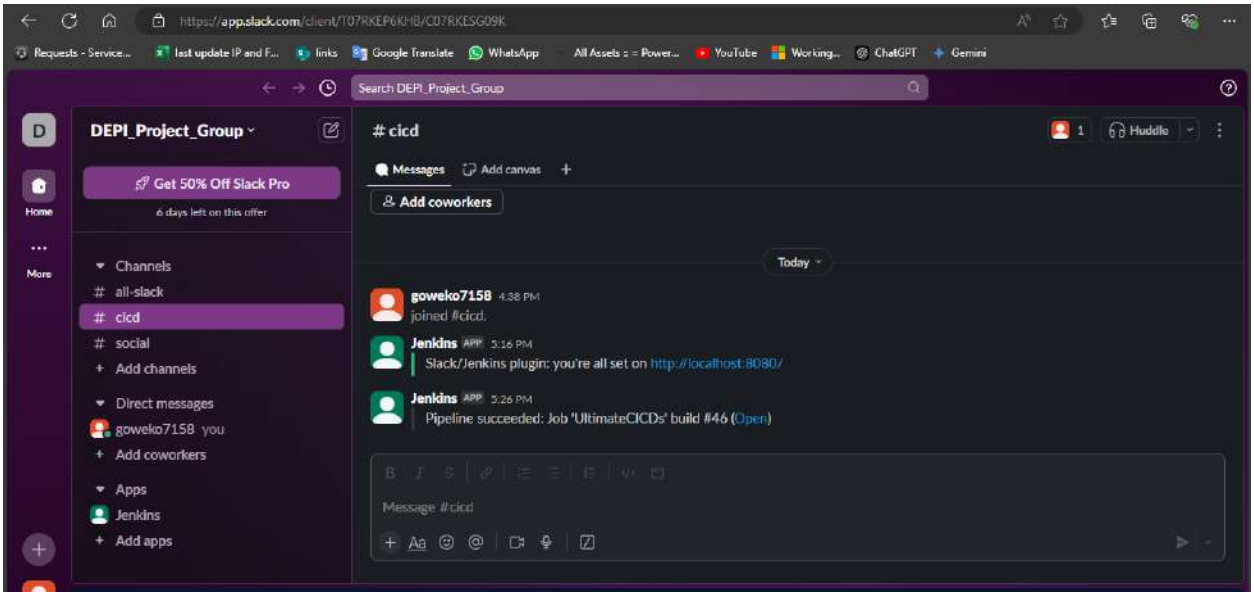

Firefox Web Browser
Build log [#43] [Jenkins]
BoardGame
Browse - Sonatype Nexu
localhost:8081/#browse/browse:maven-releases
Sonatype Nexus Repository
OSS 3.72.0-04
Browse
Welcome
Search
Browse
Browse / maven-releases
HTML View
com
javaproject
database_service_project
0.0.4
database_service_project-0.0.4.jar
database_service_project-0.0.4.jar.md5
database_service_project-0.0.4.jar.sha1
database_service_project-0.0.4.pom
database_service_project-0.0.4.pom.md5
database_service_project-0.0.4.pom.sha1
maven-metadata.xml
maven-metadata.xml.md5
maven-metadata.xml.sha1


UltimateCICDs #43 [Jenk
BoardGame
New Tab
localhost:9000/dashboard?id=BoardGame
sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
BoardGame
main
Overview
Issues
Security Hotspots
Measures
Code
Activity
Project Settings
Project Information
Failed
1 conditions failed
31.12% Duplicated Lines (%) on New Code is greater than 3.0%
New Code
Since October 5, 2024
Started 8 days ago
Overall Code
15 Bugs
Reliability C
0 Vulnerabilities
Security A
1 Security Hotspots
0.0% Reviewed
Security Review E
3h 28min Debt
35 Code Smells
Maintainability A
30.8%
Coverage on 244 Lines to cover
6
Unit Tests
10.0%
Duplications on 1.8k Lines
12
Duplicated Blocks
ACTIVITY
to this VM, click inside or press Ctrl+G.


Filter...
#43
(pending—Finished waiting)


Firefox Web Browser
16:06 13 أكت
Jenkins
Webhooks - Administrati
New Tab
localhost:8080/job/UltimateCICDs/GitHubPollLog/
Jenkins
Search (CTRL+K)
admin
log out
Dashboard > UltimateCICDs > GitHub Hook Log
Status
Changes
Build Now
Configure
Delete Pipeline
GitHub
SonarQube
Stages
Rename
Pipeline Syntax
GitHub Hook Log
Build History
trend
Last GitHub Push
Started on Oct 13, 2024, 4:05:36 PM
Started by event from 140.82.115.41 → https://b572-41-44-158-85.ngrok-free.app:8080/github-webhook/ on Sun Oct 13 16:05:36 EEST 2024
Using strategy: Default
[poll] Last Built Revision: Revision 161a7041aa0edf51a21f1dc9cc10d4668d17d965 (refs/remotes/origin/main)
The recommended git tool is: NONE
using credential git-cred
> git --version # timeout=10
> git --version # 'git version 2.34.1'
using GIT_ASKPASS to set credentials git-cred
> git ls-remote -h -- https://github.com/adelromany/Boardgame # timeout=10
Found 1 remote heads on https://github.com/adelromany/Boardgame
[poll] Latest remote head revision on refs/heads/main is:
09c53eacc7ffbef35821ab05df6e080adc42bcf9
Done. Took 1.1 sec
Changes found

