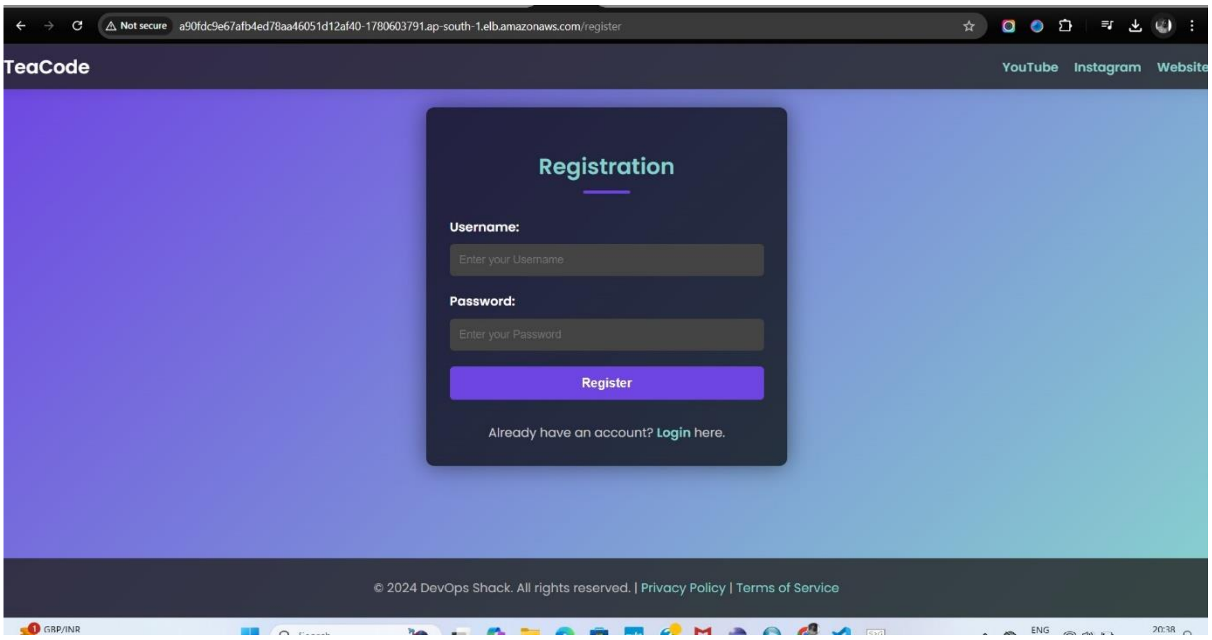
# Production Level CICD Pipeline Project | CICD DevOps Project


Not secure
a90fdc9e67afb4ed78aa46051d12af40-1780603791.ap-south-1.elb.amazonaws.com/register
TeaCode
YouTube
Instagram
Website
Registration
Username:
Enter your Username
Password:
Enter your Password
Register
Already have an account? Login here.
© 2024 DevOps Shack. All rights reserved. | Privacy Policy | Terms of Service

Divya Satpute

@TeaCode1122

Divya satpute

**What we are doing ????**

1. Setup Repo
2. Set-Up Required Servers[Jenkins, SonarQube, Nexus, Monitoring Tools
3. Configure Tools
4. Create The Pipelines & Create EKS Clusters
5. Trigger The Pipeline To Deploy the Application
6. Assign a Custom domain to the deployed application
7. Monitor The Application

**Prerequisites**

## Step 1

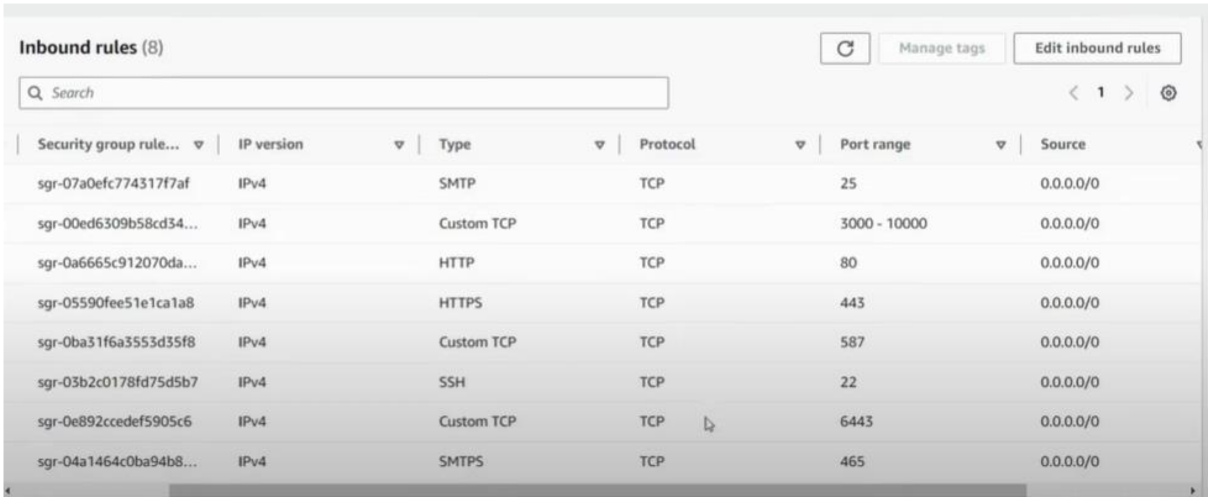
### Setting up EKS Cluster Using Terraform

AWS Console launch server for terraform

t2 medium

40 storage

open this Ports inbound rule on security group



| Security group rule... | IP version | Type | Protocol | Port range | Source |
|---|---|---|---|---|---|
| sgr-07a0efc774317f7af | IPv4 | SMTP | TCP | 25 | 0.0.0.0/0 |
| sgr-00ed6309b58cd34... | IPv4 | Custom TCP | TCP | 3000 - 10000 | 0.0.0.0/0 |
| sgr-0a6665c912070da... | IPv4 | HTTP | TCP | 80 | 0.0.0.0/0 |
| sgr-05590fee51e1ca1a8 | IPv4 | HTTPS | TCP | 443 | 0.0.0.0/0 |
| sgr-0ba31f6a3553d35f8 | IPv4 | Custom TCP | TCP | 587 | 0.0.0.0/0 |
| sgr-03b2c0178fd75d5b7 | IPv4 | SSH | TCP | 22 | 0.0.0.0/0 |
| sgr-0e892ccedef5905c6 | IPv4 | Custom TCP | TCP | 6443 | 0.0.0.0/0 |
| sgr-04a1464c0ba94b8... | IPv4 | SMTPS | TCP | 465 | 0.0.0.0/0 |

update repo

**$sudo apt update -y**

Install AWS CLI

**$curl "https://awscli.amazonaws.com/awscli-exe-linux-x86_64.zip" -o "awscliv2.zip"**

**sudo apt install unzip**

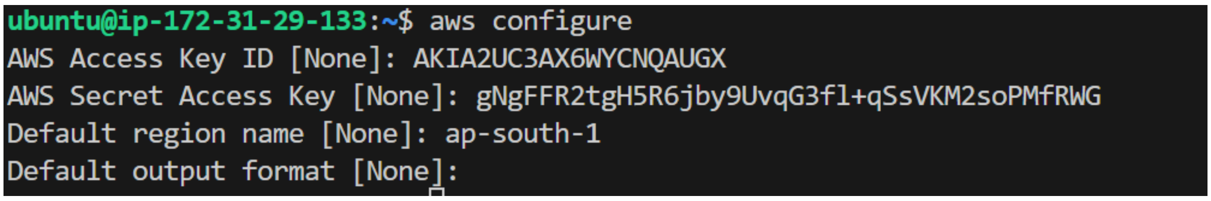
**unzip awscliv2.zip**

**sudo ./aws/install**

**@TeaCode1122**

AWS Configure Provide Access key and Secret key on Aws Console

aws configure



```
ubuntu@ip-172-31-29-133:~$ aws configure
AWS Access Key ID [None]: AKIA2UC3AX6WYCNQAUGX
AWS Secret Access Key [None]: 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
Default region name [None]: ap-south-1
Default output format [None]:
```

Install Kubectl

**$curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl**

**$chmod +x ./kubectl**

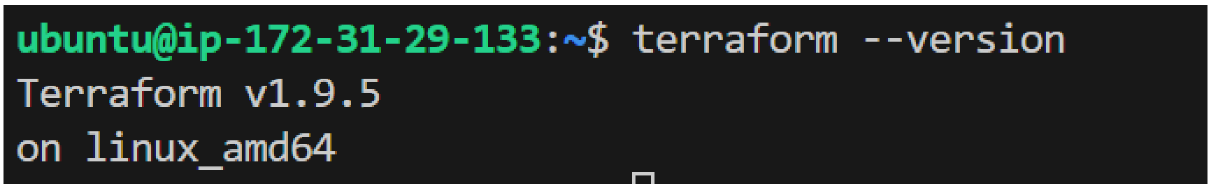
**$sudo mv ./kubectl /usr/local/bin**

**$kubectl version --short --client**

Installation of Terafform

**$sudo snap install terraform --classic**

terraform --version



```
ubuntu@ip-172-31-29-133:~$ terraform --version
Terraform v1.9.5
on linux_amd64
```

clone the Repo for EKS Terraform Script

**$git clone https://github.com/divyasatpute/FullStack-Blogging-App.git**

change directory

**$cd FullStack-Blogging-App/**

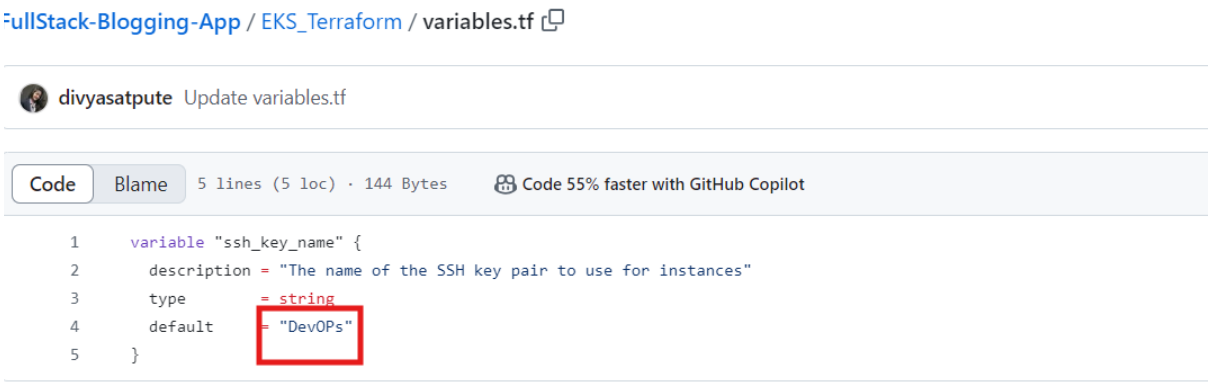
change directory

**$cd EKS_Terraform/**

In Variables.tf file you just need to change Your key name

AND in main.tf file you just need to change region and availability zone as per your requirement

Divya satpute



FullStack-Blogging-App / EKS_Terraform / variables.tf

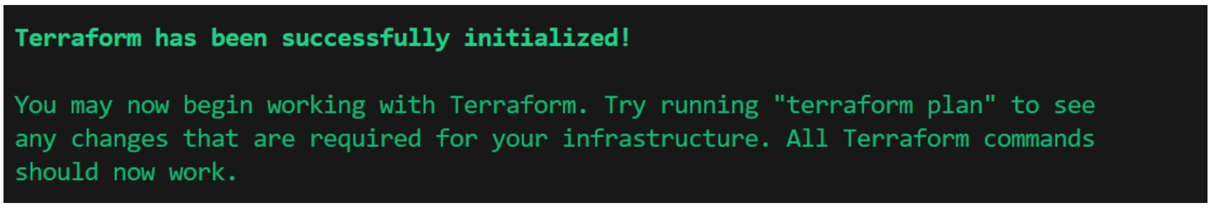
divyasatpute Update variables.tf

Code Blame 5 lines (5 loc) · 144 Bytes Code 55% faster with GitHub Copilot

```
variable "ssh_key_name" {
  description = "The name of the SSH key pair to use for instances"
  type        = string
  default     = "DevOPs"
}
```

