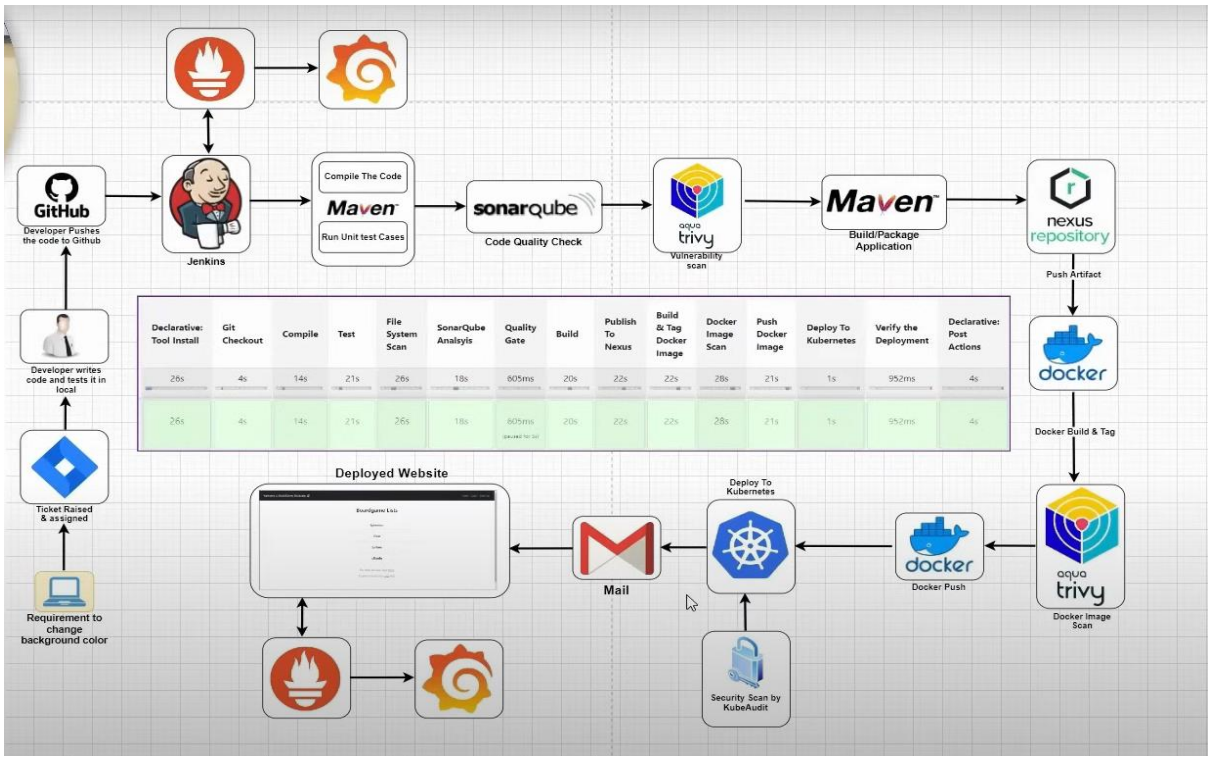
# Ekart Ultimate CI/CD Pipeline:




Step1: Take EC2 Instance for Cluster Creation One for Master and two or more for Worker nodes with Minimum t2.medium (2 CPU, 4 GB Memory)

→Launch another Instances for Jenkins, SonarQube, Nexus.

→Enable these Ports on Security Group in INBOUND Rules

**Inbound rules** (8)

Manage tags | Edit inbound rules

Search

| Type | Protocol | Port range | Source |
|---|---|---|---|
| SMTP | TCP | 25 | 0.0.0.0/0 |
| Custom TCP | TCP | 3000 - 10000 | 0.0.0.0/0 |
| HTTP | TCP | 80 | 0.0.0.0/0 |
| HTTPS | TCP | 443 | 0.0.0.0/0 |
| SSH | TCP | 22 | 0.0.0.0/0 |
| Custom TCP | TCP | 6443 | 0.0.0.0/0 |
| SMTPS | TCP | 465 | 0.0.0.0/0 |
| Custom TCP | TCP | 30000 - 32767 | 0.0.0.0/0 |

**→Setup K8-Cluster using kubeadm [K8 Version-->1.28.1]**

**→Installing Jenkins on Ubuntu:**

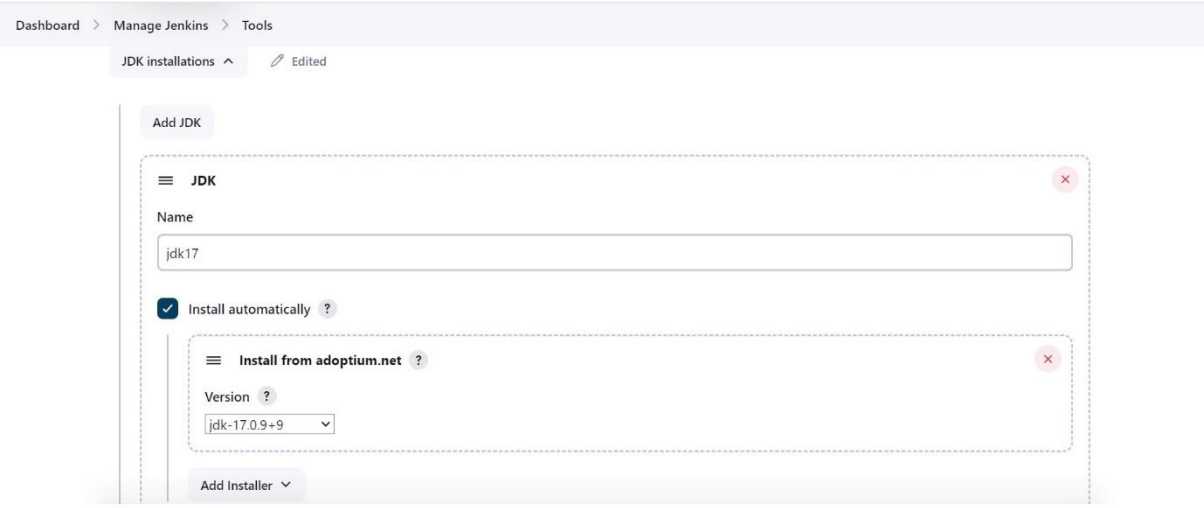
→ Take Jenkins Server and install java17 and Jenkins, trivy, docker, kubectl.

**→Install docker for future use**

## Install Plugins:

- Eclipse temurin installer
- Maven Integration plugin
- SonarQube Scanner
- Nexus Artifact uploader
- Config file provider
- OWASP dependency check
- Docker, docker pipeline, Cloud Bees, docker build
- Kubernetes, Kubernetes CLI
- Prometheus

Go To Jenkins →Manage Jenkins → Tools→SetUp all these



Dashboard > Manage Jenkins > Tools

Maven installations ^ Edited

Add Maven

**Maven**

Name

maven3

☑ Install automatically ?

**Install from Apache**

Version

3.6.3

Save Apply

Dashboard > Manage Jenkins > Tools

SonarQube Scanner installations ^ Edited

Add SonarQube Scanner

**SonarQube Scanner**

Name

sonar-scanner

☑ Install automatically ?

**Install from Maven Central**

Version

SonarQube Scanner 5.0.1.3006

Save Apply

Dashboard > Manage Jenkins > Tools

Dependency-Check installations ^ Edited

Add Dependency-Check

**Dependency-Check**

Name

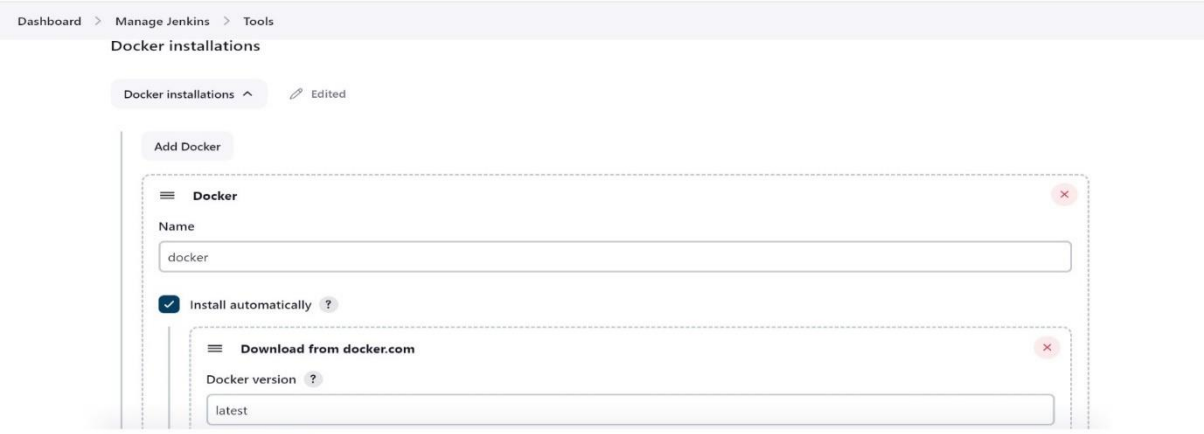
Dp-Check

☑ Install automatically ?