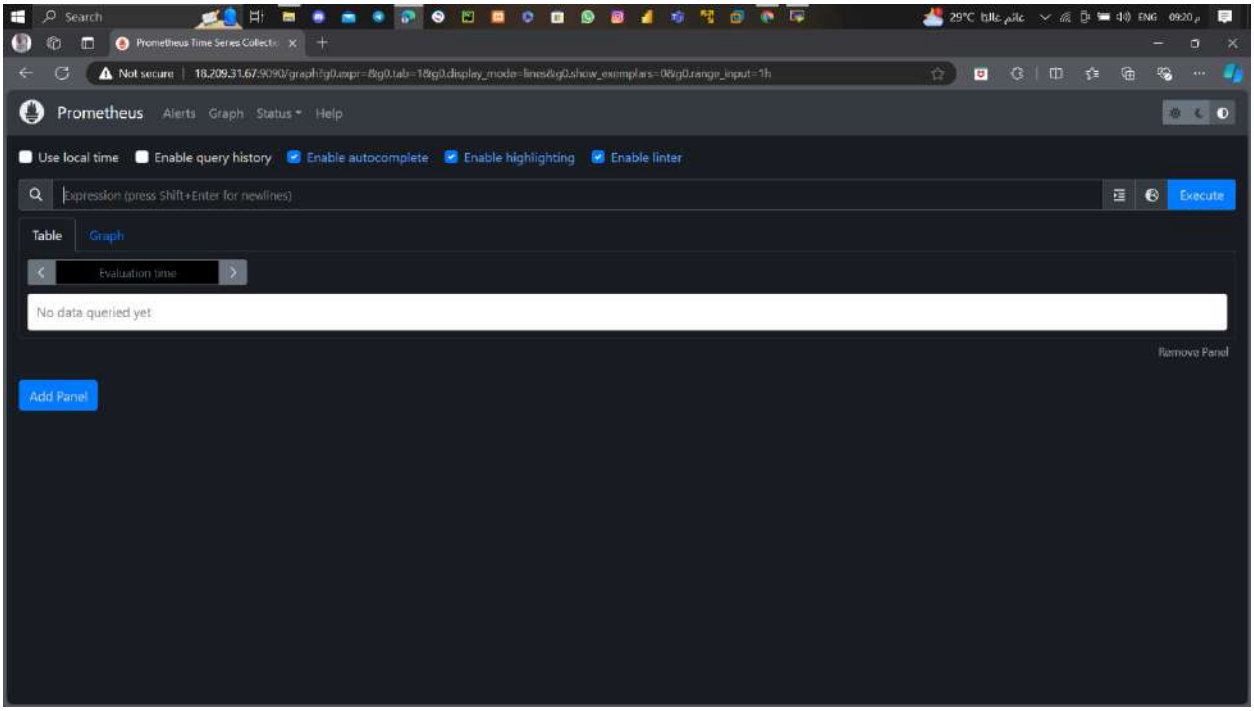

Not secure | 3.82.15.161:30289
Welcome to BoardGame Database!
Home Login Sign-up
Boardgame Lists
Splendor
Clue
Linkee
For more services, login Here
To join to the service, Click here


DEPI_Project_Group
Get 50% Off Slack Pro
6 days left on this offer
Channels
all-slack
cicd
social
Add channels
Direct messages
goweko7158 you
Add coworkers
Apps
Jenkins
Add apps
cicd
Messages
Add canvas
Add coworkers
Today
goweko7158 4:38 PM
joined #cicd.
Jenkins APP 5:16 PM
Slack/Jenkins plugin: you're all set on http://localhost:8080/
Jenkins APP 5:26 PM
Pipeline succeeded: Job 'UltimateCICDs' build #46 (Open)
Message #cicd