Now terraform initialization

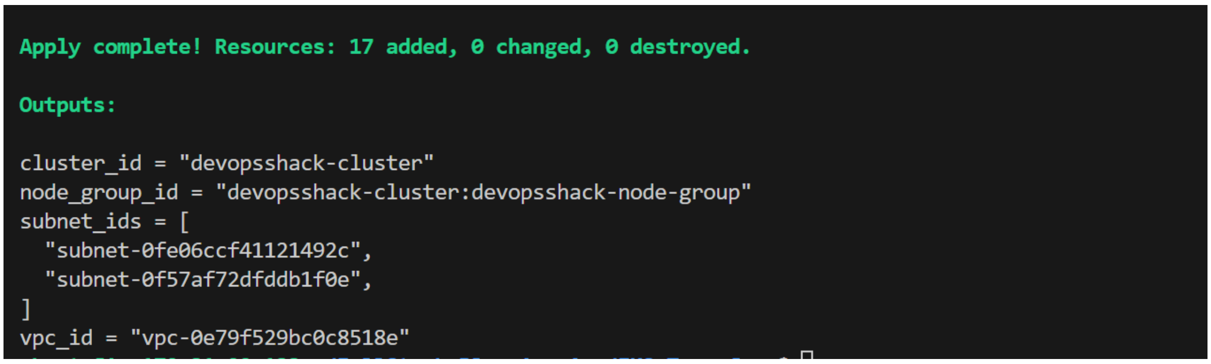
**$terraform init**



```
Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
```

**$terraform plan**

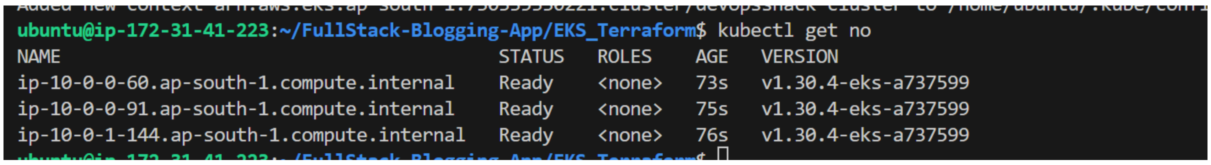
**$terraform apply --auto-approve**



```
Apply complete! Resources: 17 added, 0 changed, 0 destroyed.

Outputs:

cluster_id = "devopsshack-cluster"
node_group_id = "devopsshack-cluster:devopsshack-node-group"
subnet_ids = [
  "subnet-0fe06ccf41121492c",
  "subnet-0f57af72dfddb1f0e",
]
vpc_id = "vpc-0e79f529bc0c8518e"
```

In Order to communicate with aws eks cluster we need to update our kubeconfig file

**$aws eks --region ap-south-1 update-kubeconfig --name devopsshack-cluster**



```
ubuntu@ip-172-31-41-223:~/FullStack-Blogging-App/EKS_Terraform$ kubectl get no
NAME                                         STATUS   ROLES    AGE   VERSION
ip-10-0-0-60.ap-south-1.compute.internal     Ready    <none>   73s   v1.30.4-eks-a737599
ip-10-0-0-91.ap-south-1.compute.internal     Ready    <none>   75s   v1.30.4-eks-a737599
ip-10-0-1-144.ap-south-1.compute.internal    Ready    <none>   76s   v1.30.4-eks-a737599
```

**Step 2**

40 GB Storage

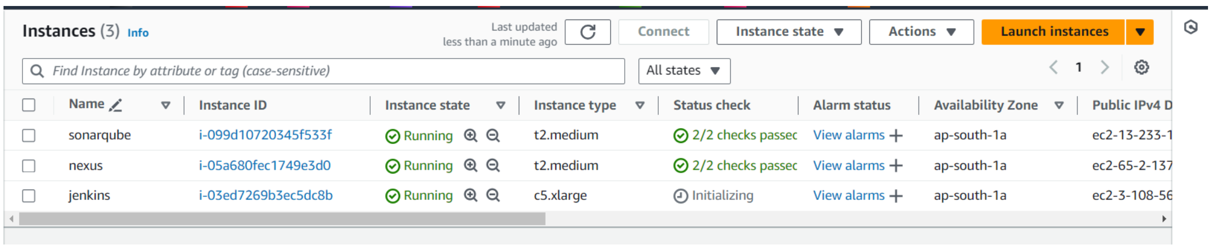
Launch 1 EC2 Machine one for Jenkins

t2.large

40 GB storage

**@TeaCode1122**

Divya satpute


Instances (3) Info

Last updated less than a minute ago

Find Instance by attribute or tag (case-sensitive)

All states

| Name | Instance ID | Instance state | Instance type | Status check | Alarm status | Availability Zone | Public IPv4 D |
|---|---|---|---|---|---|---|---|
| sonarqube | i-099d10720345f533f | Running | t2.medium | 2/2 checks passec | View alarms + | ap-south-1a | ec2-13-233-1 |
| nexus | i-05a680fec1749e3d0 | Running | t2.medium | 2/2 checks passec | View alarms + | ap-south-1a | ec2-65-2-137 |
| jenkins | i-03ed7269b3ec5dc8b | Running | c5.xlarge | Initializing | View alarms + | ap-south-1a | ec2-3-108-56 |

Connect them with using gitbash

## Installation Jenkins

### step 1

**Install java (latest stable version)**

```
$sudo apt install openjdk-17-jre-headless -y
```

**Install Jenkins**

```
$vi 1.sh
```

Paste the all command in 1.sh file

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

**Change the permission**

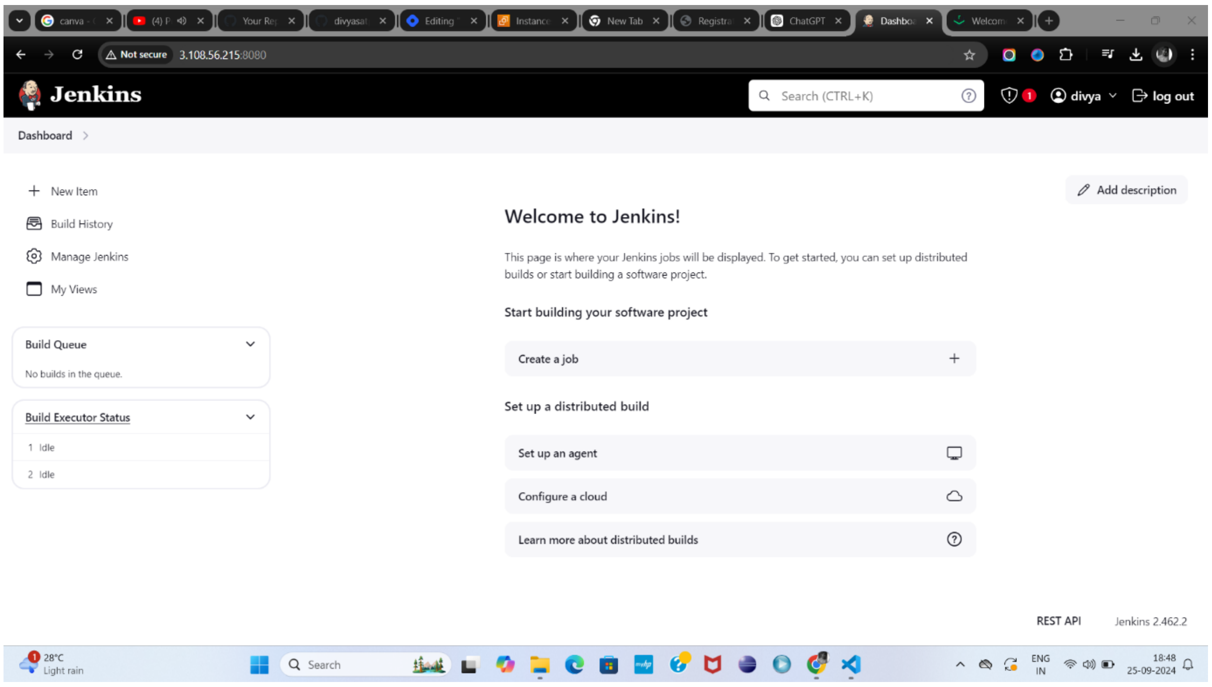
```
$sudo chmod +x 1.sh
```

Run the file

```
$./1.sh
```

**@TeaCode1122**

Divya satpute



No builds in the queue.

Welcome to Jenkins!

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

Start building your software project

Set up a distributed build

## Installation docker on Jenkins machine

Install docker

**$sudo apt install docker.io -y**

change permission

**$sudo chmod 666 /var/run/docker.sock**

## Installation Trivy on Jenkins machine

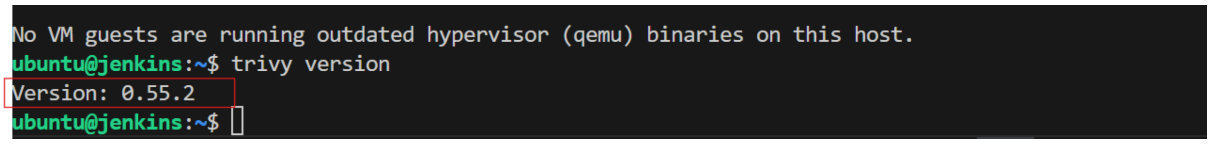
**$sudo apt-get install wget apt-transport-https gnupg lsb-release**

**wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | sudo apt-key add -**

**echo deb https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main | sudo tee -a /etc/apt/sources.list.d/trivy.list**

**sudo apt-get update**

**sudo apt-get install trivy -y**



```
No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@jenkins:~$ trivy version
Version: 0.55.2
ubuntu@jenkins:~$
```

## Installation kubectl on Jenkins machine

**$curl -o kubectl https://amazon-eks.s3.us-west-2.amazonaws.com/1.19.6/2021-01-05/bin/linux/amd64/kubectl**

**chmod +x ./kubectl**

**@TeaCode1122**

**sudo mv ./kubectl /usr/local/bin**

**kubectl version --short --client**

## Installation Nexus as a docker container

update machine

**$sudo apt update -y**

Install docker

**$sudo apt install docker.io -y**

Create container

**$sudo docker run -d -p 8081:8081 sonatype/nexus3**

Access your Nexus On Browser **http://PUBLIC_IP:8081/**

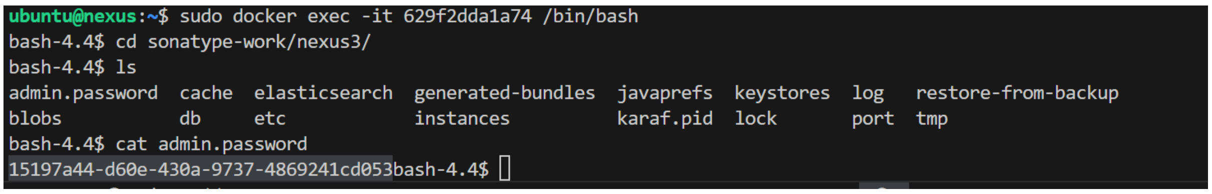
our Nexus up and running but password is stored inside the container so for that we need to go inside the container