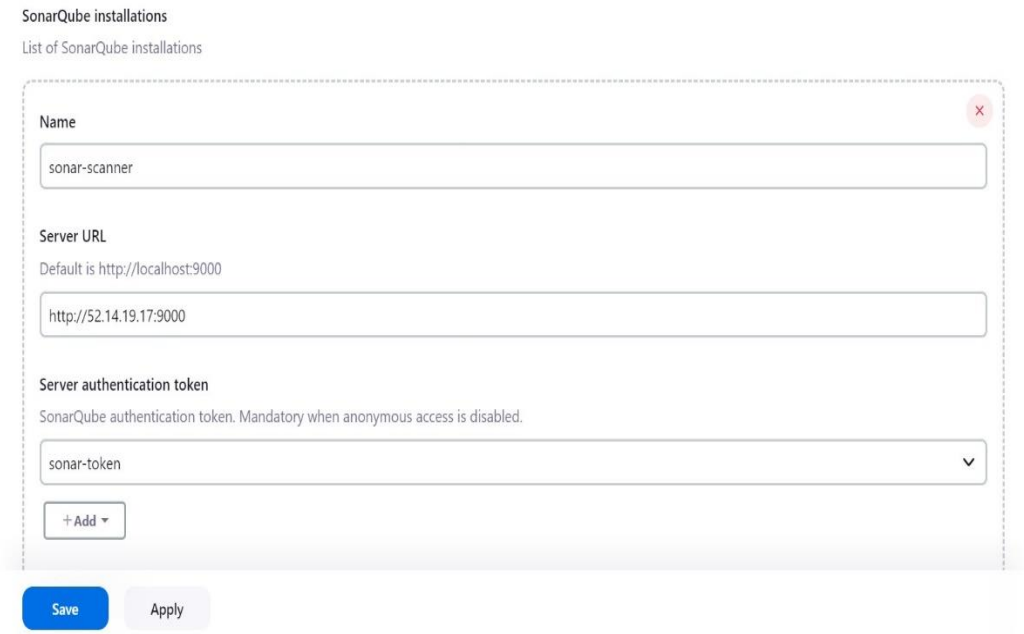
**Install from github.com**

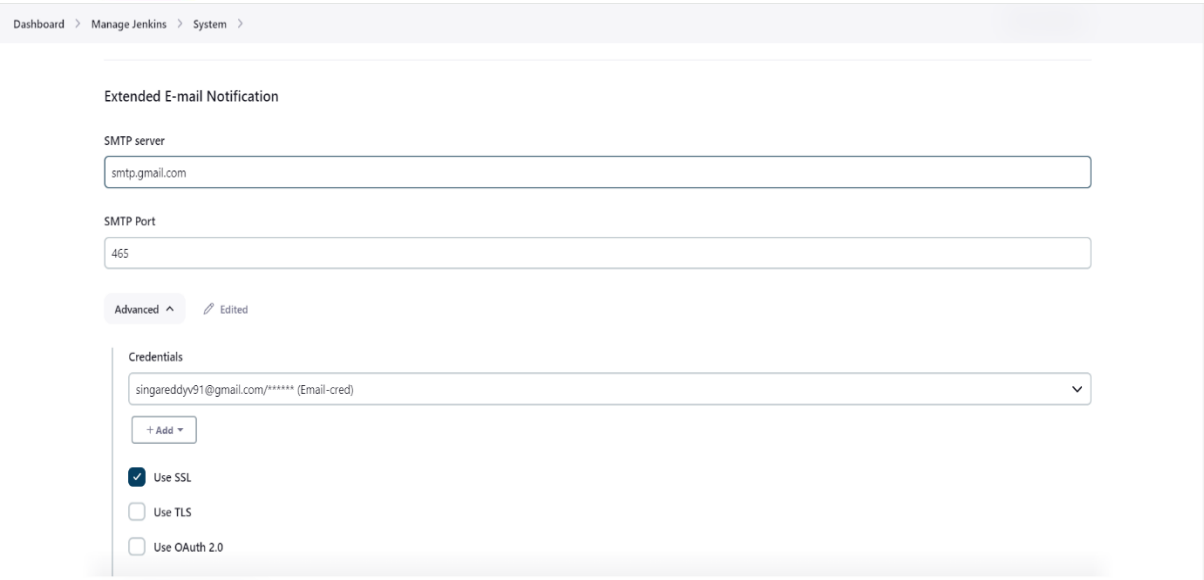
Version

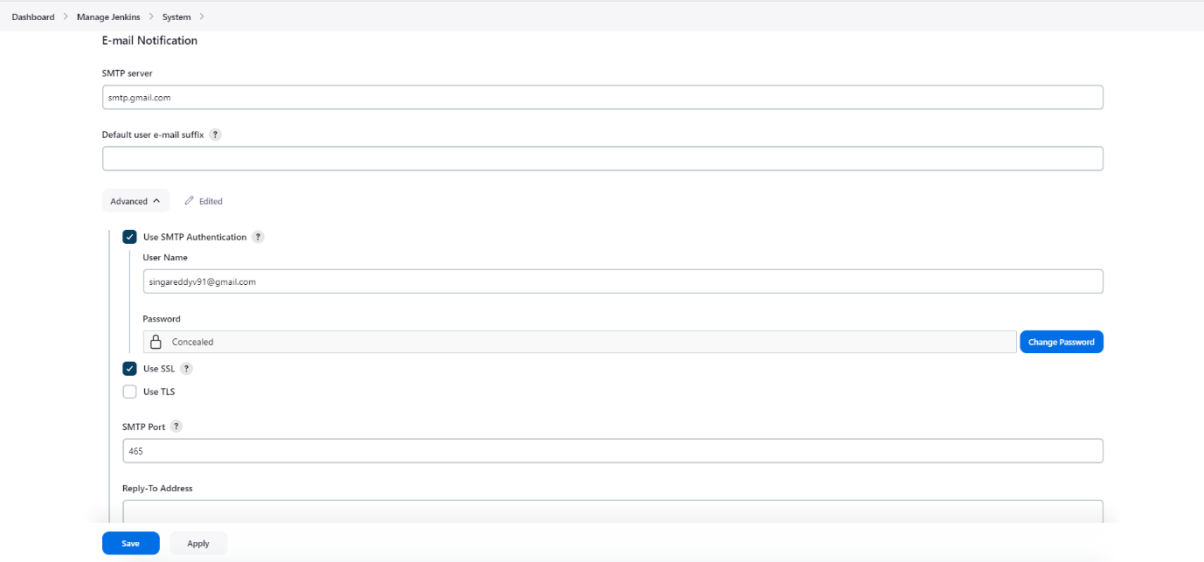
dependency-check 6.5.1



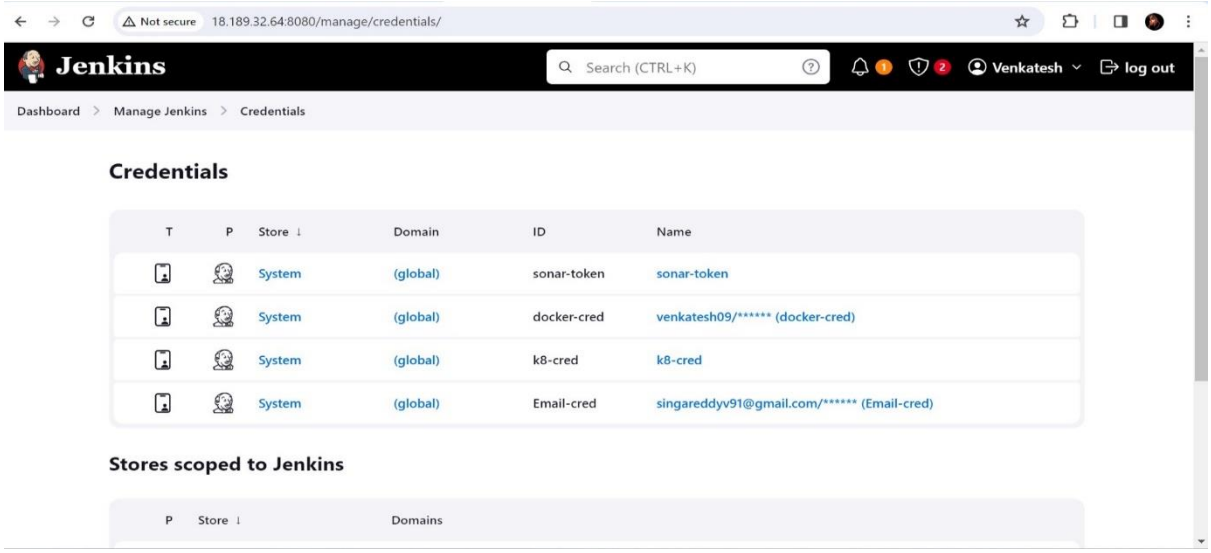

Go To Jenkins →Manage Jenkins →Systems→SetUp all these



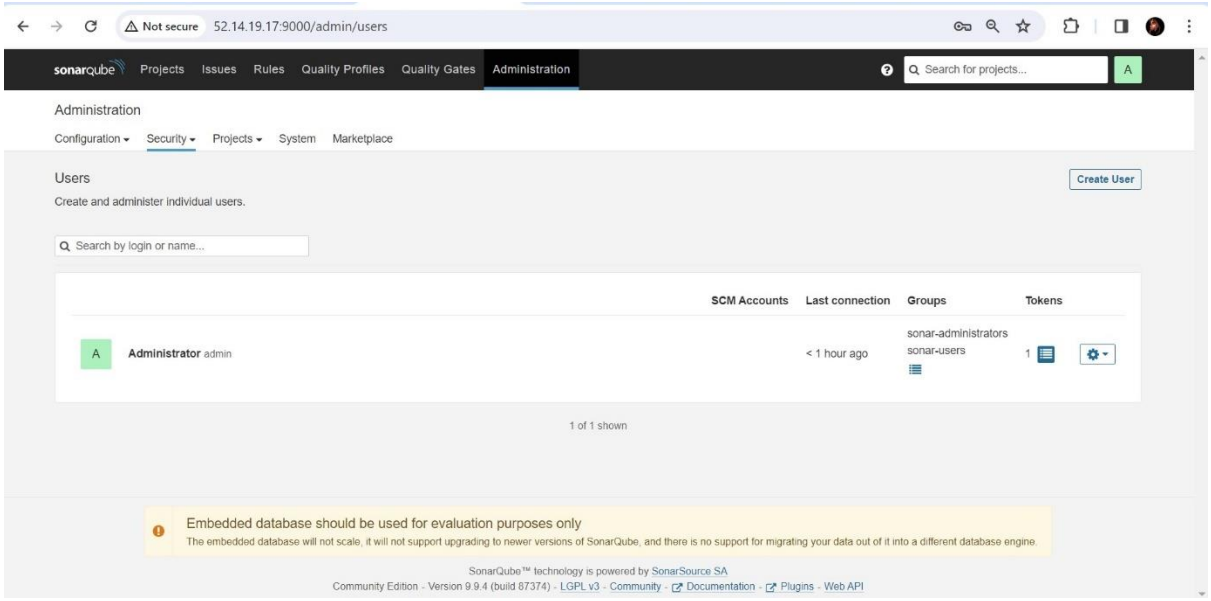


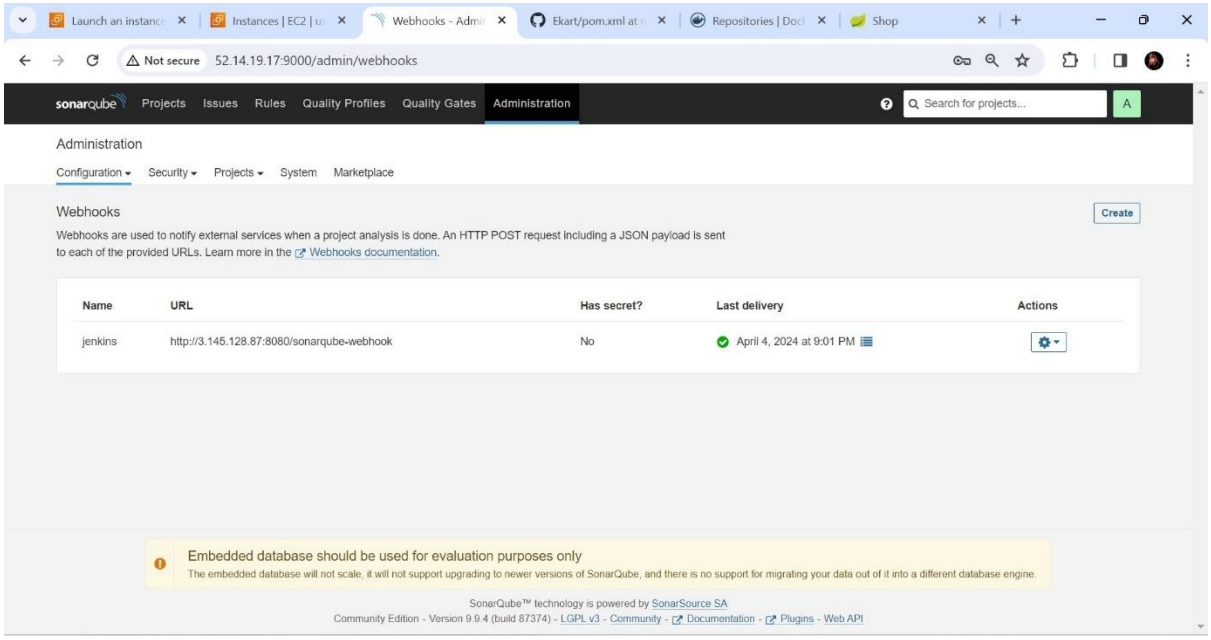

Go To Jenkins →Manage Jenkins →Credentials→SetUp all these




For SonarQube Integrate to Jenkins:

In SonarQube → Go to Administration→Security→Create User with Token for Credentials.




# For Nexus Integrate to Jenkins:

Go To Jenkins →Manage Jenkins →Managed Files→SetUp all these


**→Go to Kubernetes Master node Create RBAC (Role Based Access Control)**

## Pipeline:

The Ultimate CICD Corporate DevOps Pipeline Project

```
pipeline {
    agent any
    tools {
        jdk 'jdk17'
        maven 'maven3'
    }
    environment {
        SCANNER_HOME= tool 'sonar-scanner'
    }
```

```
    stages {
        stage('Git Checkout') {
            steps {
                checkout scmGit(branches: [[name: '*/master']], extensions: [], userRemoteConfigs: [[url: 'https://github.com/Singareddy-Venkatesh/Ekart.git']])
            }
        }
        stage('Compile') {
            steps {
                sh 'mvn compile'
            }
        }
        stage('Test') {