Monitoring

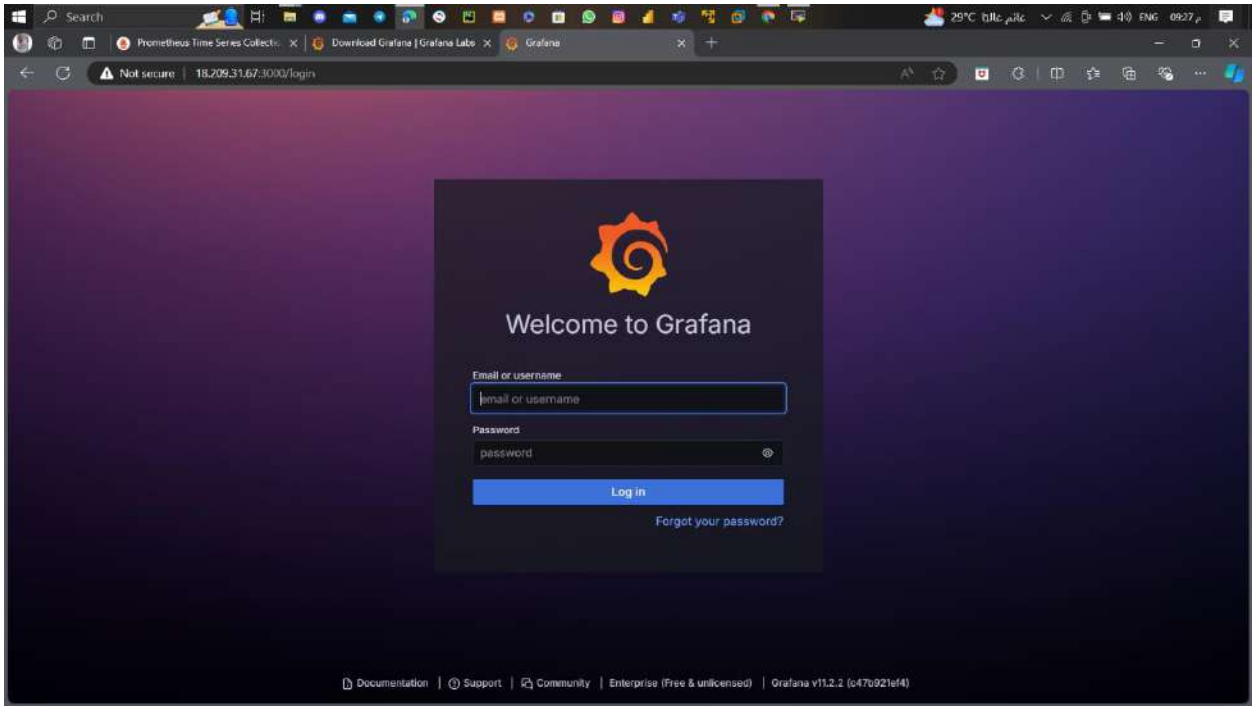
##Installing Prometheus

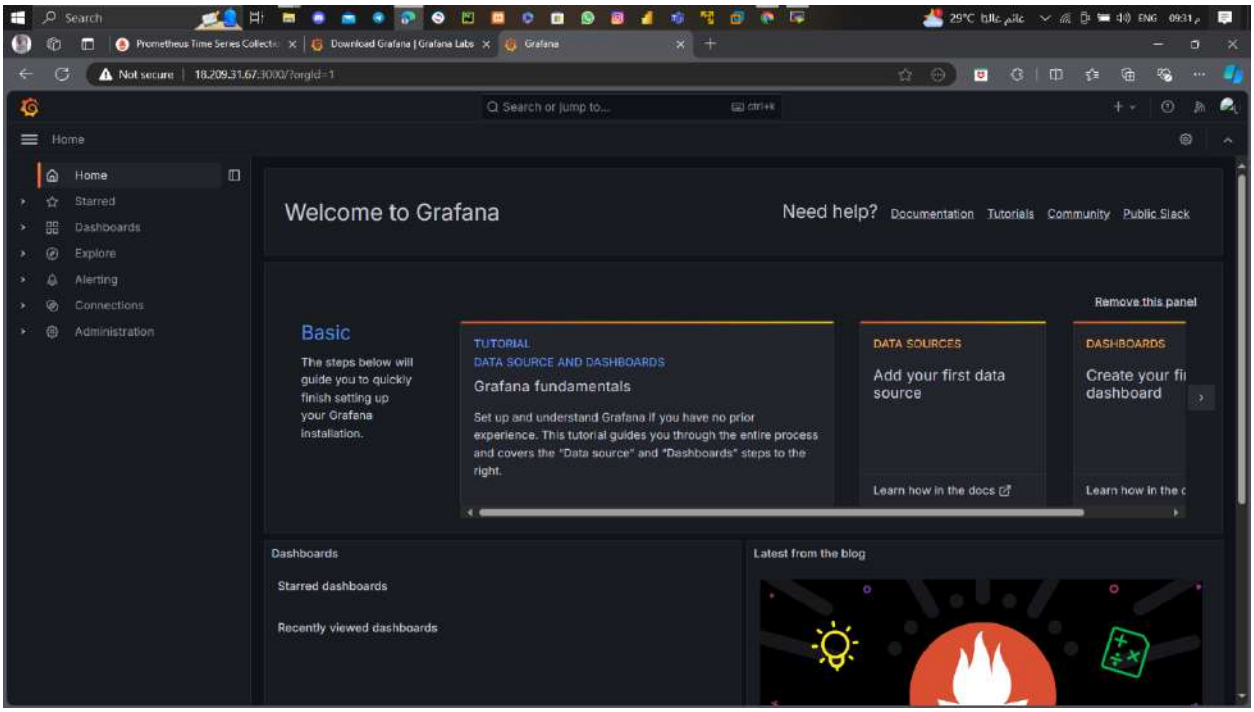
- sudo apt update
- wget https://github.com/prometheus/prometheus/releases/download/v2.53.2/prometheus-2.53.2.linux-amd64.tar.gz
- tar -xvf prometheus-2.53.2.linux-amd64.tar.gz
- cd prometheus-2.53.2.linux-amd64/
- ./prometheus & ## to start Prometheus
- 18.209.31.67:9090 ## to access prometheus