**$sudo docker exec -it 629f2dda1a74 /bin/bash**

**$cd sonatype-work/nexus3/**

**$cat admin.password**

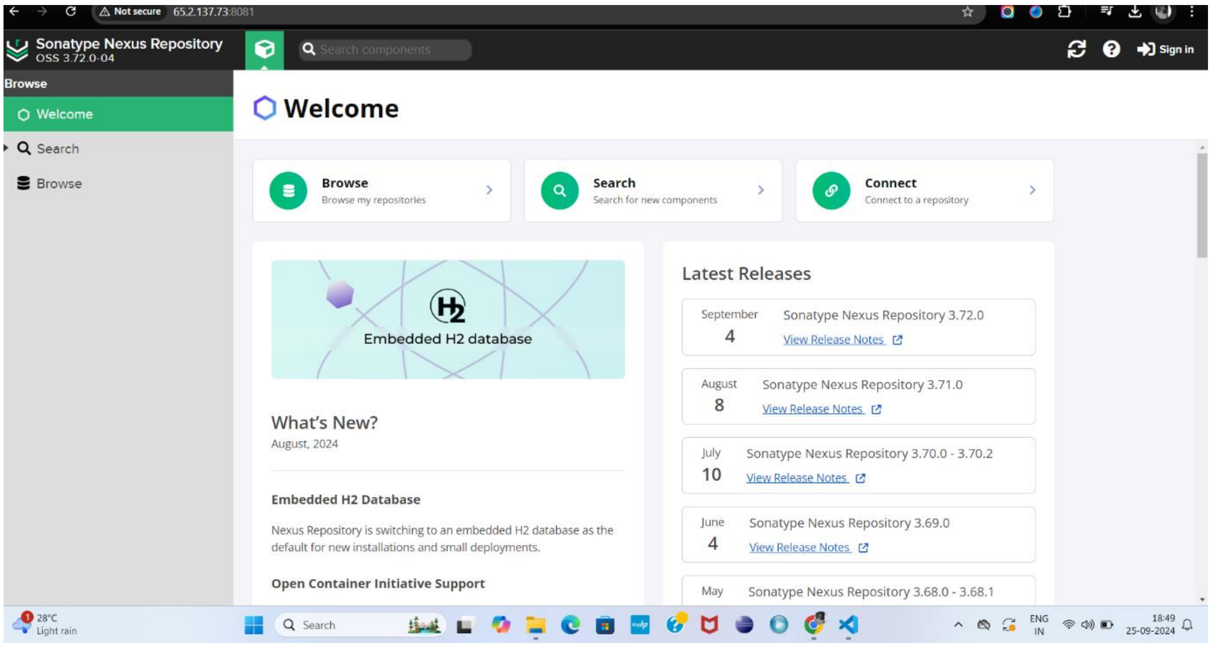
here you can got password



```
ubuntu@nexus:~$ sudo docker exec -it 629f2dda1a74 /bin/bash
bash-4.4$ cd sonatype-work/nexus3/
bash-4.4$ ls
admin.password  cache  elasticsearch  generated-bundles  javaprefs  keystores  log   restore-from-backup
blobs           db     etc            instances          karaf.pid  lock       port  tmp
bash-4.4$ cat admin.password
15197a44-d60e-430a-9737-4869241cd053bash-4.4$
```

Now You Can See Our Nexus also working fine and able to sign in

Divya satpute



Welcome

Browse — Browse my repositories

Search — Search for new components

Connect — Connect to a repository

Embedded H2 database

What's New?
August, 2024

Embedded H2 Database

Nexus Repository is switching to an embedded H2 database as the default for new installations and small deployments.

Open Container Initiative Support

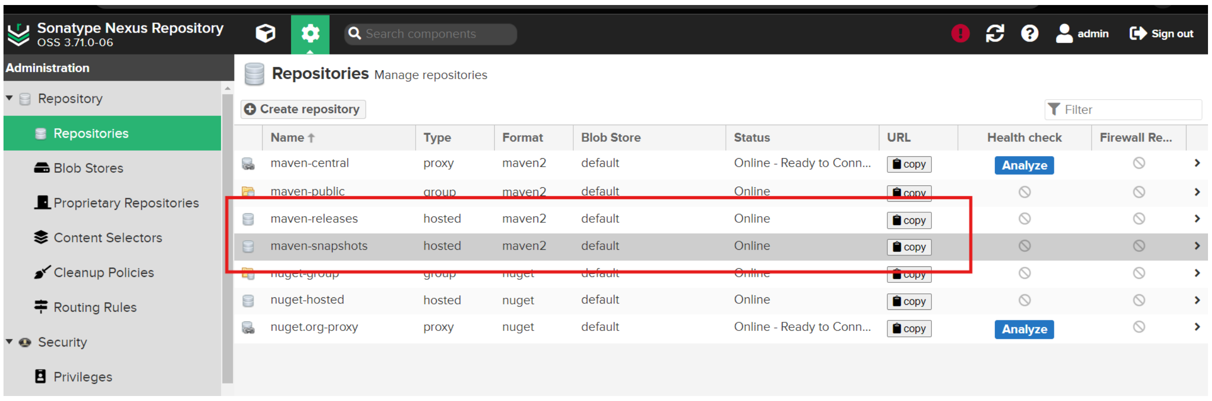
Latest Releases

| | |
|---|---|
| September 4 | Sonatype Nexus Repository 3.72.0 View Release Notes |
| August 8 | Sonatype Nexus Repository 3.71.0 View Release Notes |
| July 10 | Sonatype Nexus Repository 3.70.0 - 3.70.2 View Release Notes |
| June 4 | Sonatype Nexus Repository 3.69.0 View Release Notes |
| May | Sonatype Nexus Repository 3.68.0 - 3.68.1 |

# Nexus Configuration

Go to nexus dashboard --> click on settings ---> click on repositories

copy the Maven-releases URL and Maven snapshot URL and paste it on POX.XML file



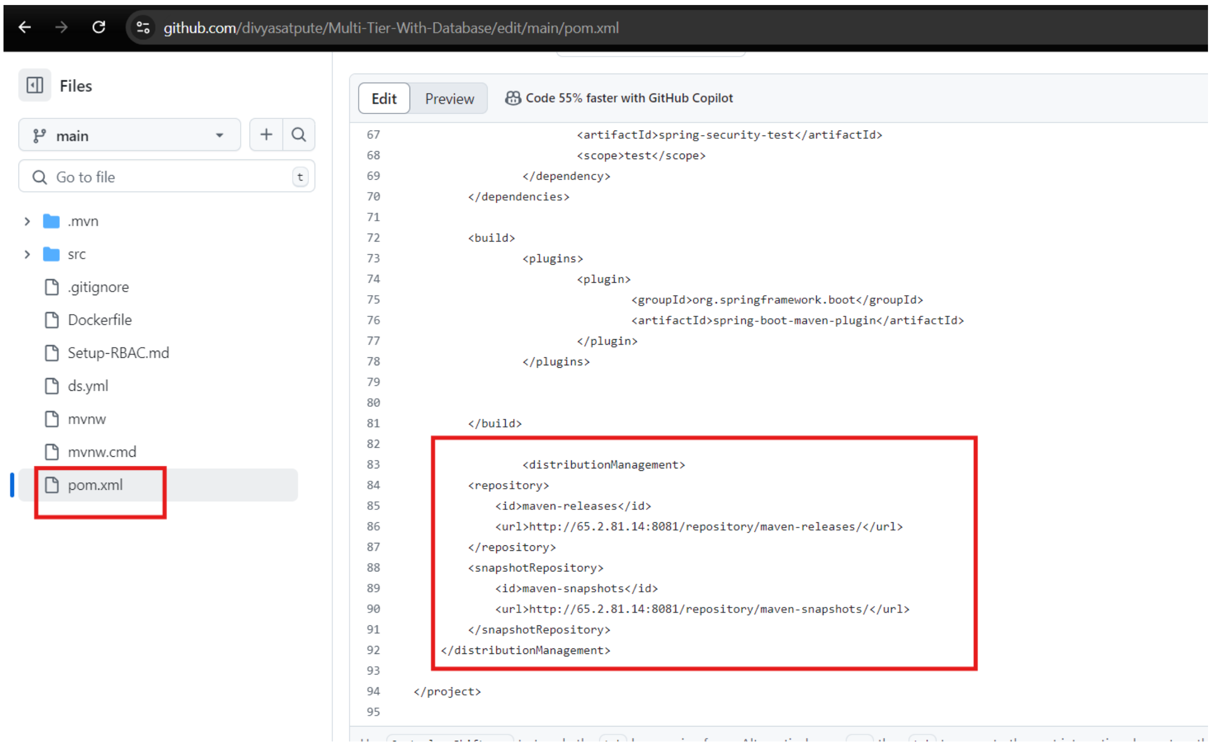
Repositories Manage repositories

Create repository

| Name | Type | Format | Blob Store | Status | URL | Health check | Firewall Re... |
|---|---|---|---|---|---|---|---|
| maven-central | proxy | maven2 | default | Online - Ready to Conn... | copy | Analyze | |
| maven-public | group | maven2 | default | Online | copy | | |
| maven-releases | hosted | maven2 | default | Online | copy | | |
| maven-snapshots | hosted | maven2 | default | Online | copy | | |
| nuget-group | group | nuget | default | Online | copy | | |
| nuget-hosted | hosted | nuget | default | Online | copy | | |
| nuget.org-proxy | proxy | nuget | default | Online - Ready to Conn... | copy | Analyze | |

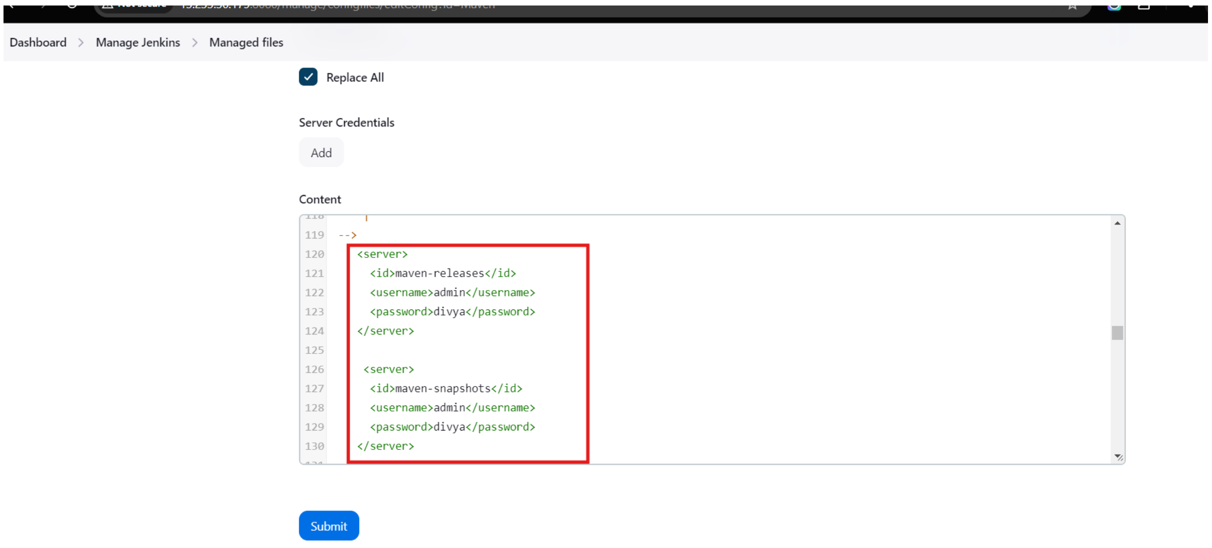
@TeaCode1122

Divya satpute


github.com/divyasatpute/Multi-Tier-With-Database/edit/main/pom.xml
Files
main
Go to file
.mvn
src
.gitignore
Dockerfile
Setup-RBAC.md
ds.yml
mvnw
mvnw.cmd
pom.xml
Edit
Preview
Code 55% faster with GitHub Copilot

```
				<artifactId>spring-security-test</artifactId>
				<scope>test</scope>
			</dependency>
		</dependencies>

		<build>
			<plugins>
				<plugin>
					<groupId>org.springframework.boot</groupId>
					<artifactId>spring-boot-maven-plugin</artifactId>
				</plugin>
			</plugins>


		</build>

			<distributionManagement>
		<repository>
		    <id>maven-releases</id>
		    <url>http://65.2.81.14:8081/repository/maven-releases/</url>
		</repository>
		<snapshotRepository>
		    <id>maven-snapshots</id>
		    <url>http://65.2.81.14:8081/repository/maven-snapshots/</url>
		</snapshotRepository>
	</distributionManagement>

</project>
```

for credentials go to Jenkins Dashboard --->click on manage Jenkins---> Managed files---> click on Add new Config--->Global Maven settings.xml--->provide id "anything"---> click on next



```
-->
 <server>
   <id>maven-releases</id>
   <username>admin</username>
   <password>divya</password>
 </server>

  <server>
   <id>maven-snapshots</id>
   <username>admin</username>
   <password>divya</password>
 </server>
```