            steps {
                sh 'mvn test -DskipTests=true'
            }
        }
        stage('SonarQube Analysis') {
            steps {
```

```
            withSonarQubeEnv('sonar-scanner') {
                sh ''' $SCANNER_HOME/bin/sonar-scanner -Dsonar.projectName=Ekart -Dsonar.projectKey=Ekart -Dsonar.java.binaries=. '''
            }
        }
    }
    stage('Quality Gate') {
        steps {
            script {
                waitForQualityGate abortPipeline: false, credentialsId: 'sonar-token'
            }
        }
    }
    stage('Maven package') {
        steps {
            sh 'mvn package -DskipTests=true'
        }
    }
```

```
        stage('OWASP Depencency Check') {
            steps {
                dependencyCheck additionalArguments: '--scan ./ --format XML', odcInstallation: 'Dp-Check'
                dependencyCheckPublisher pattern: '**/dependency-check-report.xml'
            }
        }
        stage('Deploy to Nexus') {
            steps {
                withMaven(globalMavenSettingsConfig: 'Global-Settings', jdk: 'jdk17', maven: 'maven3', mavenSettingsConfig: '', traceability: true) {
                    sh "mvn deploy -DskipTests=true"
                }
            }
        }
        stage('Build & Docker Image') {
            steps {
                script {
```