##Installing Grafana

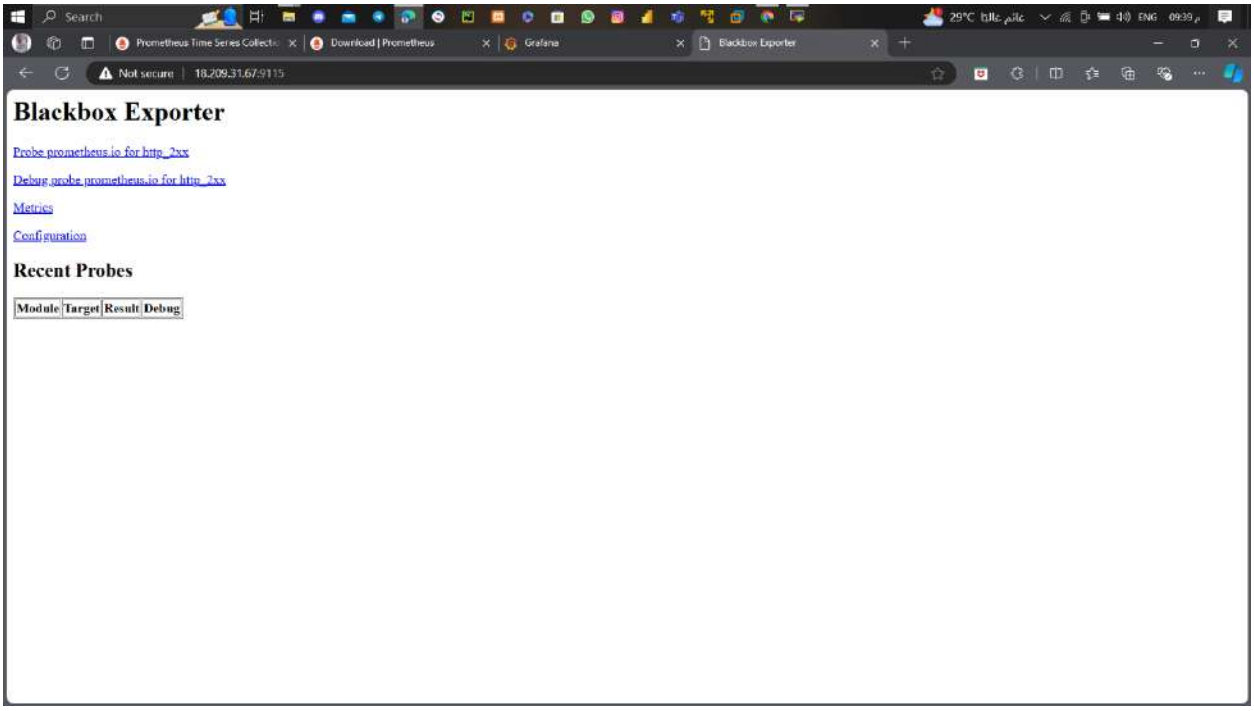
- sudo apt-get install -y adduser libfontconfig1 musl
- wget https://dl.grafana.com/enterprise/release/grafana-enterprise_11.2.2_amd64.deb
- sudo dpkg -i grafana-enterprise_11.2.2_amd64.deb
- sudo /bin/systemctl start grafana-server ## to start Grafana
- 18.209.31.67:3000 ## default username and password : admin