## Installation SonarQube as a docker container

update machine

**$sudo apt update -y**

Install docker

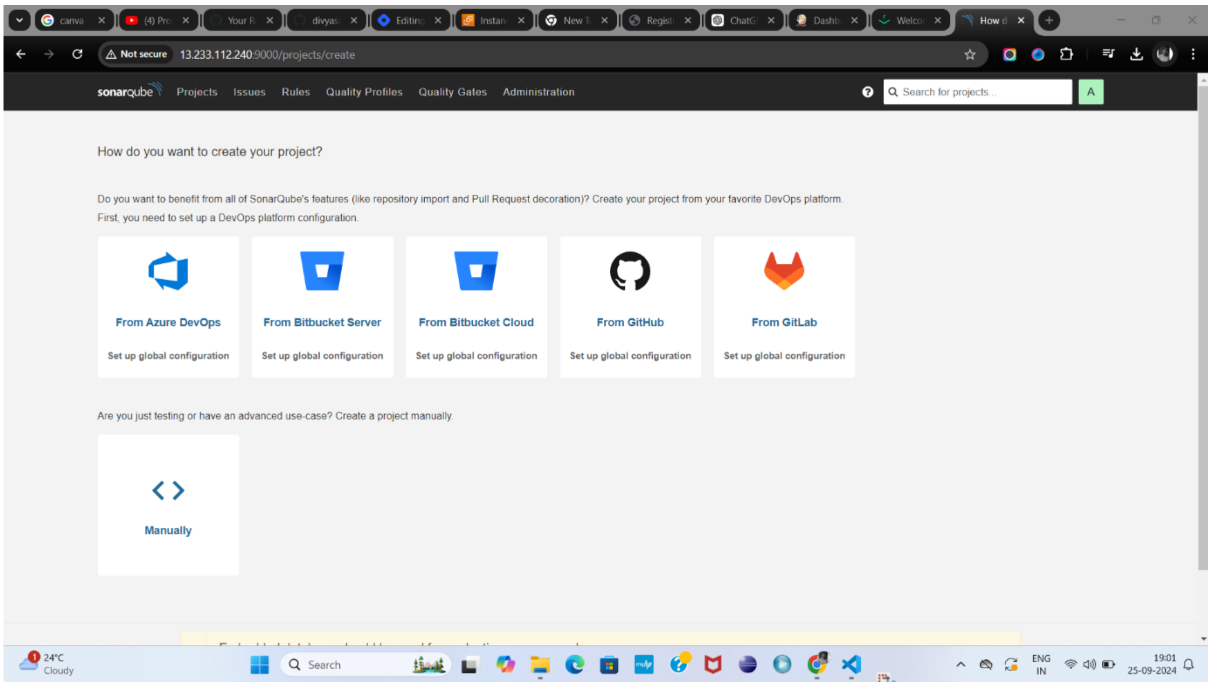
**$sudo apt install docker.io -y**

Create container

**@TeaCode1122**

Divya satpute

**$sudo docker run -it -p 9000:9000 sonarqube:lts-community**



How do you want to create your project?

Do you want to benefit from all of SonarQube's features (like repository import and Pull Request decoration)? Create your project from your favorite DevOps platform.
First, you need to set up a DevOps platform configuration.

From Azure DevOps — Set up global configuration
From Bitbucket Server — Set up global configuration
From Bitbucket Cloud — Set up global configuration
From GitHub — Set up global configuration
From GitLab — Set up global configuration

Are you just testing or have an advanced use-case? Create a project manually.