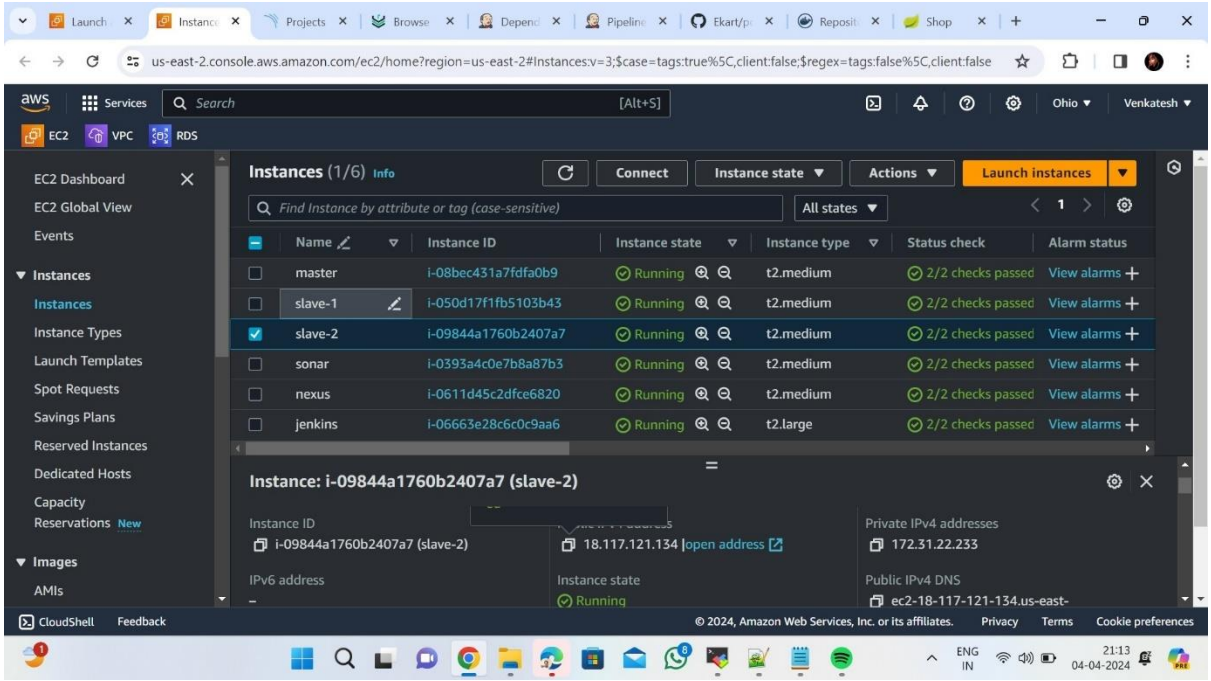
```
                withDockerRegistry(credentialsId: 'docker-cred',
toolName: 'docker') {
                    sh "docker build -t venkatesh09/ekart:latest -f
docker/Dockerfile ."
                }
            }
        }
    }
    stage('Trivy') {
        steps {
            sh "trivy image venkatesh09/ekart:latest > trivy-
report.txt"
        }
    }
    stage('Docker Push Image') {
        steps {
            script {
                withDockerRegistry(credentialsId: 'docker-cred',
toolName: 'docker') {
                    sh "docker push venkatesh09/ekart:latest"
                }
```

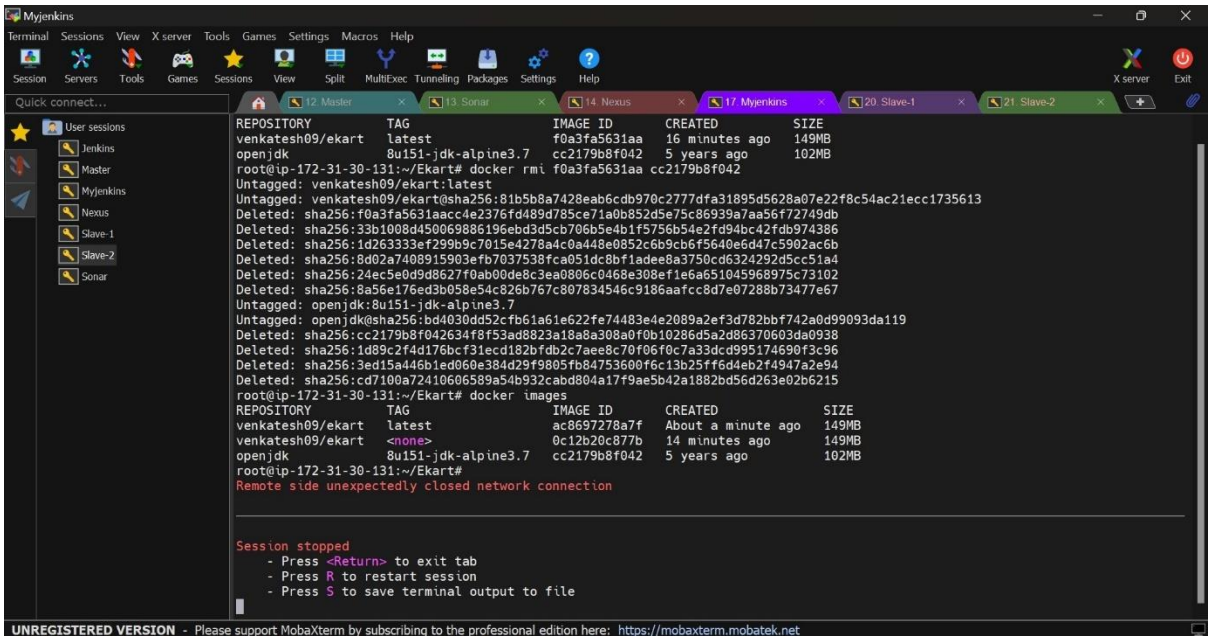
```
            }
        }
    }
    stage('Deploy to Kubernetes') {
        steps {
            withKubeConfig(caCertificate: '', clusterName:
'kubernetes', contextName: '', credentialsId: 'k8-cred',
namespace: 'webapps', restrictKubeConfigAccess: false,
serverUrl: 'https://172.31.29.58:6443') {
                sh "kubectl apply -f deploymentservice.yml"
                sh "kubectl get svc -n webapps"
            }
        }
    }
}
post {
    always {
    script {
    def jobName = env.JOB_NAME
    def buildNumber = env.BUILD_NUMBER
```

```
def pipelineStatus = currentBuild.result ?: 'UNKNOWN'

def bannerColor = pipelineStatus.toUpperCase() ==
'SUCCESS' ? 'green' : 'red'



def body = """

<html>

<body>

<div style="border: 4px solid ${bannerColor}; padding:
10px;">

<h2>${jobName} - Build ${buildNumber}</h2>

<div style="background-color: ${bannerColor}; padding:
10px;">

<h3 style="color: white;">Pipeline Status:
${pipelineStatus.toUpperCase()}</h3>

</div>

<p>Check the <a href="${BUILD_URL}">console
output</a>.</p>

</div>

</body>

</html>

"""
```

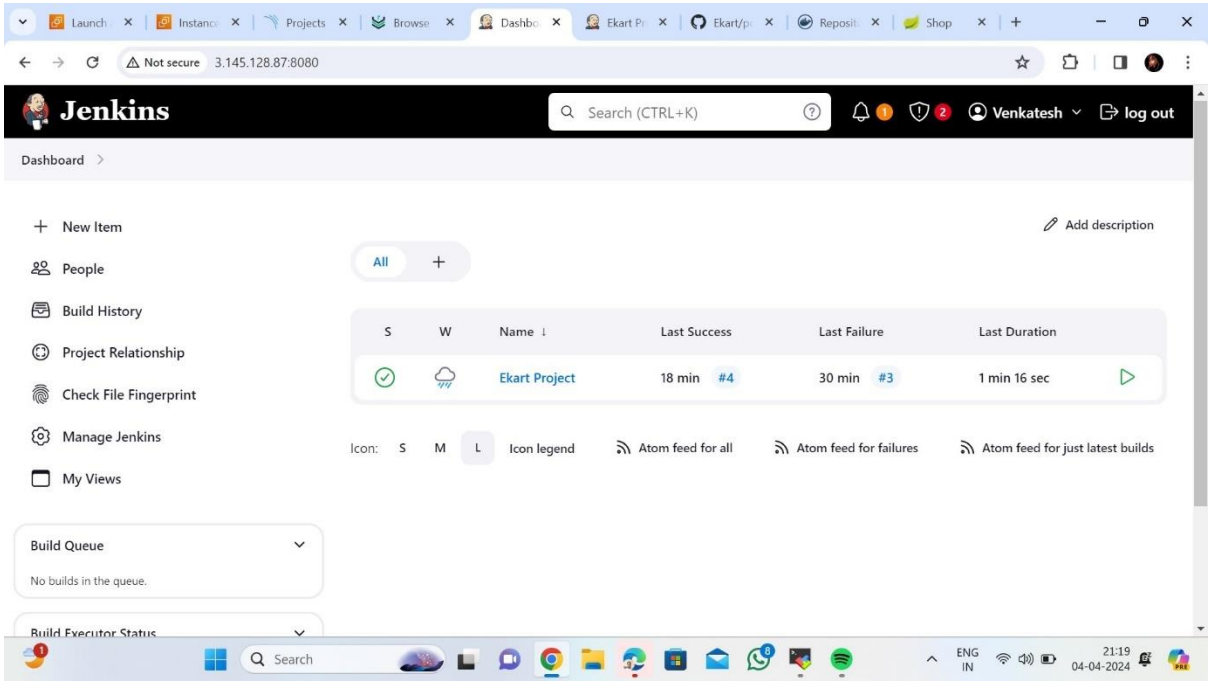
```
        emailext (
            subject: "${jobName} - Build ${buildNumber} - ${pipelineStatus.toUpperCase()}",
            body: body,
            to: 'singareddyv91@gmail.com',
            from: 'jenkins@example.com',
            replyTo: 'jenkins@example.com',
            mimeType: 'text/html',
            attachmentsPattern: 'trivy-report.txt'
            )
        }
        }
    }
}
```