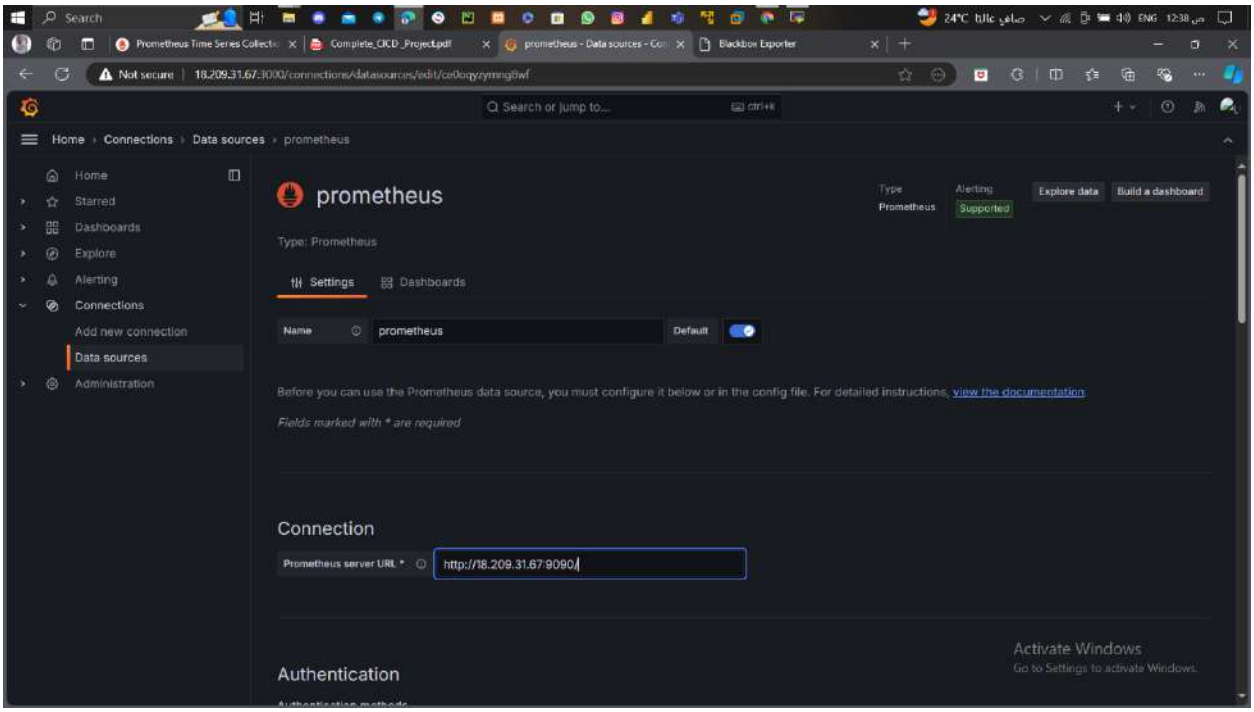


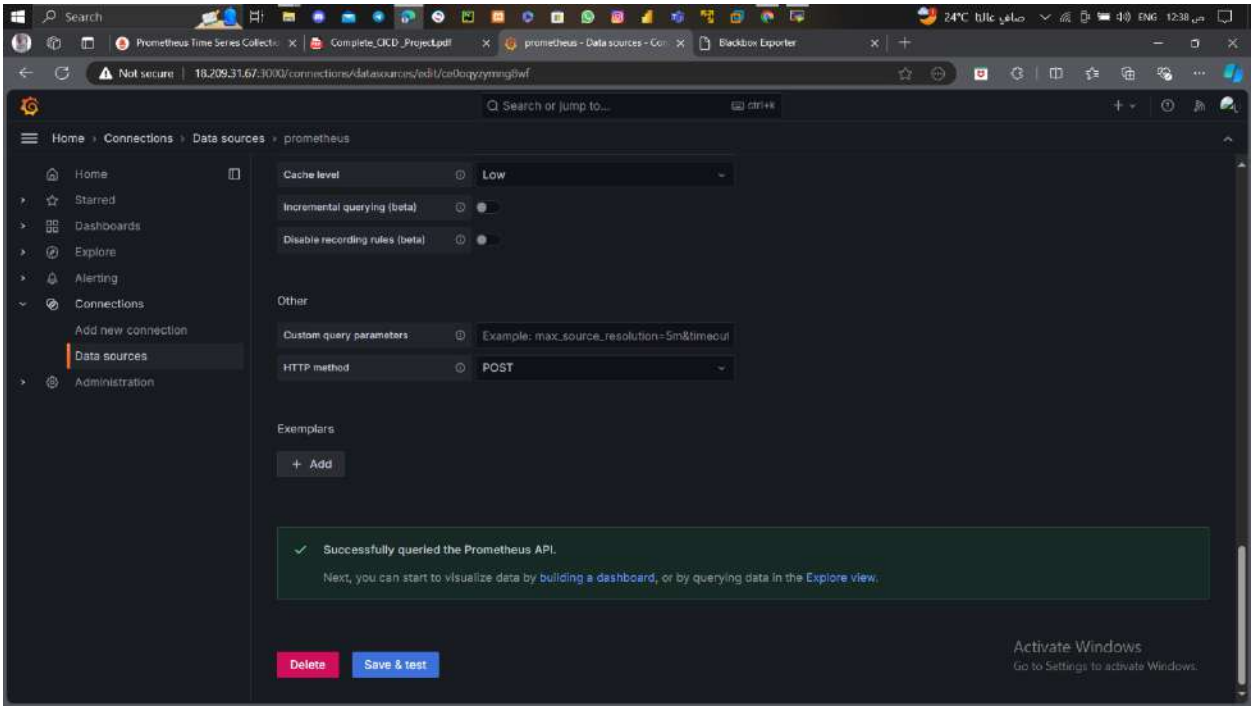
Installing blackbox_exporter

- Wget https://github.com/prometheus/blackbox_exporter/releases/download/v0.25.0/blackbox_exporter-0.25.0.linux-amd64.tar.gz
- tar -xvf blackbox_exporter-0.25.0.linux-amd64.tar.gz
- cd blackbox_exporter-0.25.0.linux-amd64/
- ./blackbox_exporter & ##to start blackbox
- 18.209.31.67:9115



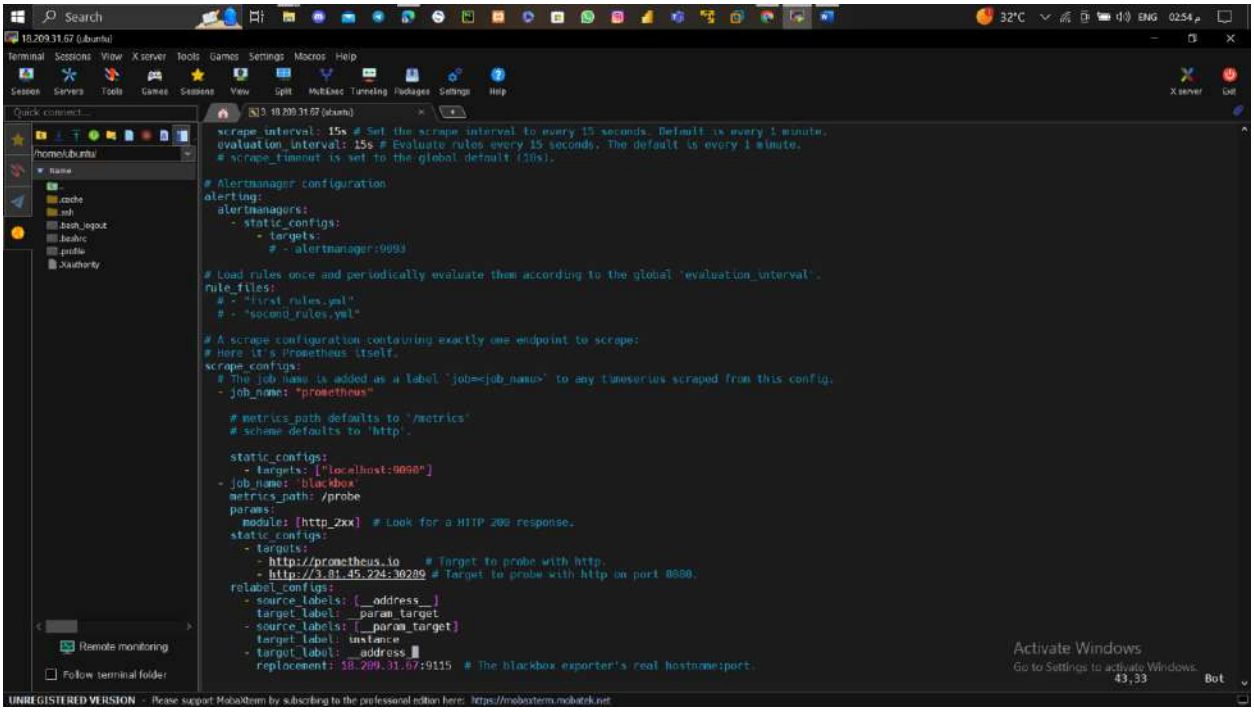
##Adding Prometheus as a Data Source to Grafana