Manually

# Configuration on Jenkins

## Installation Plugins

**SonarQube Scanner**

**Config File Provider**

**Maven Integration**

**Pipeline Maven Integration**

**Kubernetes**

**Kubernetes Client API**

**Kubernetes Credentials**

**Kubernetes CLI**

**Kubernetes Credentials Provider**

**Docker Pipeline**

**Docker Commons**

**Docker**

**Eclipse Temurin installer**

**Pipeline: Stage View**

Divya satpute

## Configuration System

Sonar Scanner

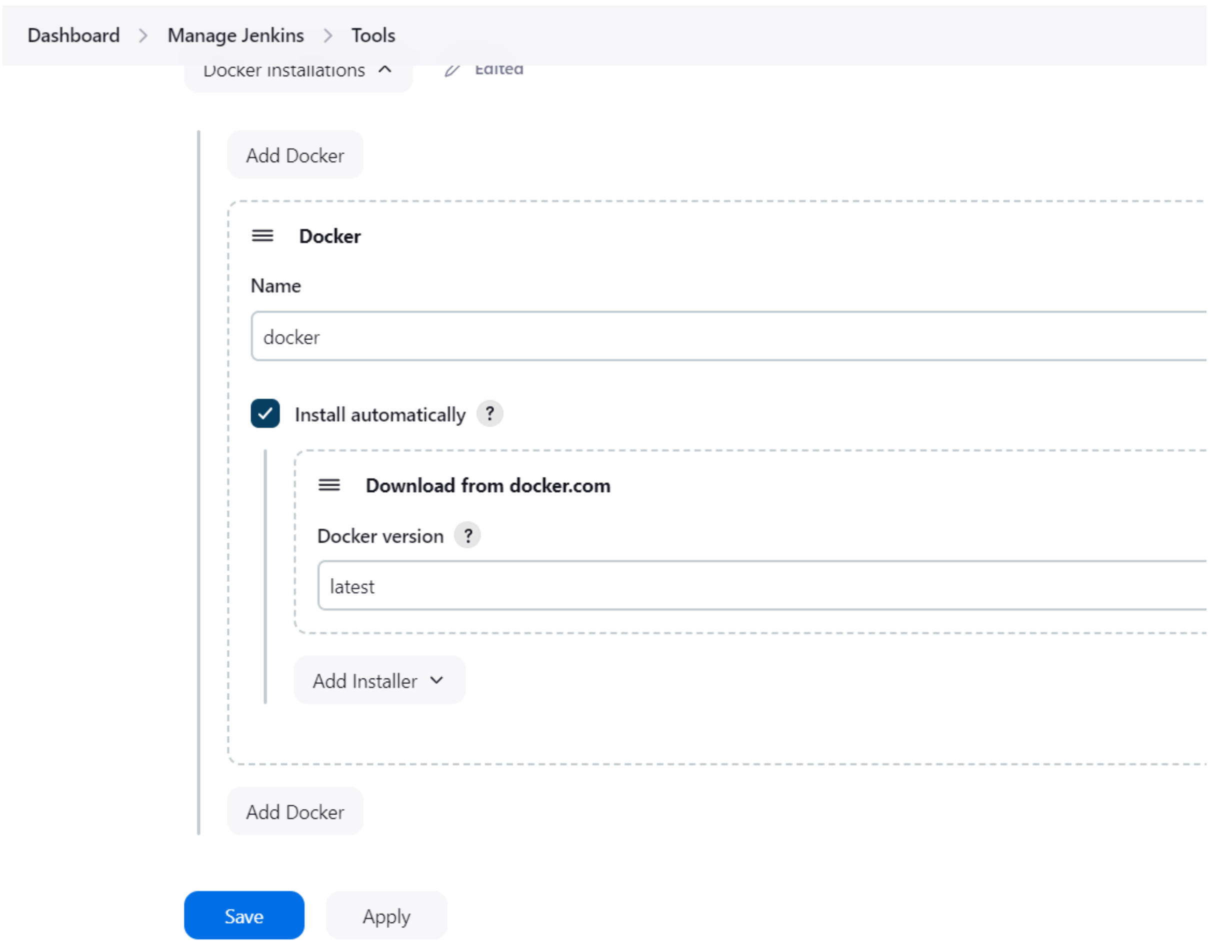
## Configuration tools

Go to Manage jenkins ----> tools

add SonarQube Scanner

add Maven

Add Docker


Dashboard > Manage Jenkins > Tools

Docker installations

Add Docker

Docker

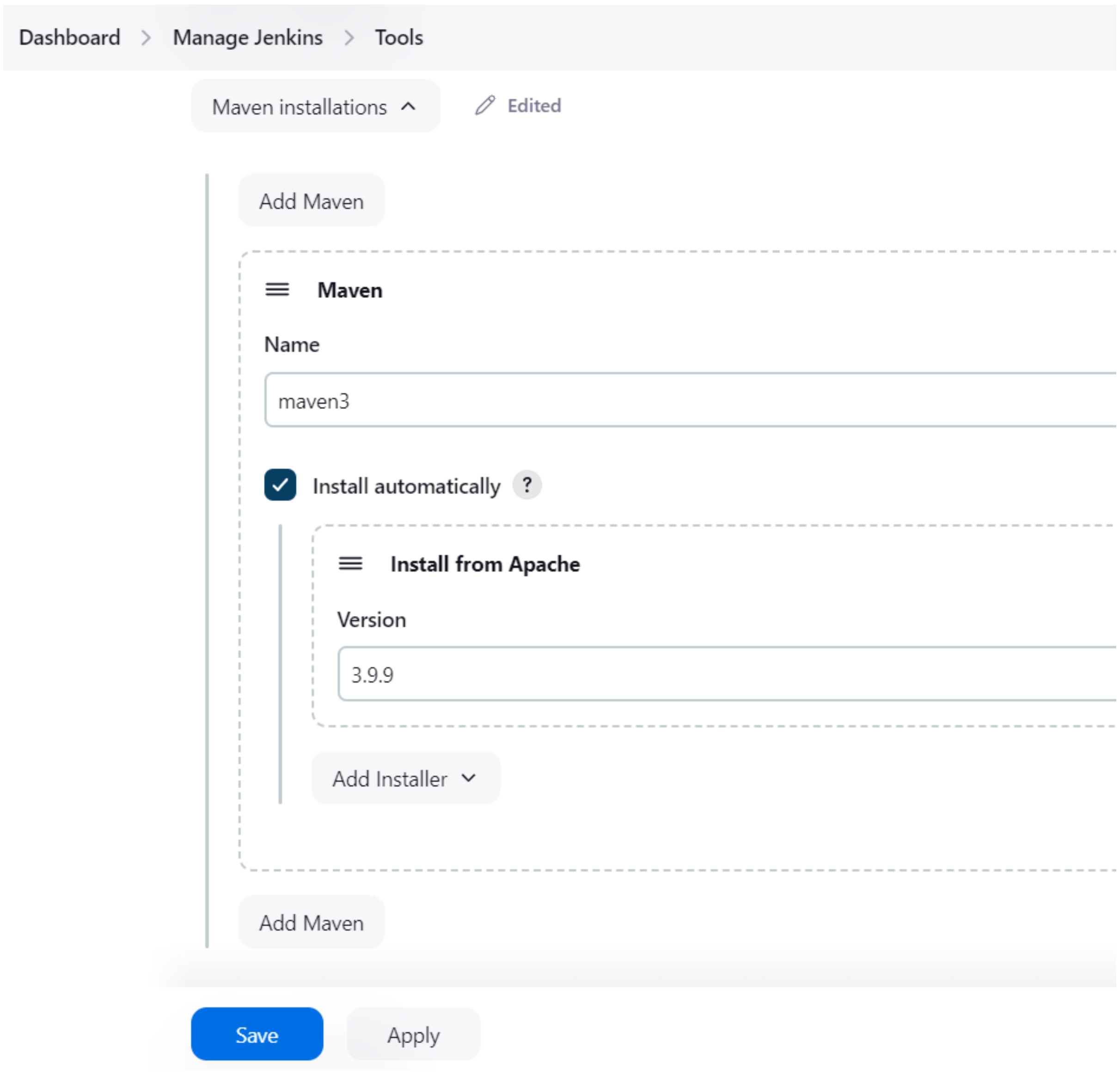
Name: docker

[✓] Install automatically

Download from docker.com

Docker version: latest

Divya satpute



Maven

Name

maven3

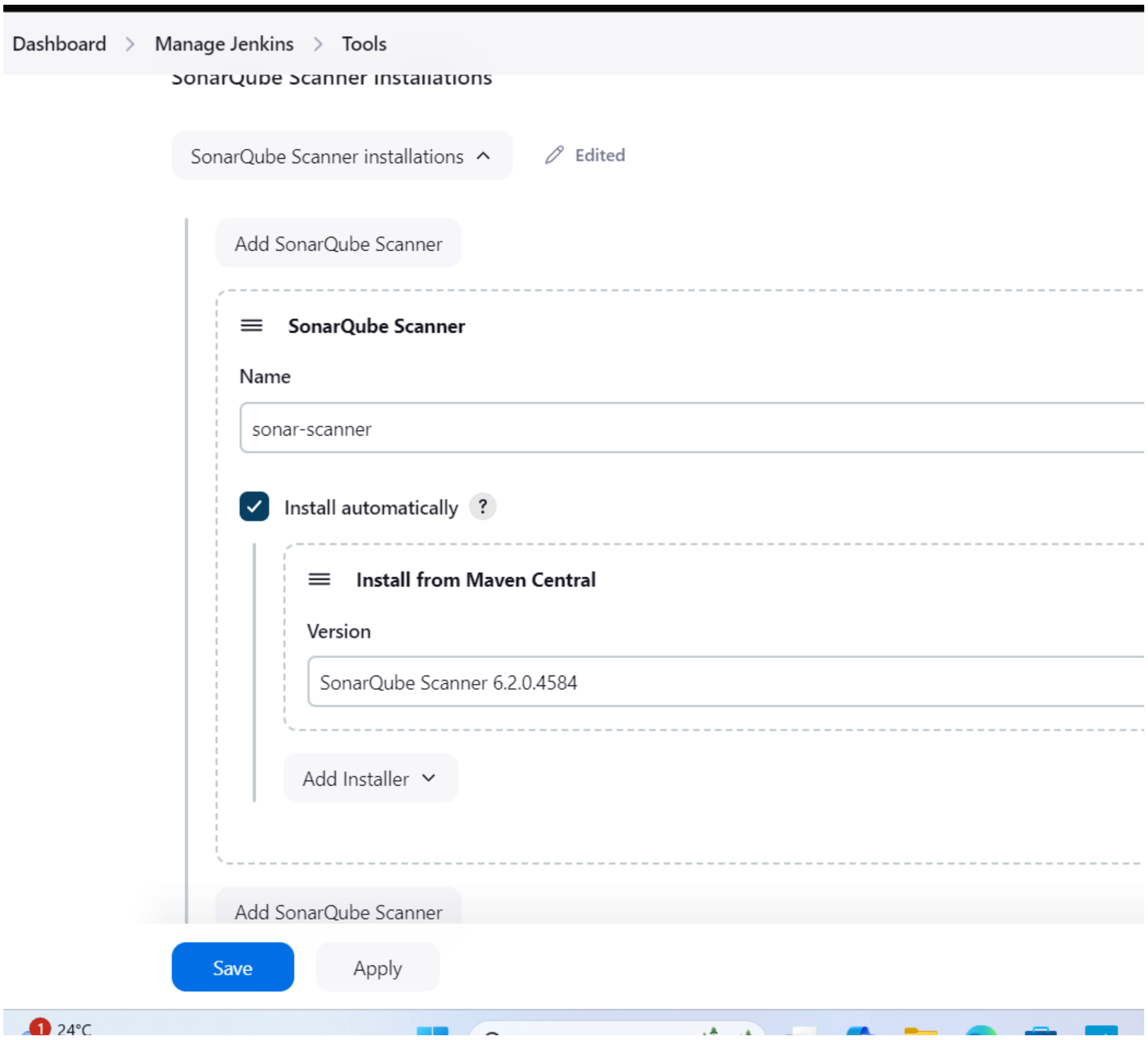
✓ Install automatically

Install from Apache

Version

3.9.9

@TeaCode1122


Dashboard > Manage Jenkins > Tools

SonarQube Scanner installations

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

**SonarQube Scanner**

Name

sonar-scanner

✓ Install automatically ?

**Install from Maven Central**

Version

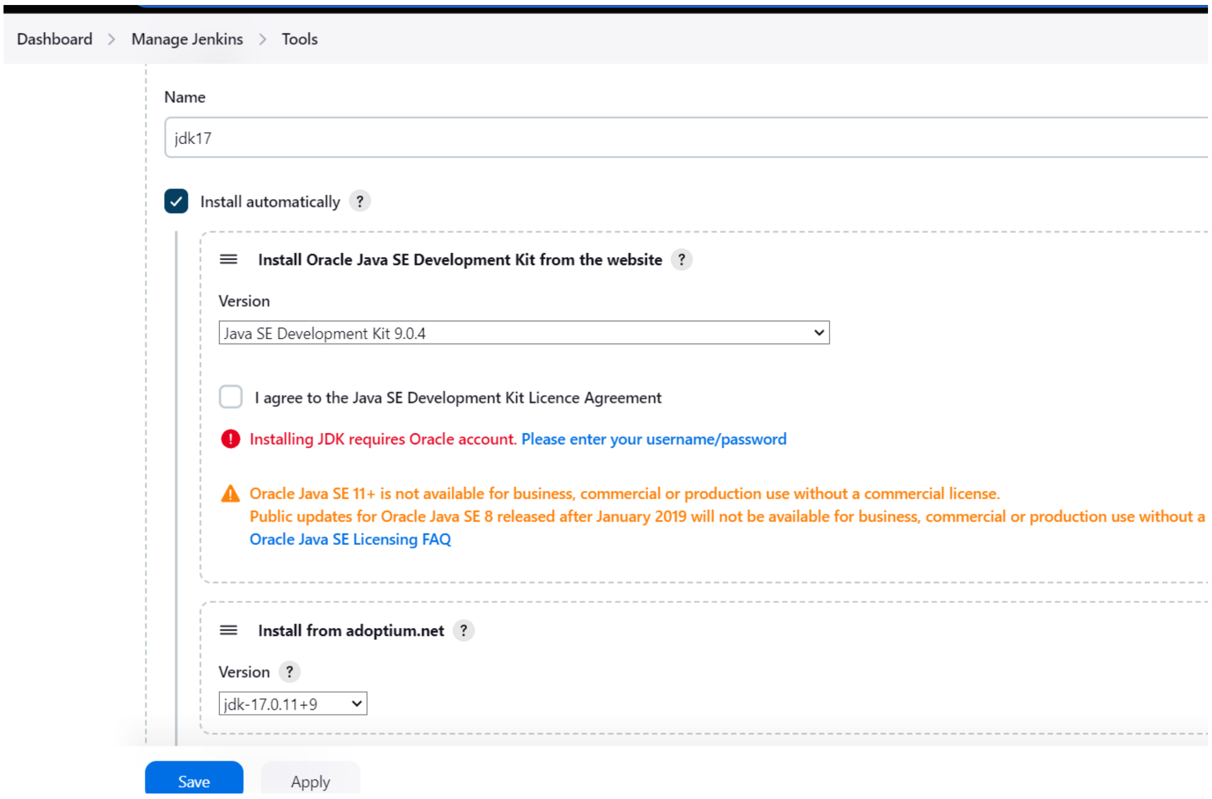
SonarQube Scanner 6.2.0.4584

Add Installer

Add SonarQube Scanner

Save Apply

24°C


Dashboard > Manage Jenkins > Tools

Name

jdk17

☑ Install automatically ?

Install Oracle Java SE Development Kit from the website ?

Version

Java SE Development Kit 9.0.4

☐ I agree to the Java SE Development Kit Licence Agreement

Installing JDK requires Oracle account. Please enter your username/password

Oracle Java SE 11+ is not available for business, commercial or production use without a commercial license.
Public updates for Oracle Java SE 8 released after January 2019 will not be available for business, commercial or production use without a
Oracle Java SE Licensing FAQ

Install from adoptium.net ?

Version ?

jdk-17.0.11+9

# Deployment

**Create Service Account, Role & Assign that role, And create a secret for Service Account and generate a Token**

**Create namespace**

**$kubectl create ns webapps**

**Creating Service Account**

**$vi svc.yml**

```
apiVersion: v1

kind: ServiceAccount

metadata:

  name: jenkins

  namespace: webapps
```

**kubectl apply -f svc.yml**

**Create Role**

**$vi role.yml**

```
apiVersion: rbac.authorization.k8s.io/v1

kind: Role
```

```
metadata:
  name: app-role
  namespace: webapps
rules:
  - apiGroups:
        - ""
        - apps
        - autoscaling
        - batch
        - extensions
        - policy
        - rbac.authorization.k8s.io
    resources:
      - pods
      - componentstatuses
      - configmaps
      - daemonsets
      - deployments
      - events
      - endpoints
      - horizontalpodautoscalers
      - ingress
      - jobs
      - limitranges
      - namespaces
      - nodes
      - secrets
      - pods
      - persistentvolumes
      - persistentvolumeclaims
      - resourcequotas
      - replicasets
      - replicationcontrollers
      - serviceaccounts
      - services
    verbs: ["get", "list", "watch", "create", "update", "patch", "delete"]
$kubectl apply -f role.yml
```

**Bind the role to service account**

```
 $vi bind.yml
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: app-rolebinding
  namespace: webapps
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: app-role
subjects:
- namespace: webapps
  kind: ServiceAccount
  name: jenkins
kubectl apply -f bind.yml
```

**for token**

```
vi jen.secret.yml
apiVersion: v1
kind: Secret
type: kubernetes.io/service-account-token
metadata:
  name: mysecretname
  annotations:
    kubernetes.io/service-account.name: jenkins
```

Divya satpute

**$kubectl apply -f jen.secret.yml -n webapps**

**for docker secret**

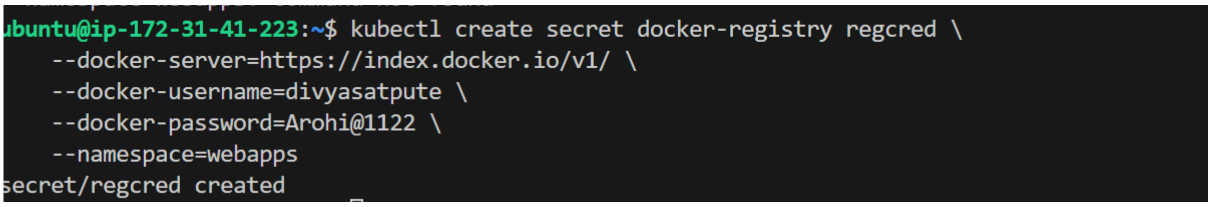
**kubectl create secret docker-registry regcred \**

   **--docker-server=https://index.docker.io/v1/ \**

   **--docker-username=divyasatpute \**

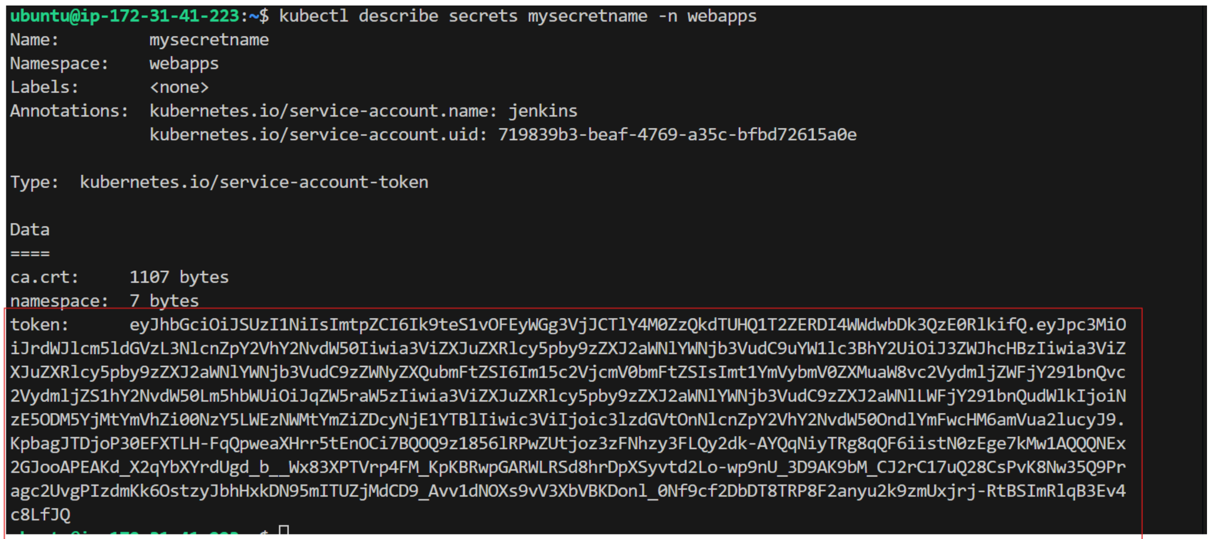
   **--docker-password=123654 \**

   **--namespace=webapps**



```
ubuntu@ip-172-31-41-223:~$ kubectl create secret docker-registry regcred \
    --docker-server=https://index.docker.io/v1/ \
    --docker-username=divyasatpute \
    --docker-password=Arohi@1122 \
    --namespace=webapps
secret/regcred created
```