# Launches EC2 Instances:

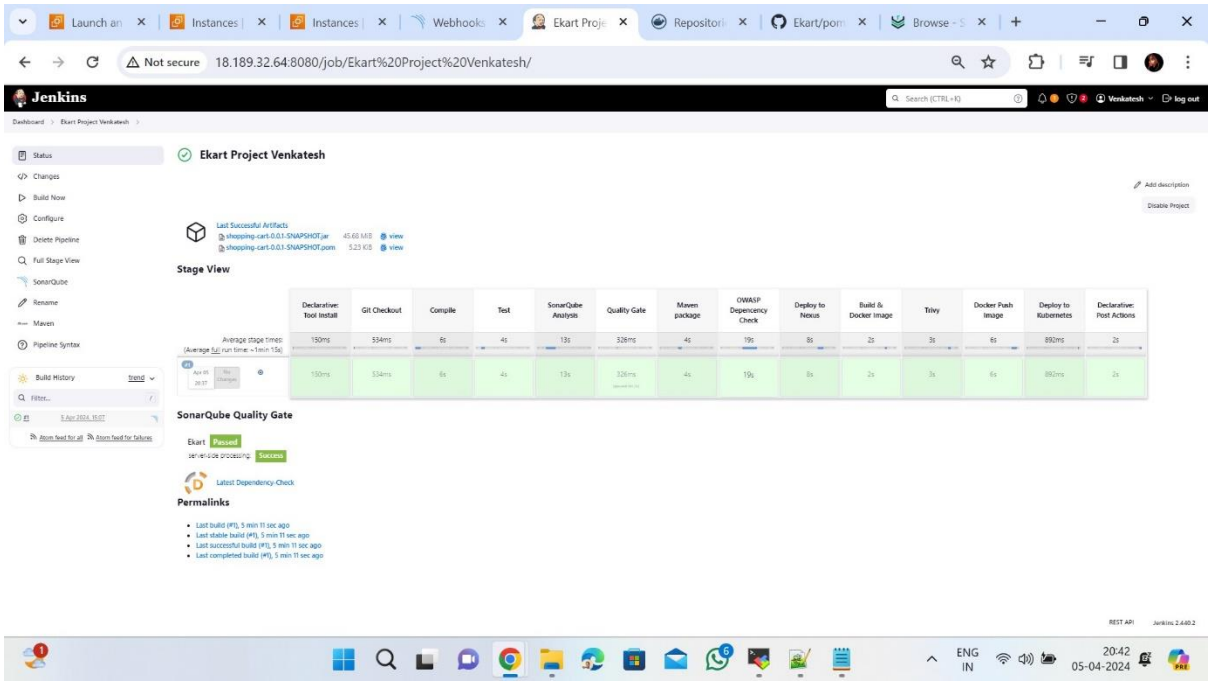


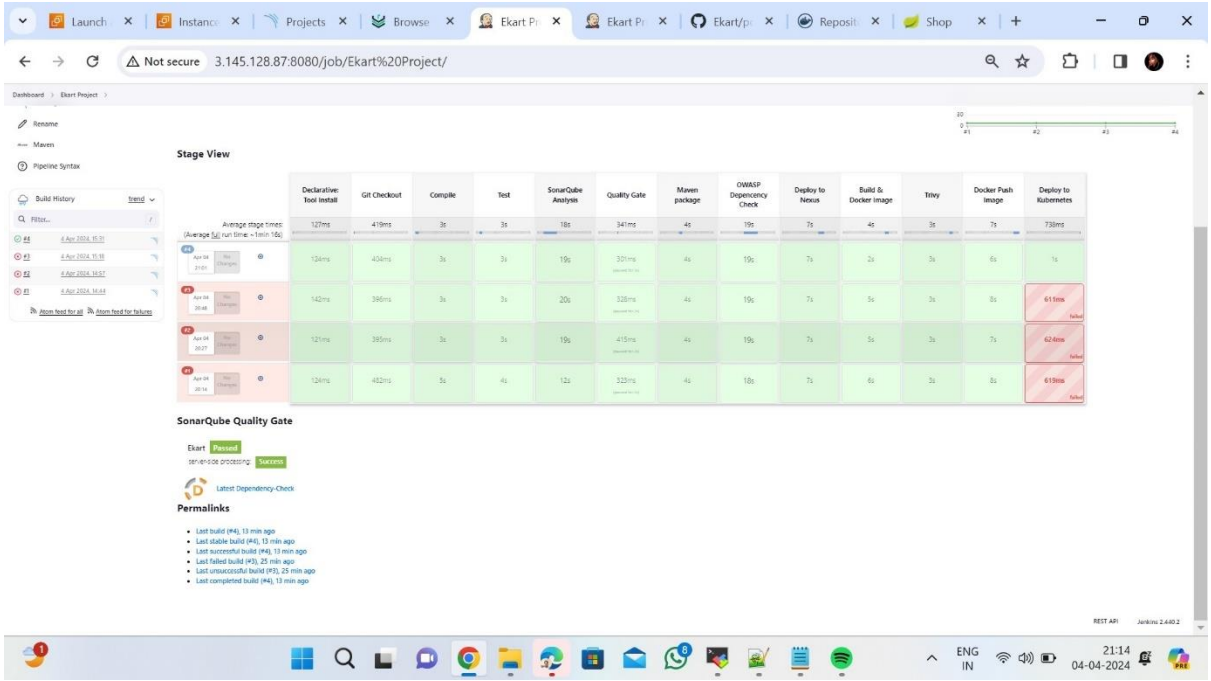



# Created A NEW JOB:

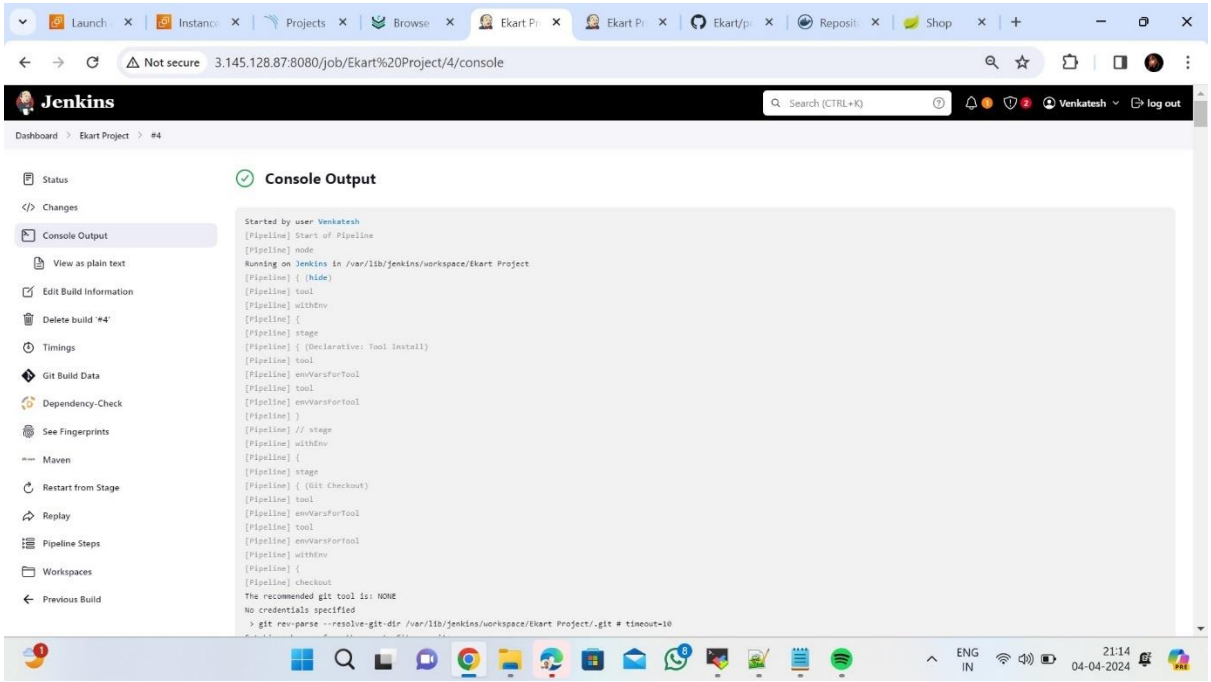


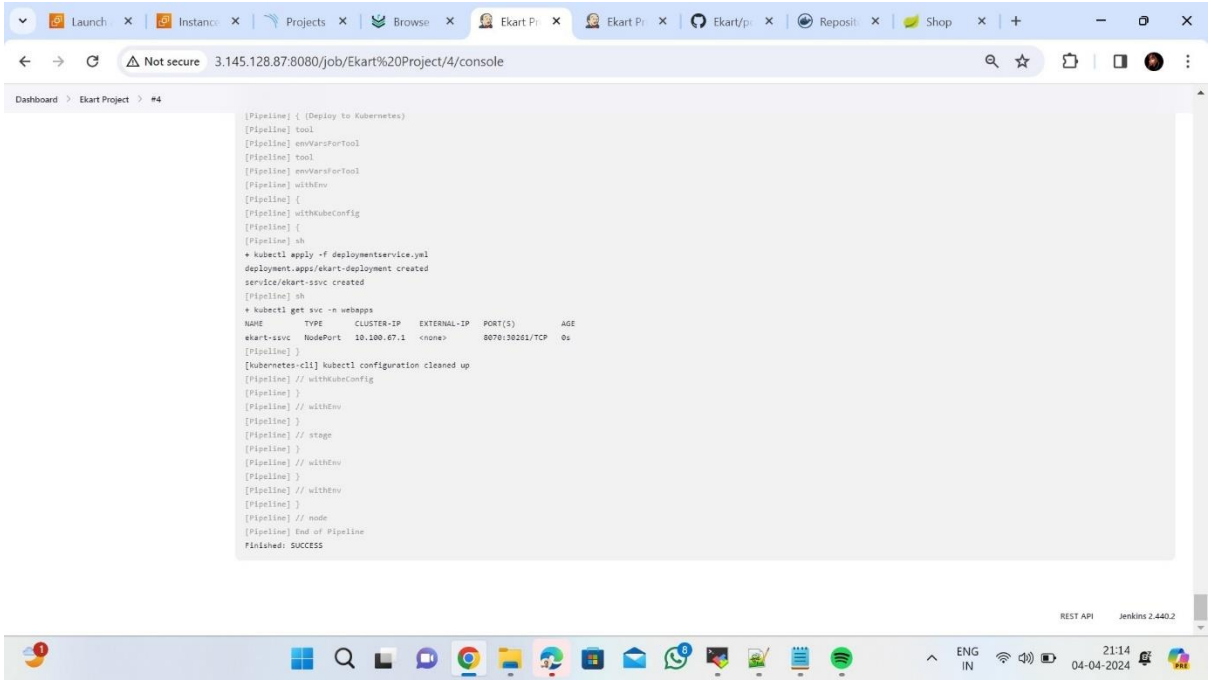