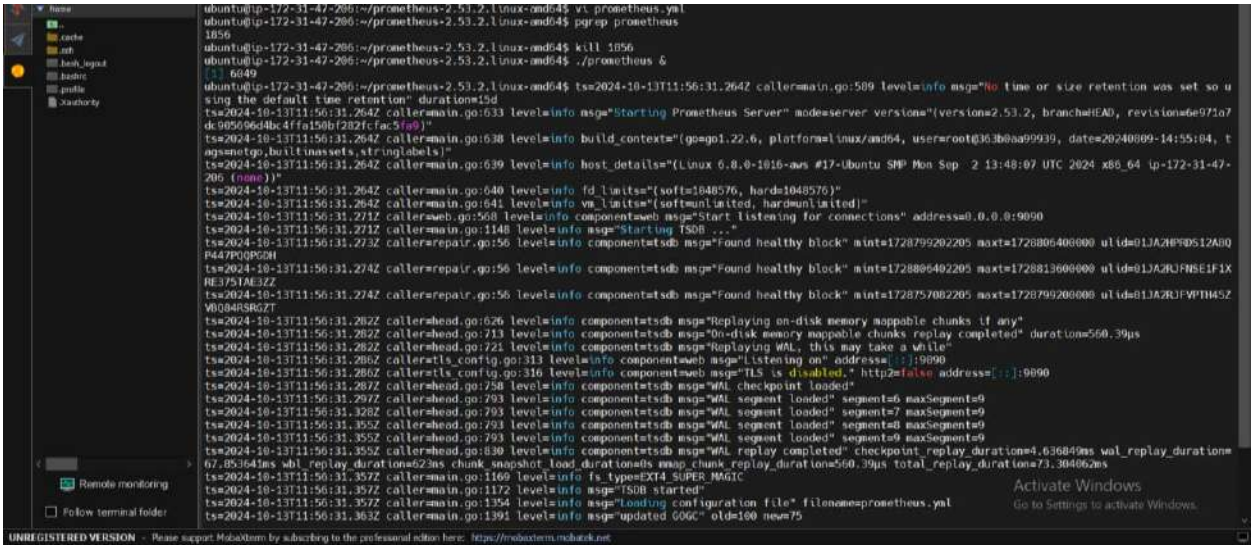
Monitoring the website

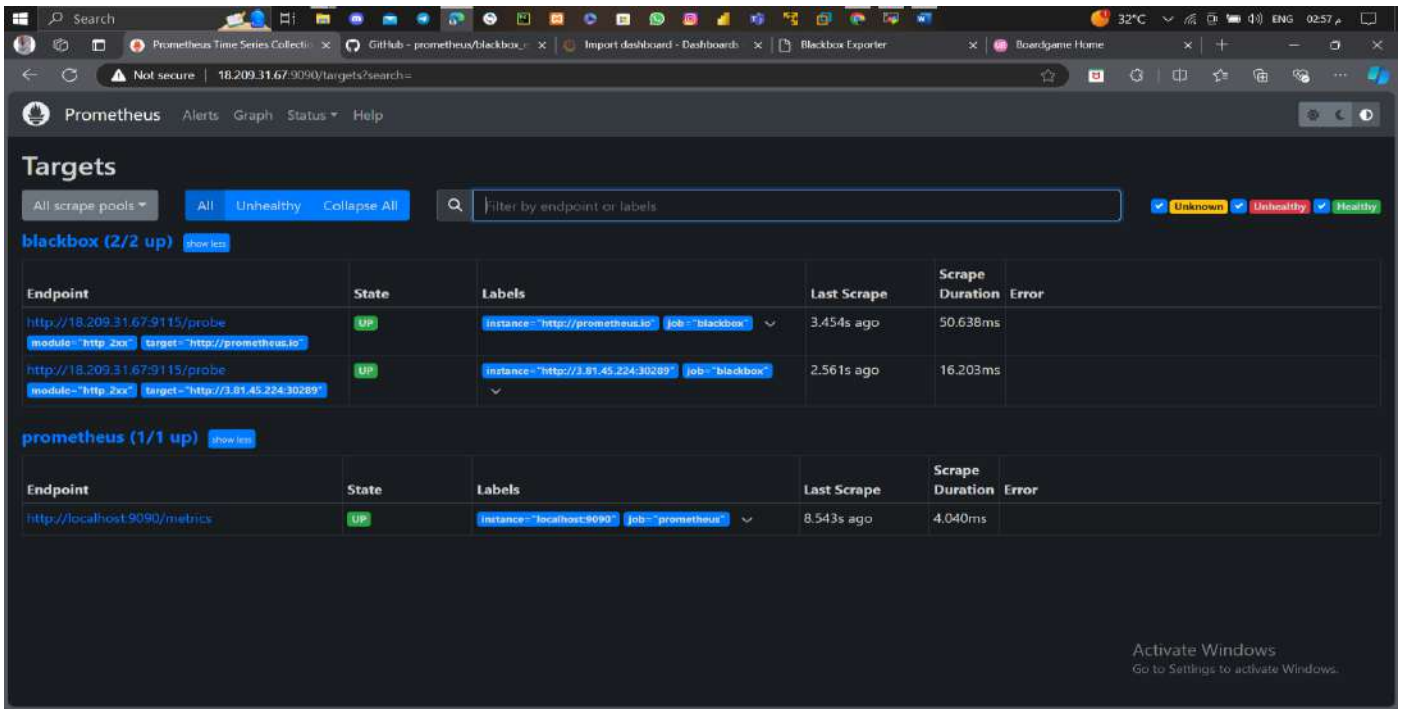
- Prometheus yaml configuration