**$kubectl describe secrets mysecretname -n webapps**



```
ubuntu@ip-172-31-41-223:~$ kubectl describe secrets mysecretname -n webapps
Name:         mysecretname
Namespace:    webapps
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: jenkins
              kubernetes.io/service-account.uid: 719839b3-beaf-4769-a35c-bfbd72615a0e

Type:  kubernetes.io/service-account-token

Data
====
ca.crt:     1107 bytes
namespace:  7 bytes
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6Ik9teS1vOFEyWGg3VjJCTlY4M0ZzQkdTUHQ1T2ZERDI4WWdwbDk3QzE0RlkifQ.eyJpc3MiO
iJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJ3ZWJhcHBzIiwia3ViZ
XJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6Im15c2VjcmV0bmFtZSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc
2VydmljZS1hY2NvdW50Lm5hbWUiOiJqZW5raW5zIiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiN
zE5ODM5YjMtYmVhZi00NzY5LWEzNWMtYmZiZDcyNjE1YTBlIiwic3ViIjoic3lzdGVtOnNlcnZpY2VhY2NvdW50OndlYmFwcHM6amVua2lucyJ9.
KpbagJTDjoP30EFXTLH-FqQpweaXHrr5tEnOCi7BQOQ9z1856lRPwZUtjoz3zFNhzy3FLQy2dk-AYQqNiyTRg8qQF6iistN0zEge7kMw1AQQQNEx
2GJooAPEAKd_X2qYbXYrdUgd_b__Wx83XPTVrp4FM_KpKBRwpGARWLRSd8hrDpXSyvtd2Lo-wp9nU_3D9AK9bM_CJ2rC17uQ28CsPvK8Nw35Q9Pr
agc2UvgPIzdmKk6OstzyJbhHxkDN95mITUZjMdCD9_Avv1dNOXs9vV3XbVBKDonl_0Nf9cf2DbDT8TRP8F2anyu2k9zmUxjrj-RtBSImRlqB3Ev4
c8LfJQ
```

# Pipeline

```
pipeline {
  agent any

  tools {
    jdk 'jdk17'
    maven 'maven3'
  }
  environment{
```

**@TeaCode1122**

```
        SCANNER_HOME= tool 'sonar-scanner'
    }

    stages {
        stage('Git Checkout') {
            steps {
                git branch: 'main', credentialsId: 'git-cred', url: 'https://github.com/divyasatpute/full-stack-
app-project.git'
            }
        }
        stage('Compile') {
            steps {
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test'
            }
        }
        stage('Trivy fs scan') {
            steps {
                sh 'trivy fs --format table -o fs.html .'
            }
        }
        stage('SonarQube Analysis') {
            steps {
                withSonarQubeEnv('sonar-server') {
                sh '''$SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Blogging-app -
Dsonar.projectKey=Blogging-app \
                 -Dsonar.java.binaries=target'''
                }
            }
        }
        stage('Build') {
            steps {
                sh 'mvn clean package'
            }
        }
        stage('Publish Artifacts') {
            steps {
                withMaven(globalMavenSettingsConfig: 'maven-settings', jdk: 'jdk17', maven: 'maven3',
mavenSettingsConfig: '', traceability: true) {
                 sh 'mvn deploy'
                }
            }
        }
         stage('Docker Build & Tag ') {
            steps {
                script{
                   withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {

                sh 'docker build -t divyasatpute/bloggingapp:latest . --no-cache '
                   }
                }
            }
        }
        stage('Trivy image scan') {
            steps {
                sh 'trivy image --format table -o image.html divyasatpute/bloggingapp:latest'
            }
        }
        stage('Docker Push') {
            steps {
                script{
                   withDockerRegistry(credentialsId: 'docker-cred', toolName: 'docker') {


                sh 'docker push divyasatpute/bloggingapp:latest'
                   }
                }
            }
        }
        stage('k8-Deploy') {
            steps {
                withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '',
credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl:
'https://0D7DFCF662ECC24043497267C6A5BDEB.gr7.ap-south-1.eks.amazonaws.com') {
                sh 'kubectl apply -f deployment-service.yml'
                sleep 20
                }
            }
        }
         stage('verify the Deployment') {
            steps {
                withKubeConfig(caCertificate: '', clusterName: 'devopsshack-cluster', contextName: '',
credentialsId: 'k8-cred', namespace: 'webapps', restrictKubeConfigAccess: false, serverUrl:
'https://0D7DFCF662ECC24043497267C6A5BDEB.gr7.ap-south-1.eks.amazonaws.com') {
                sh 'kubectl get pods'
                sh 'kubectl get svc'
                }
            }
        }


      }
}
```

Divya satpute

# Installation Monitaring tool

**$sudo apt update -y**

**$wget https://github.com/prometheus/prometheus/releases/download/v3.0.0-beta.0/prometheus-3.0.0-beta.0.linux-amd64.tar.gz**

**$tar -xvf prometheus-3.0.0-beta.0.linux-amd64.tar.gz**

**$wget https://github.com/prometheus/blackbox_exporter/releases/download/v0.25.0/blackbox_exporter-0.25.0.linux-amd64.tar.gz**

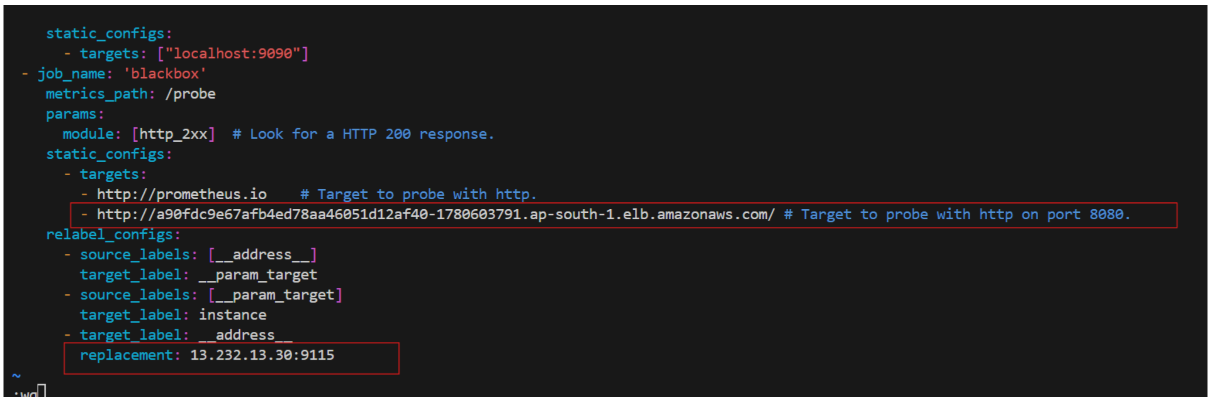
**$tar -xvf blackbox_exporter-0.25.0.linux-amd64.tar.gz**

**$cd prometheus-3.0.0-beta.0.linux-amd64**

**$./prometheus &**

**$cd prometheus-3.0.0-beta.0.linux-amd64**

**$vi prometheus.yml**



```
    static_configs:
      - targets: ["localhost:9090"]
  - job_name: 'blackbox'
    metrics_path: /probe
    params:
      module: [http_2xx]  # Look for a HTTP 200 response.
    static_configs:
      - targets:
        - http://prometheus.io    # Target to probe with http.
        - http://a90fdc9e67afb4ed78aa46051d12af40-1780603791.ap-south-1.elb.amazonaws.com/ # Target to probe with http on port 8080.
    relabel_configs:
      - source_labels: [__address__]
        target_label: __param_target
      - source_labels: [__param_target]
        target_label: instance
      - target_label: __address__
        replacement: 13.232.13.30:9115
~
:wq
```

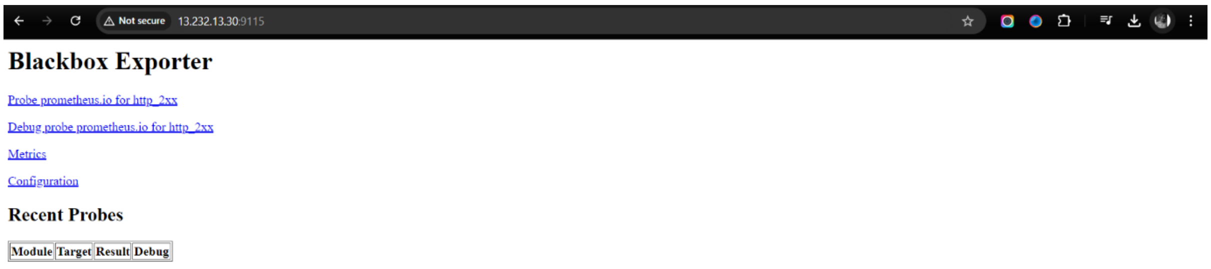
access prometheus http://13.232.13.30:9090