# Successfully Runs the Pipeline:



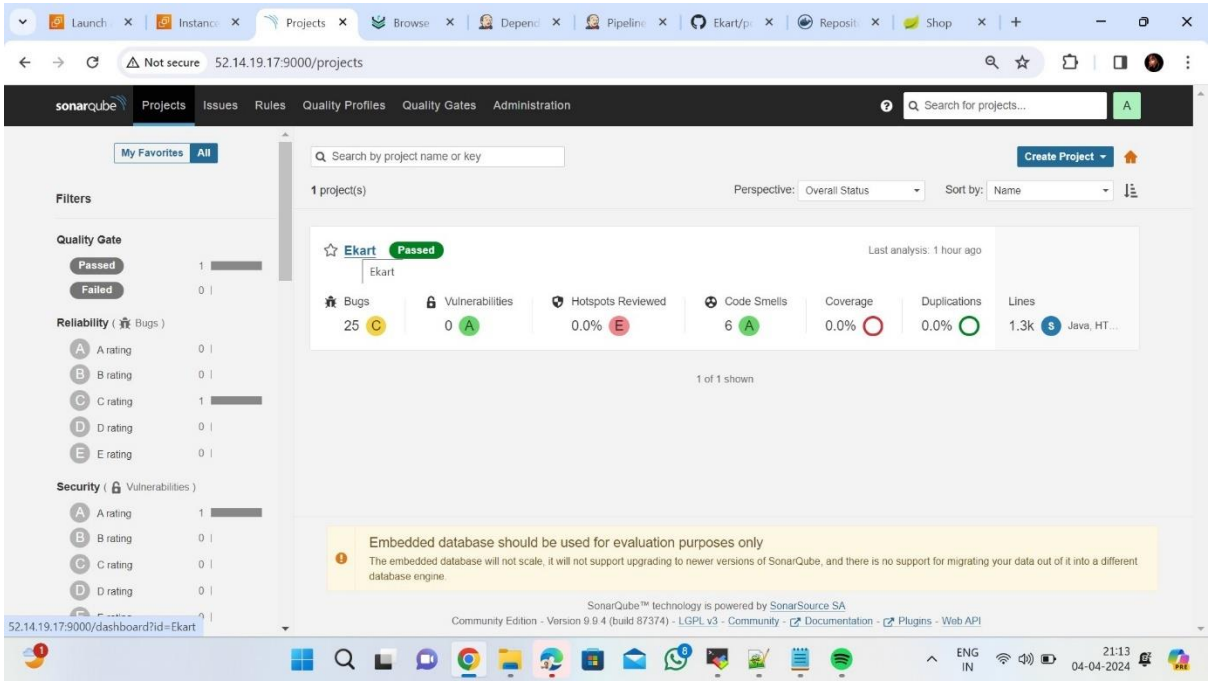



# Console Output:

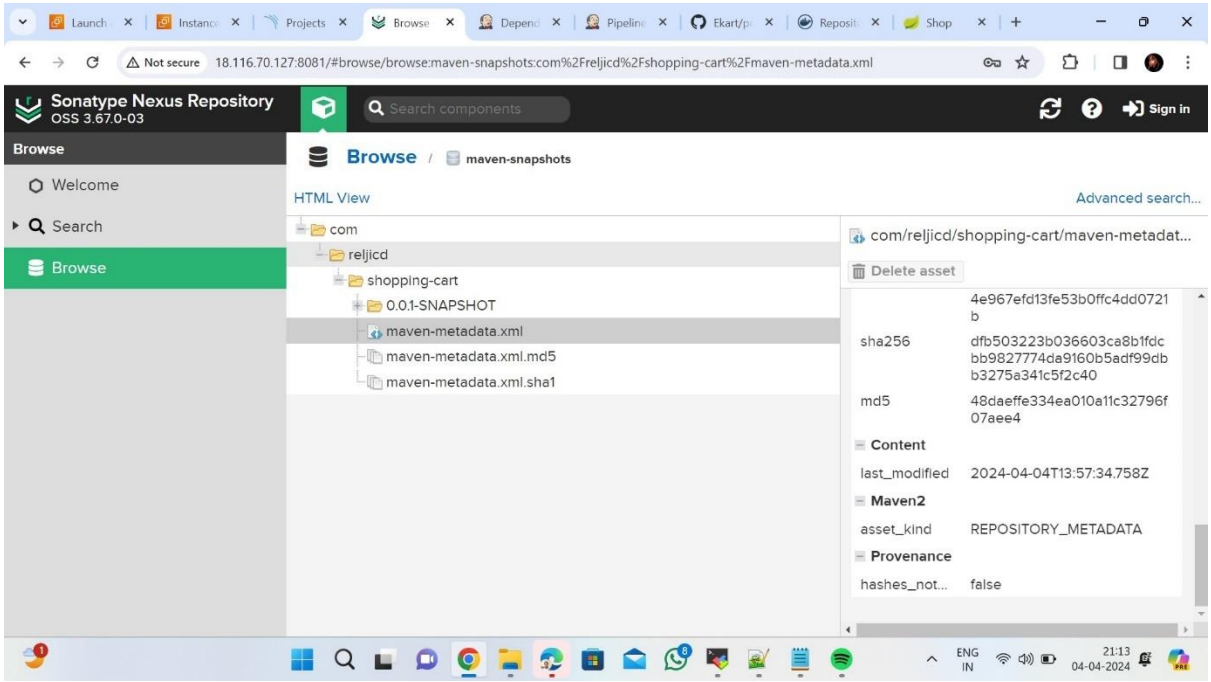




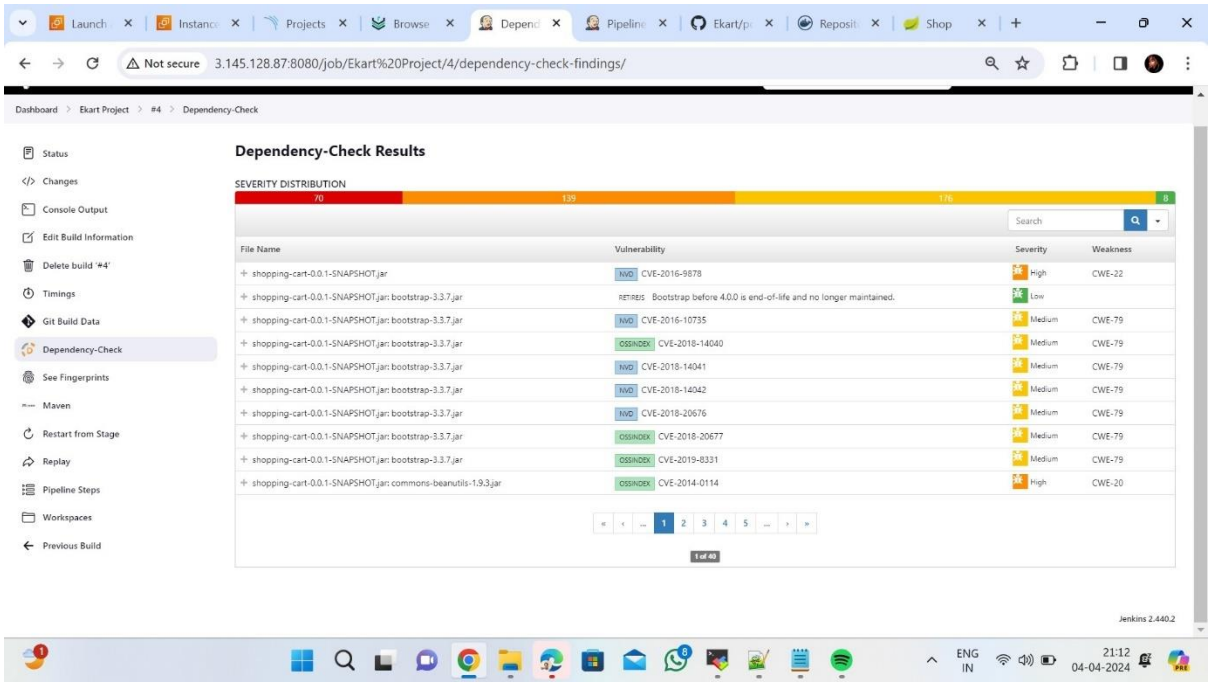

# SonarQube Output:



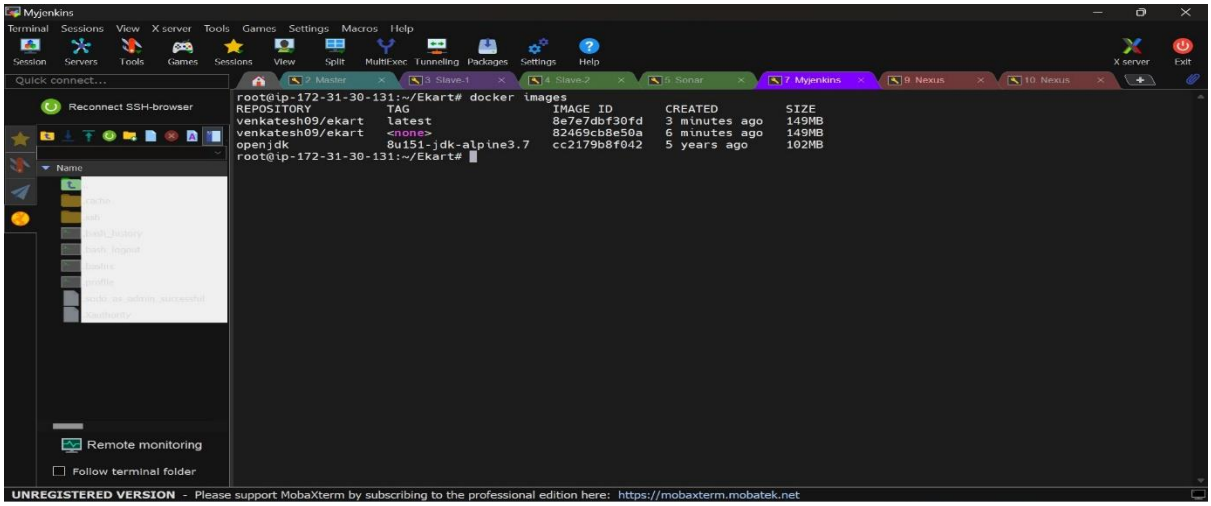
# Nexus:

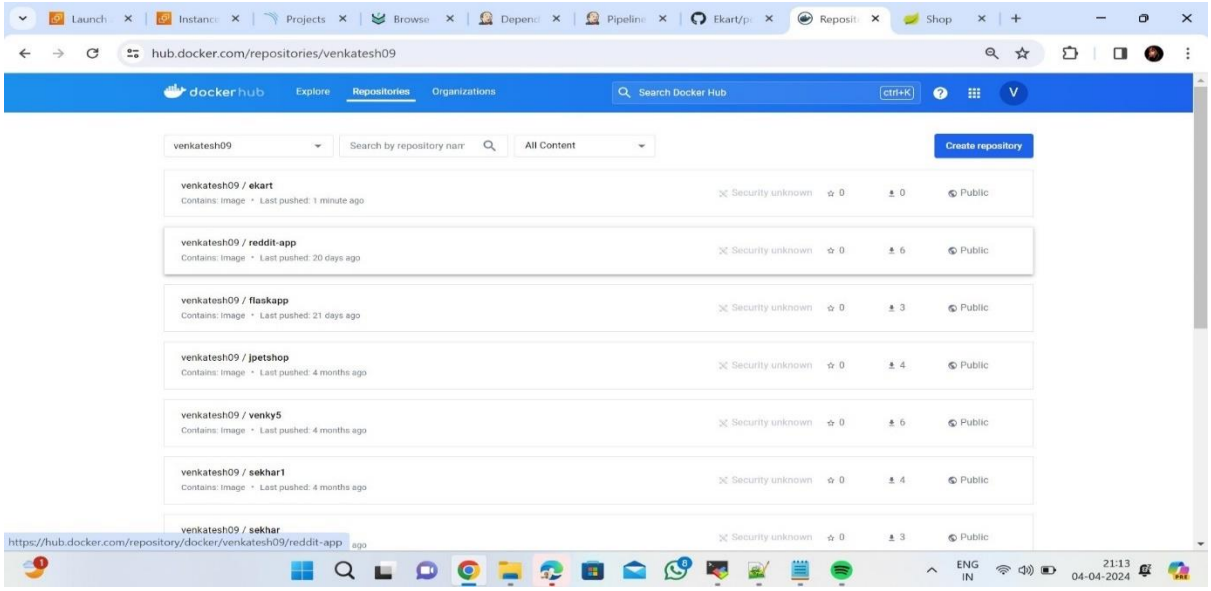


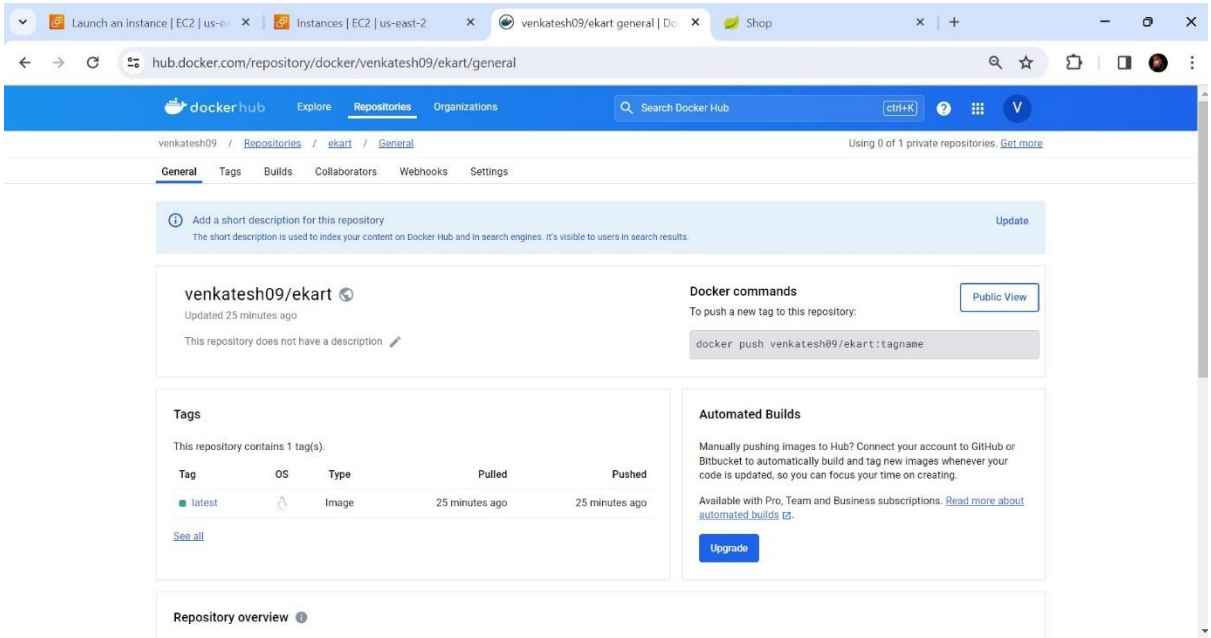

# Dependency Check:



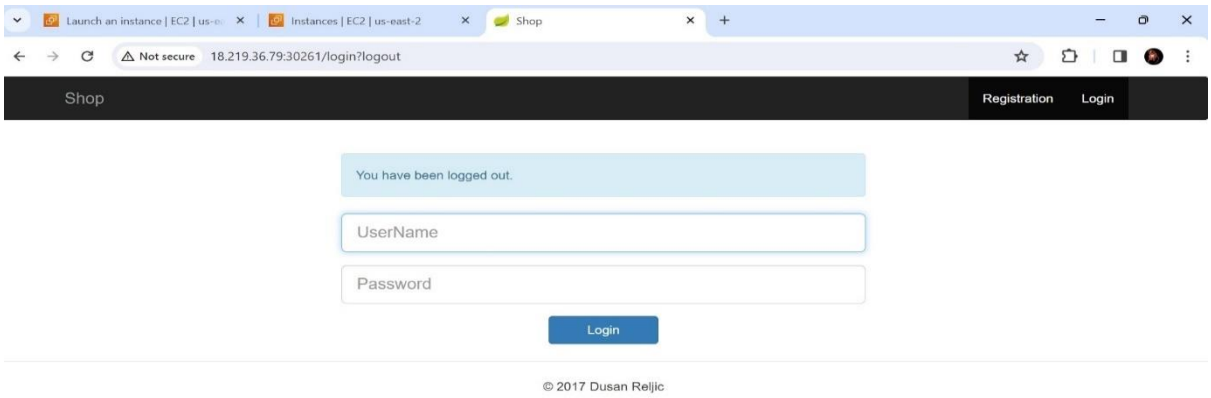
# Docker:


Myjenkins
Terminal Sessions View X server Tools Games Settings Macros Help
Reconnect SSH-browser
Remote monitoring
Follow terminal folder
root@ip-172-31-30-131:~/Ekart# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
venkatesh09/ekart latest 8e7e7dbf30fd 3 minutes ago 149MB
venkatesh09/ekart <none> 82469cb8e50a 6 minutes ago 149MB
openjdk 8u151-jdk-alpine3.7 cc2179b8f042 5 years ago 102MB
root@ip-172-31-30-131:~/Ekart#
UNREGISTERED VERSION - Please support MobaXterm by subscribing to the professional edition here: https://mobaxterm.mobatek.net

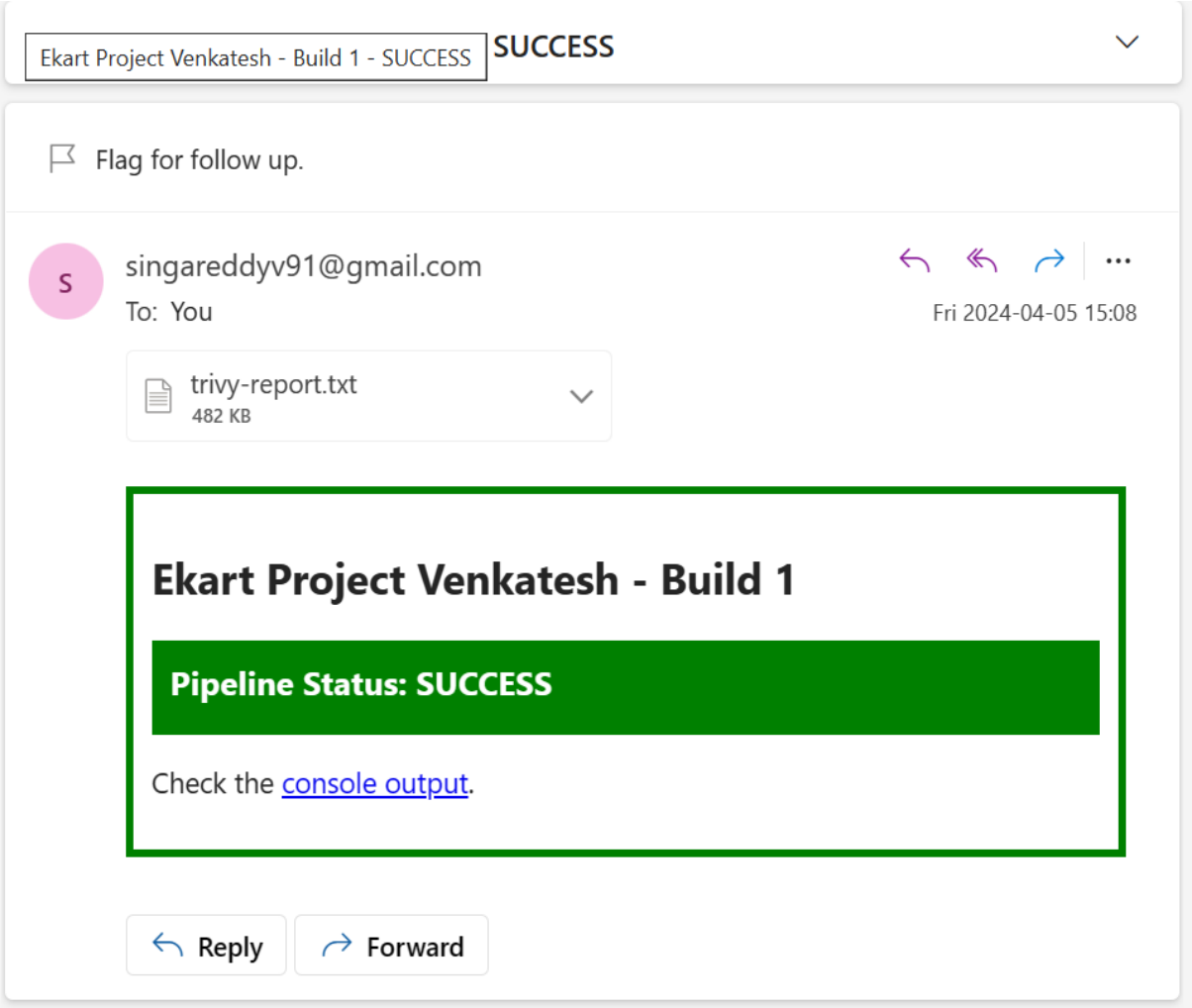

hub.docker.com/repositories/venkatesh09
dockerhub Explore Repositories Organizations
Search Docker Hub
venkatesh09
All Content
Create repository
venkatesh09 / ekart
venkatesh09 / reddit-app
venkatesh09 / flaskapp
venkatesh09 / jpetshop
venkatesh09 / venky5
venkatesh09 / sekhar1
venkatesh09 / sekhar
https://hub.docker.com/repository/docker/venkatesh09/reddit-app


hub.docker.com/repository/docker/venkatesh09/ekart/general
dockerhub Explore Repositories Organizations
Search Docker Hub
venkatesh09 / Repositories / ekart / General
Using 0 of 1 private repositories. Get more
General Tags Builds Collaborators Webhooks Settings
Add a short description for this repository
The short description is used to index your content on Docker Hub and in search engines. It's visible to users in search results.
Update
venkatesh09/ekart
Updated 25 minutes ago
This repository does not have a description
Docker commands
To push a new tag to this repository:
docker push venkatesh09/ekart:tagname
Public View
Tags
This repository contains 1 tag(s).
Tag OS Type Pulled Pushed
latest Image 25 minutes ago 25 minutes ago
See all
Automated Builds
Manually pushing images to Hub? Connect your account to GitHub or Bitbucket to automatically build and tag new images whenever your code is updated, so you can focus your time on creating.
Available with Pro, Team and Business subscriptions. Read more about automated builds
Upgrade
Repository overview

# Final Output:




# Email Notification:



# Monitoring our websites by using Prometheus, Grafana, Blackbox,NodeExporter

## Highly recomended to follow the steps

**Links to download Prometheus, Node_Exporter & black Box exporter https://prometheus.io/download/**
**Links to download Grafana**
**https://grafana.com/grafana/download**

**Other link from video**
**https://github.com/prometheus/blackbox_exporter**