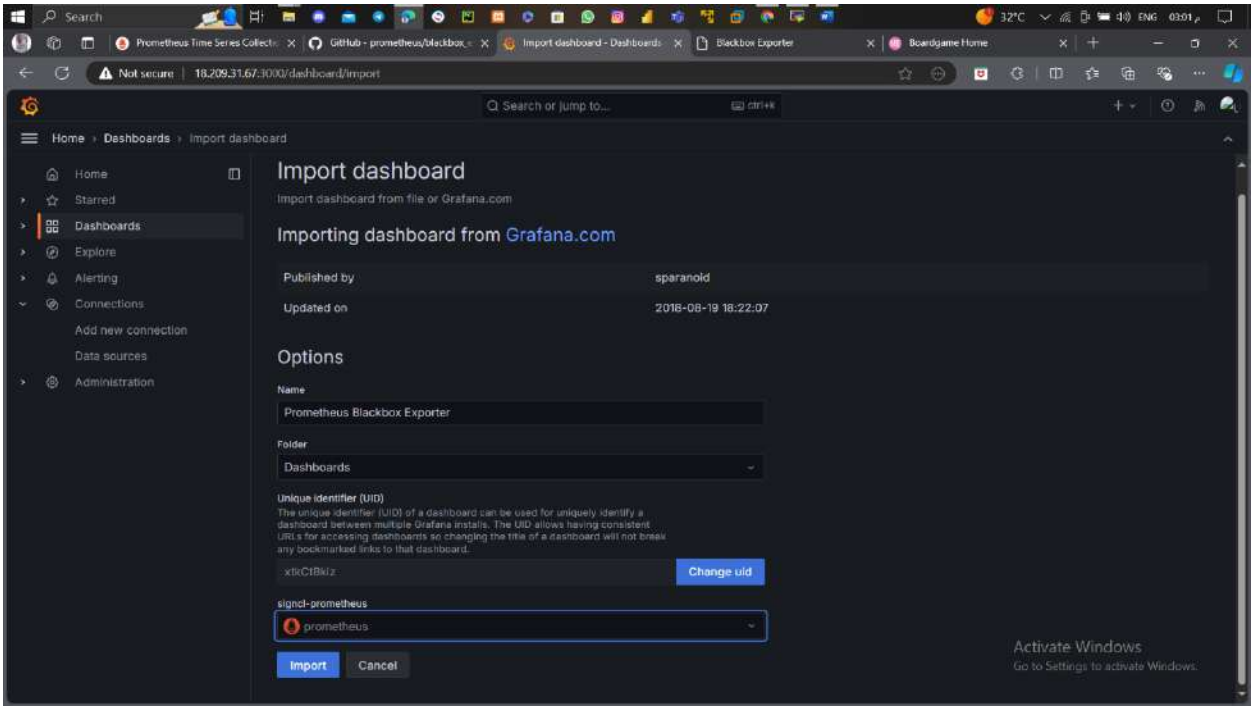
Restart Prometheus

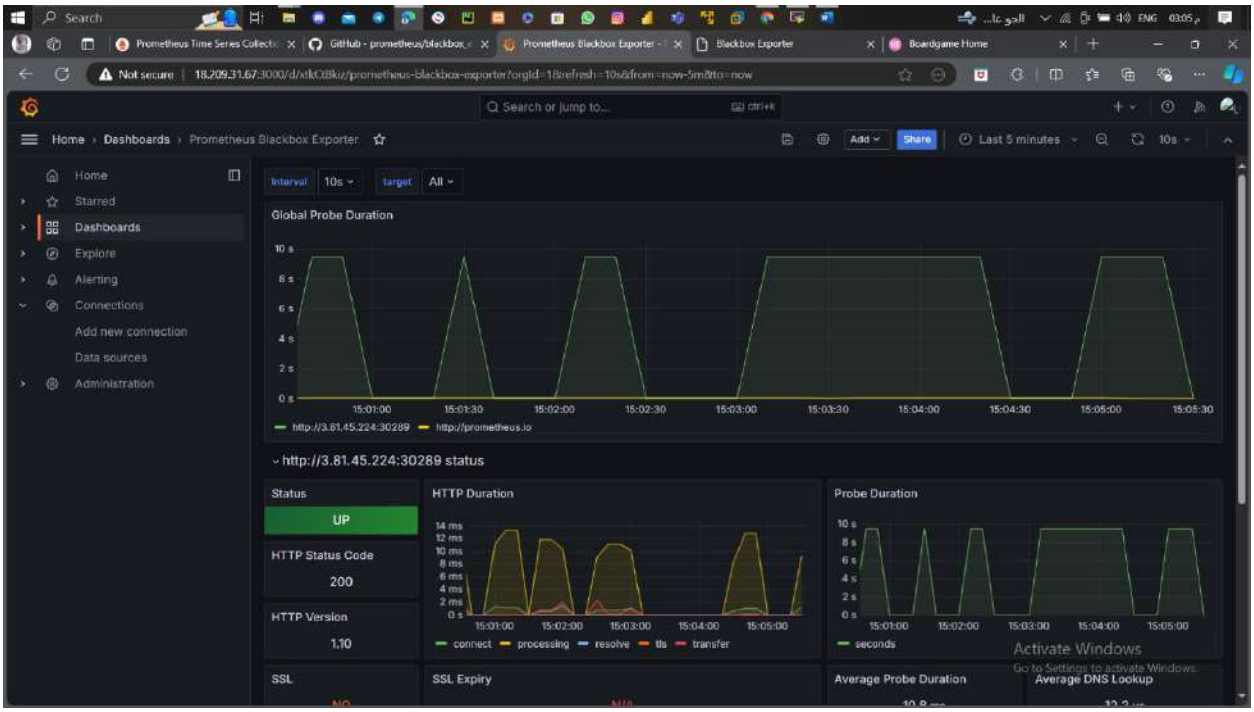
- pgrep prometheus
- kill ##the id
- ./prometheus &