for blackbox exporter

**$cd blackbox_exporter-0.25.0.linux-amd64**

**$./blackbox_exporter &**

access blackbox http://13.232.13.30:9090



Not secure 13.232.13.30:9115

Blackbox Exporter

Probe prometheus.io for http_2xx

Debug probe prometheus.io for http_2xx

Metrics

Configuration

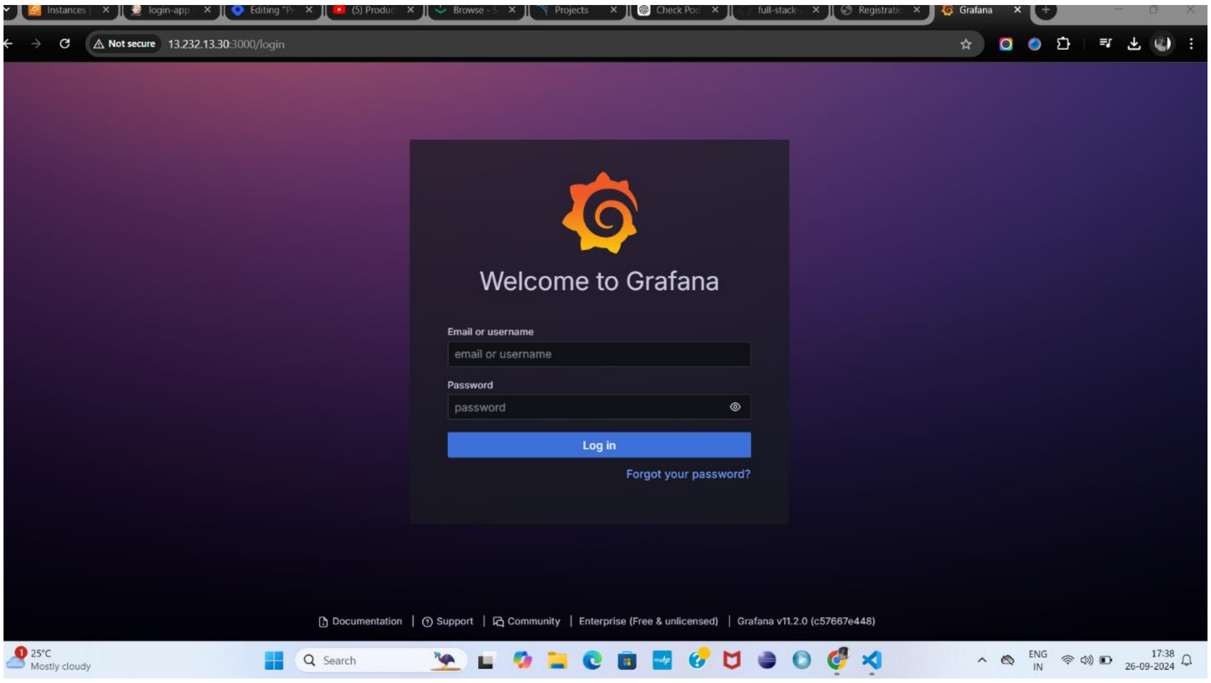
Recent Probes

| Module | Target | Result | Debug |
|---|---|---|---|

**@TeaCode1122**

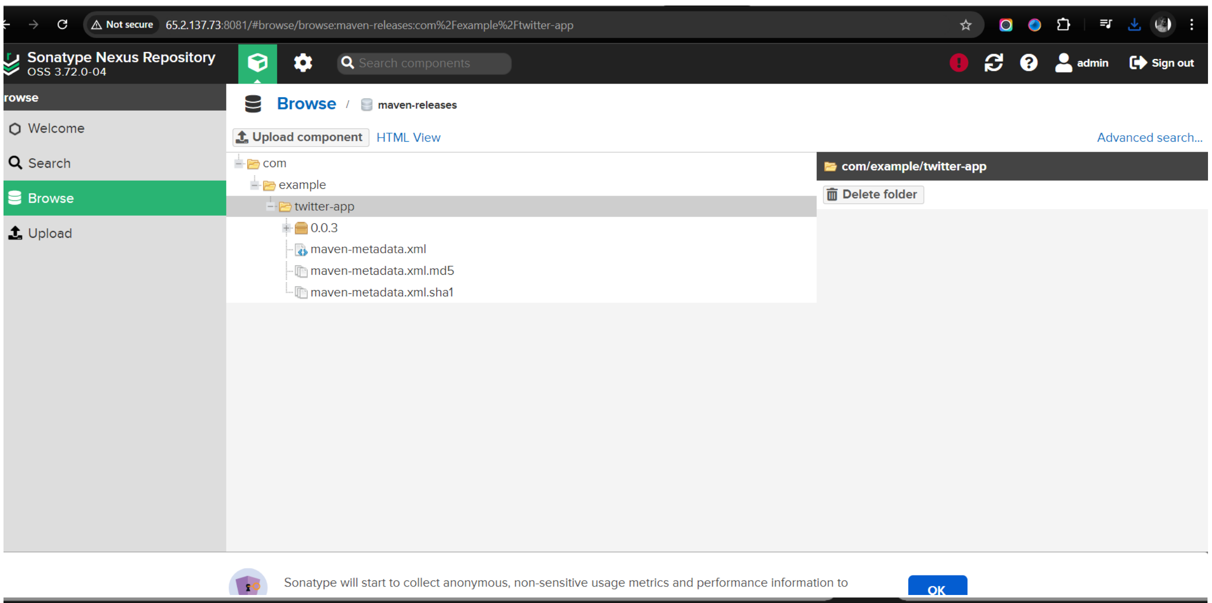
## For Grafana

```
$sudo apt-get install -y adduser libfontconfig1 musl

$wget https://dl.grafana.com/enterprise/release/grafana-enterprise_11.2.0_amd64.deb

$sudo dpkg -i grafana-enterprise_11.2.0_amd64.deb

$sudo /bin/systemctl start grafana-server
```


Not secure 13.232.13.30:3000/login
Welcome to Grafana
Email or username
email or username
Password
password
Log in
Forgot your password?
Documentation | Support | Community | Enterprise (Free & unlicensed) | Grafana v11.2.0 (c57667e448)

Divya satpute

## Test Results



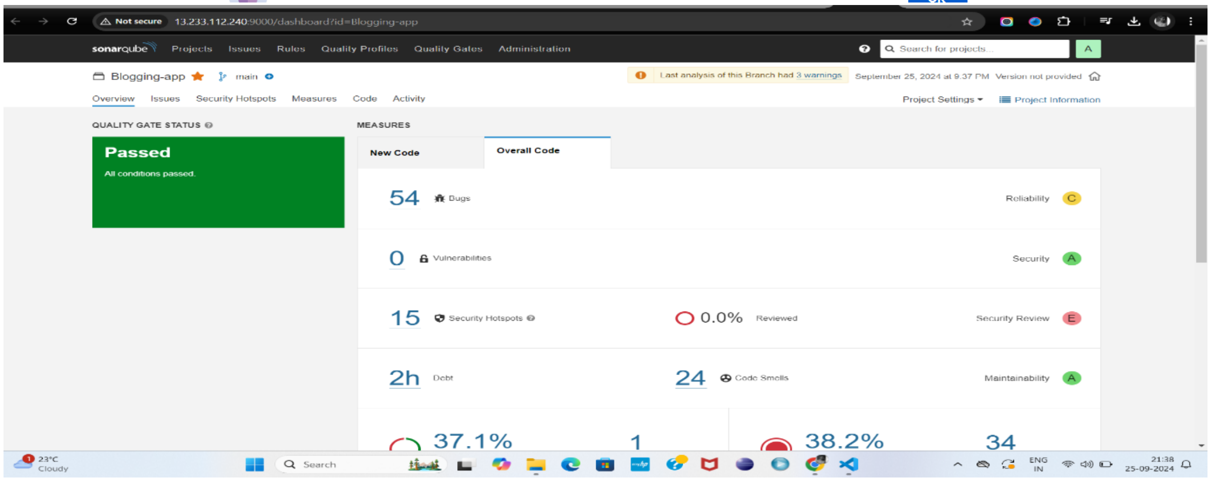
65.2.137.73:8081/#browse/browse:maven-releases:com%2Fexample%2Ftwitter-app

Browse / maven-releases

```
com
  example
    twitter-app
      0.0.3
      maven-metadata.xml
      maven-metadata.xml.md5
      maven-metadata.xml.sha1
```

com/example/twitter-app

Sonatype will start to collect anonymous, non-sensitive usage metrics and performance information to



13.233.112.240:9000/dashboard?id=Blogging-app

Blogging-app main

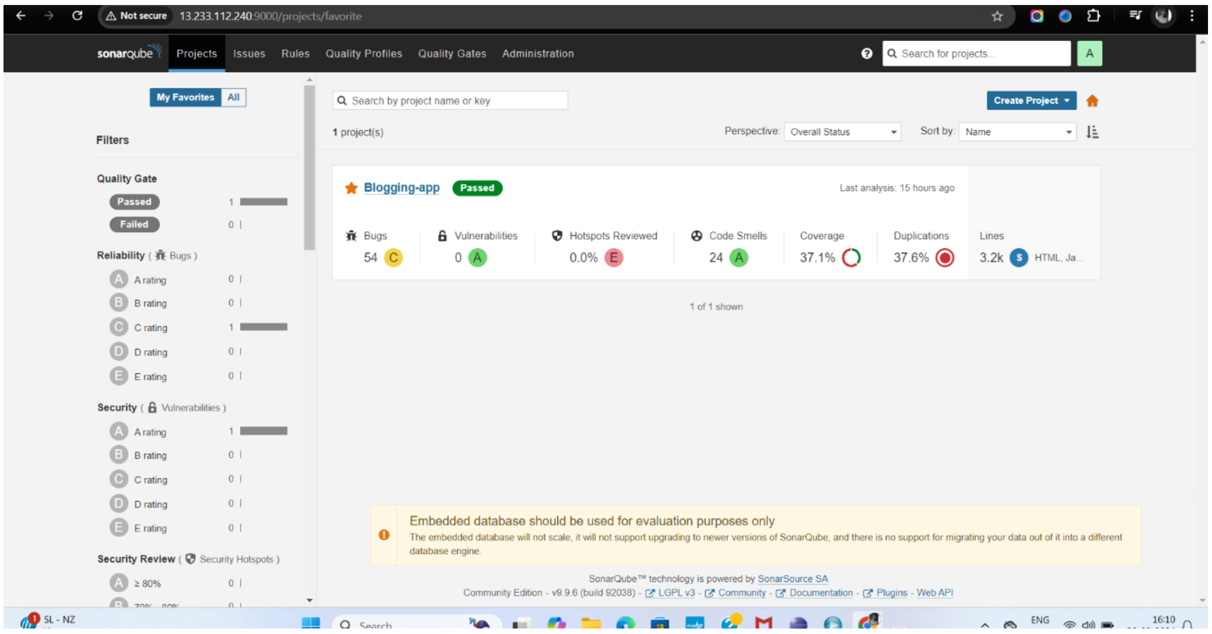
Last analysis of this Branch had 3 warnings September 25, 2024 at 9:37 PM Version not provided

QUALITY GATE STATUS

Passed
All conditions passed.

MEASURES

| | | |
|---|---|---|
| 54 Bugs | | Reliability C |
| 0 Vulnerabilities | | Security A |
| 15 Security Hotspots | 0.0% Reviewed | Security Review E |
| 2h Debt | 24 Code Smells | Maintainability A |
| 37.1% | 1 | 38.2% 34 |



13.233.112.240:9000/projects/favorite

Search by project name or key

1 project(s) Perspective: Overall Status Sort by: Name

Filters

Quality Gate

| | | |
|---|---|---|
| Passed | 1 | |
| Failed | 0 | |

Reliability ( Bugs )

| | | |
|---|---|---|
| A | A rating | 0 |
| B | B rating | 0 |
| C | C rating | 1 |
| D | D rating | 0 |
| E | E rating | 0 |

Security ( Vulnerabilities )

| | | |
|---|---|---|
| A | A rating | 1 |
| B | B rating | 0 |
| C | C rating | 0 |
| D | D rating | 0 |
| E | E rating | 0 |

Security Review ( Security Hotspots )

| | | |
|---|---|---|
| A | ≥ 80% | 0 |
| B | 70% - 80% | 0 |

Blogging-app Passed Last analysis: 15 hours ago

| Bugs | Vulnerabilities | Hotspots Reviewed | Code Smells | Coverage | Duplications | Lines |
|---|---|---|---|---|---|---|
| 54 C | 0 A | 0.0% E | 24 A | 37.1% | 37.6% | 3.2k S HTML, Ja... |

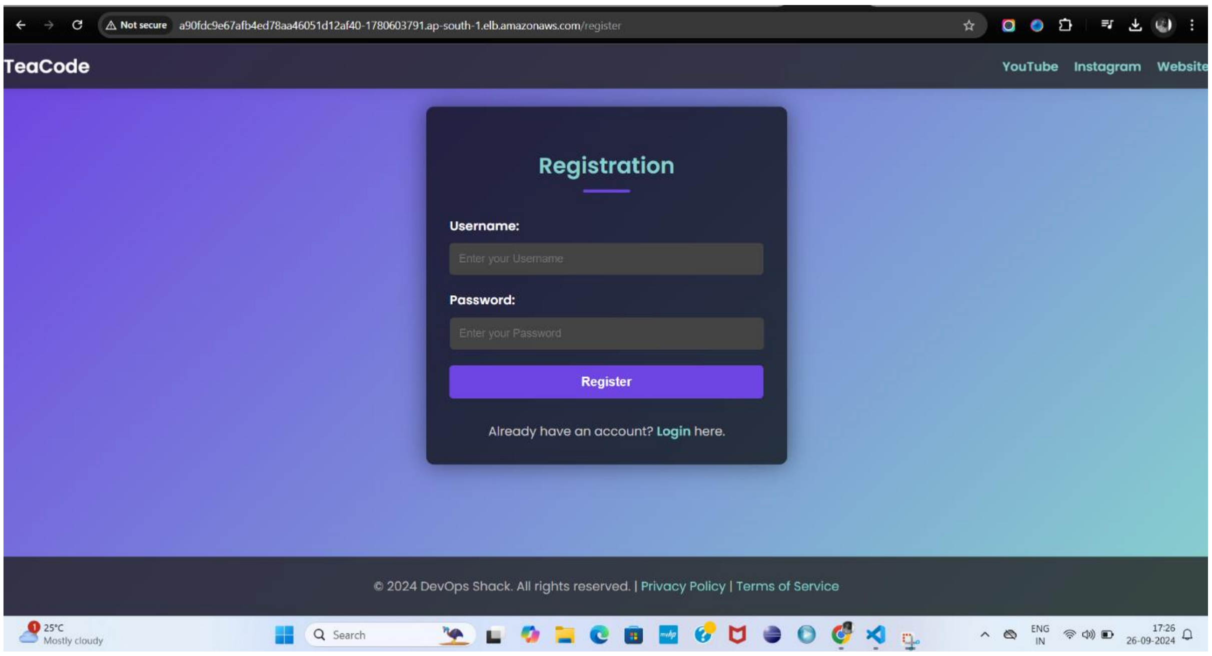
1 of 1 shown

Embedded database should be used for evaluation purposes only
The embedded database will not scale, it will not support upgrading to newer versions of SonarQube, and there is no support for migrating your data out of it into a different database engine.