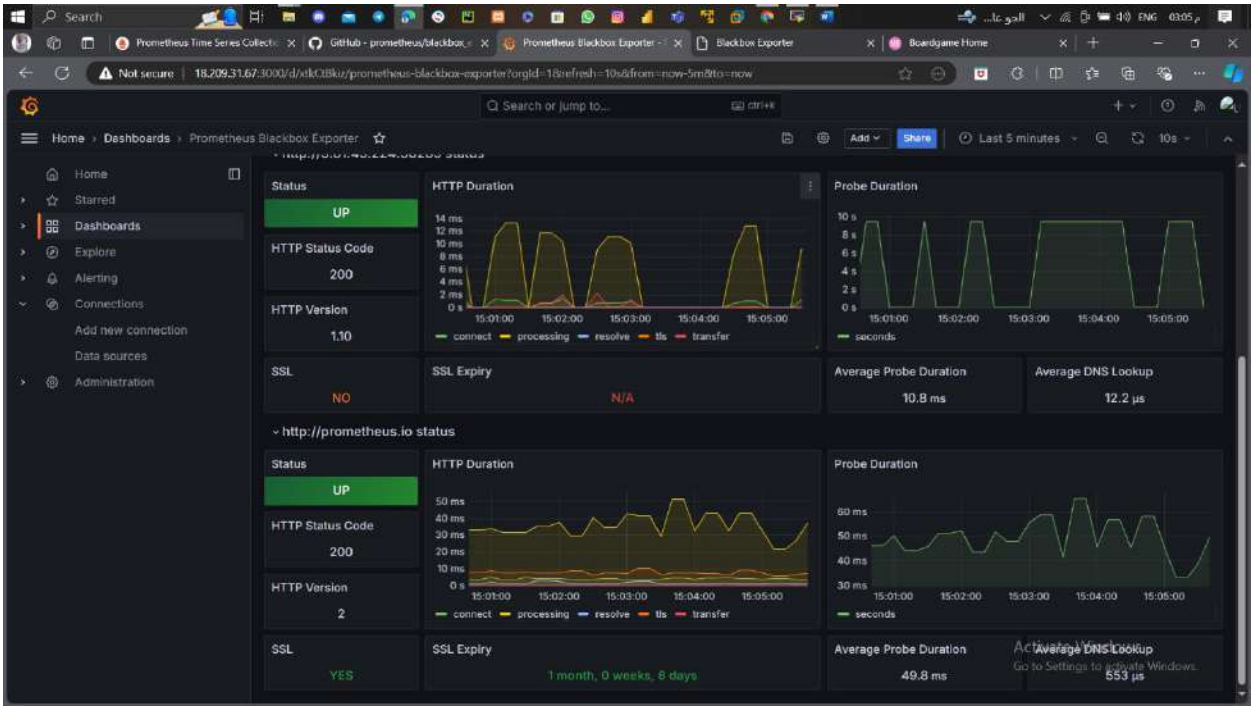




Import the dashboard


Import dashboard
Import dashboard from file or Grafana.com
Importing dashboard from Grafana.com
Published by
sparanoid
Updated on
2018-08-19 18:22:07
Options
Name
Prometheus Blackbox Exporter
Folder
Dashboards
Unique identifier (UID)
The unique identifier (UID) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The UID allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.
xtkCtBkiz
Change uid
signcl-prometheus
prometheus
Import
Cancel
Activate Windows
Go to Settings to activate Windows.


Global Probe Duration
http://3.81.45.224:30289 status
Status
UP
HTTP Status Code
200
HTTP Version
1.10
SSL
HTTP Duration
Probe Duration
SSL Expiry
Average Probe Duration
Average DNS Lookup
Activate Windows
Go to Settings to activate Windows.


Home › Dashboards › Prometheus Blackbox Exporter
Add
Share
Last 5 minutes
10s
Home
Starred
Dashboards
Explore
Alerting
Connections
Add new connection
Data sources
Administration
Status
UP
HTTP Status Code
200
HTTP Version
1.10
HTTP Duration
connect
processing
resolve
tls
transfer
Probe Duration
seconds
SSL
NO
SSL Expiry
N/A
Average Probe Duration
10.8 ms
Average DNS Lookup
12.2 µs
http://prometheus.io status
Status
UP
HTTP Status Code
200
HTTP Version
2
HTTP Duration
connect
processing
resolve
tls
transfer
Probe Duration
seconds
SSL
YES
SSL Expiry
1 month, 0 weeks, 8 days
Average Probe Duration
49.8 ms
Average DNS Lookup
553 µs