SonarQube™ technology is powered by SonarSource SA
Community Edition - v9.9.6 (build 92038) - LGPL v3 - Community - Documentation - Plugins - Web API

@TeaCode1122

Divya satpute



a90fdc9e67afb4ed78aa46051d12af40-1780603791.ap-south-1.elb.amazonaws.com/register

TeaCode — YouTube · Instagram · Website

Registration

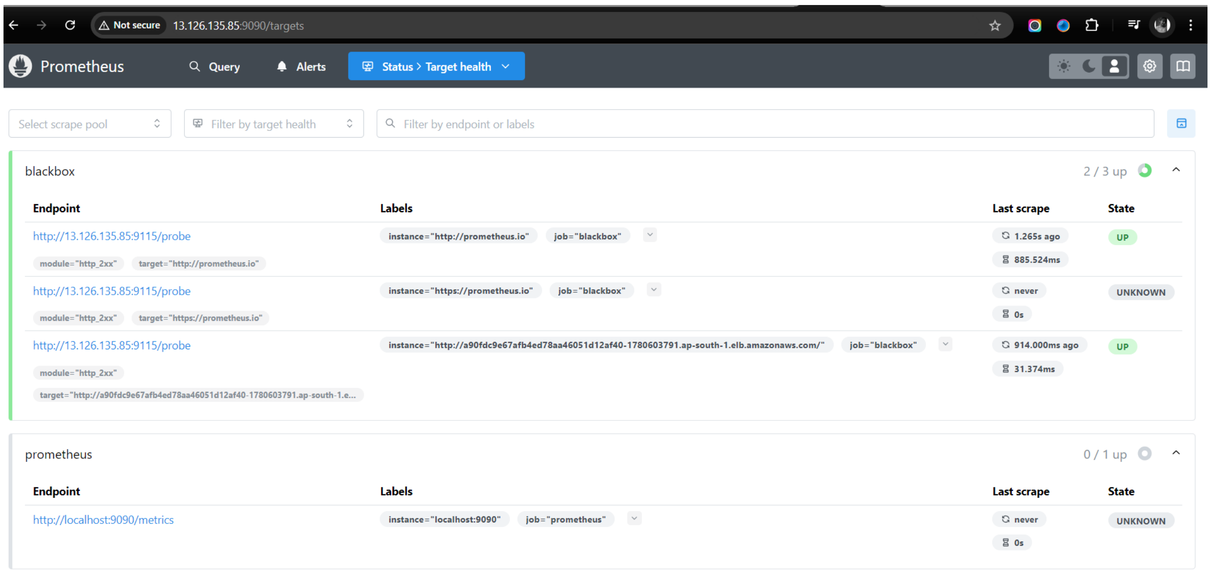
Username: Enter your Username

Password: Enter your Password

Register

Already have an account? Login here.

© 2024 DevOps Shack. All rights reserved. | Privacy Policy | Terms of Service



13.126.135.85:9090/targets

Prometheus — Query · Alerts · Status > Target health

**blackbox** — 2 / 3 up

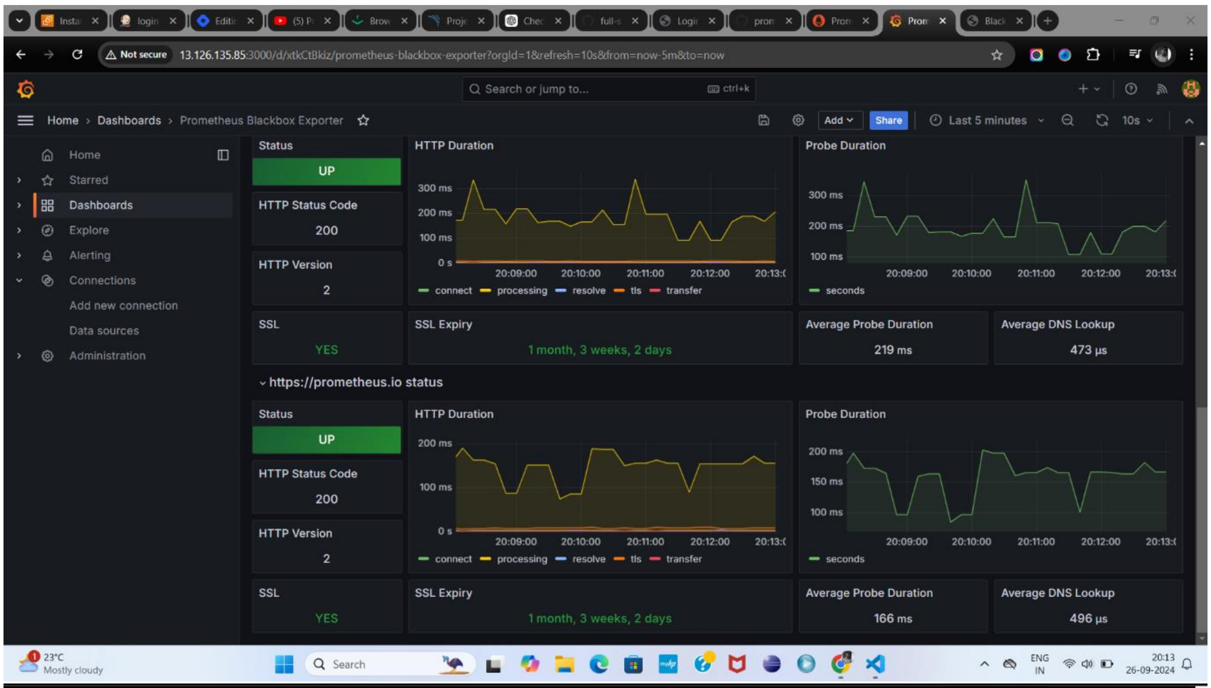
| Endpoint | Labels | Last scrape | State |
|---|---|---|---|
| http://13.126.135.85:9115/probe module="http_2xx" target="http://prometheus.io" | instance="http://prometheus.io" job="blackbox" | 1.265s ago 885.524ms | UP |
| http://13.126.135.85:9115/probe module="http_2xx" target="https://prometheus.io" | instance="https://prometheus.io" job="blackbox" | never 0s | UNKNOWN |
| http://13.126.135.85:9115/probe module="http_2xx" target="http://a90fdc9e67afb4ed78aa46051d12af40-1780603791.ap-south-1.e... | instance="http://a90fdc9e67afb4ed78aa46051d12af40-1780603791.ap-south-1.elb.amazonaws.com/" job="blackbox" | 914.000ms ago 31.374ms | UP |

**prometheus** — 0 / 1 up

| Endpoint | Labels | Last scrape | State |
|---|---|---|---|
| http://localhost:9090/metrics | instance="localhost:9090" job="prometheus" | never 0s | UNKNOWN |

**@TeaCode1122**

Divya satpute



Status
UP

HTTP Status Code
200

HTTP Version
2

SSL
YES

SSL Expiry
1 month, 3 weeks, 2 days

Average Probe Duration
219 ms

Average DNS Lookup
473 µs

https://prometheus.io status

Status
UP

HTTP Status Code
200

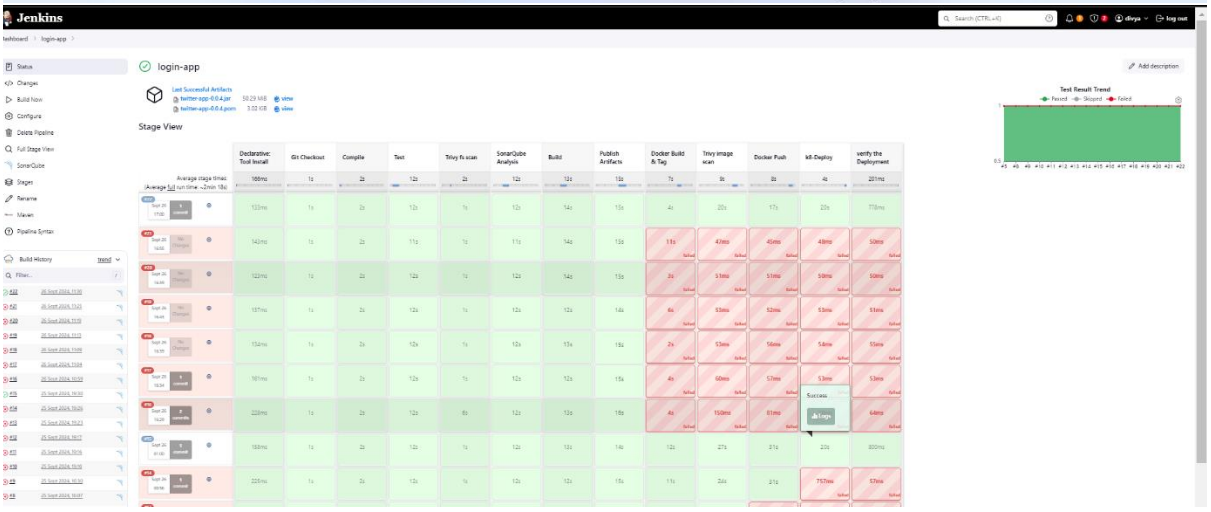
HTTP Version
2

SSL
YES

SSL Expiry
1 month, 3 weeks, 2 days

Average Probe Duration
166 ms

Average DNS Lookup
496 µs



login-app

Last Successful Artifacts
twitter-app-0.0.4.jar 50.29 MiB view
twitter-app-0.0.4.pom 3.02 KiB view

Stage View

| | Declarative: Tool Install | Git Checkout | Compile | Test | Trivy fs scan | SonarQube Analysis | Build | Publish Artifacts | Docker Build & Tag | Trivy image scan | Docker Push | k8-Deploy | verify the Deployment |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Average stage times: (Average full run time: ~2min 18s) | 166ms | 1s | 2s | 12s | 2s | 12s | 13s | 15s | 7s | 9s | 8s | 4s | 201ms |
| #22 Sept 26 17:00 | 133ms | 1s | 2s | 12s | 1s | 12s | 14s | 15s | 4s | 20s | 17s | 20s | 778ms |
| #21 Sept 26 16:00 | 143ms | 1s | 2s | 11s | 1s | 11s | 14s | 15s | 11s failed | 47ms failed | 45ms failed | 48ms failed | 50ms failed |
| #20 Sept 26 16:39 | 123ms | 1s | 2s | 12s | 1s | 12s | 14s | 15s | 3s failed | 51ms failed | 51ms failed | 50ms failed | 50ms failed |
| #19 Sept 26 16:49 | 137ms | 1s | 2s | 12s | 1s | 12s | 12s | 14s | 6s failed | 53ms failed | 52ms failed | 53ms failed | 51ms failed |

Thanks youuu did it hurrayyyyy

@TeaCode1122