

Boardgame Ultimate CI/CD Pipeline:

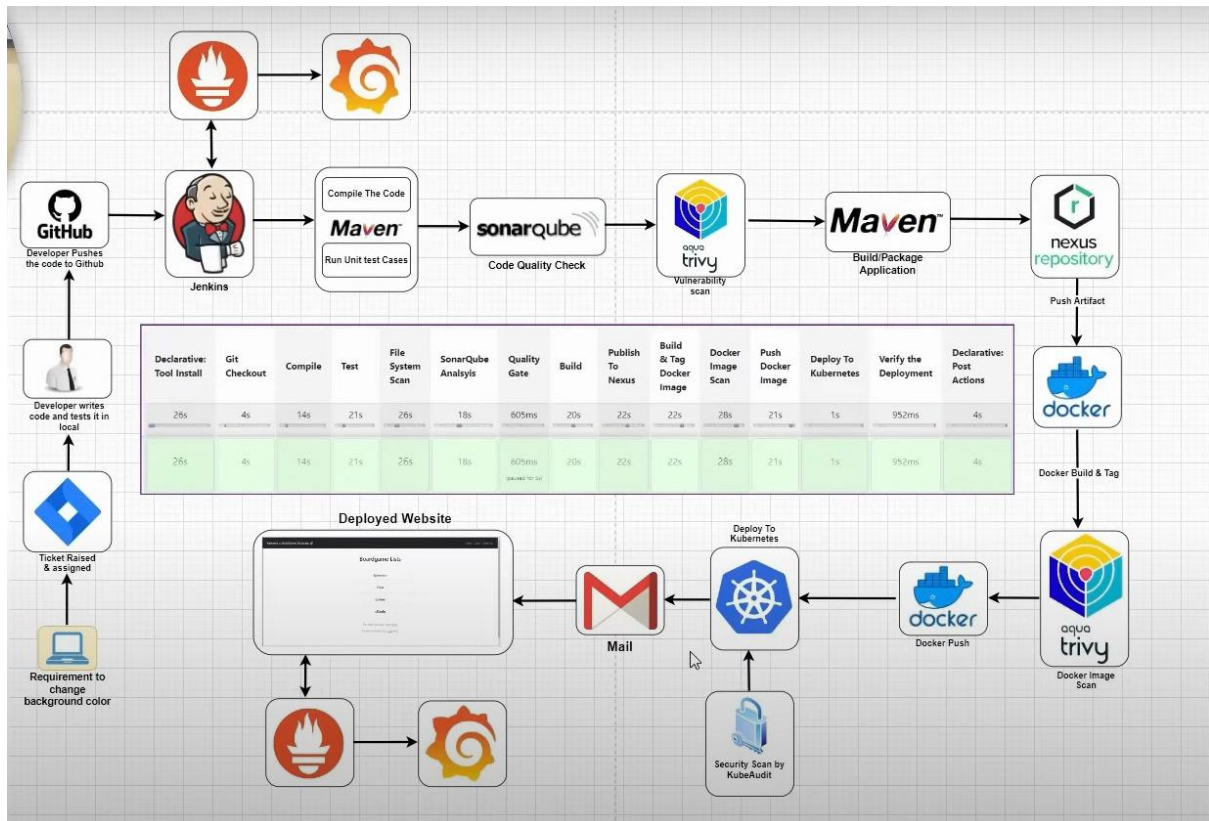
NAME: S. Venkateswarareddy

EMAIL: Singareddyv91@gmail.com

PHONE NO: 9491029062

BATCH NO: 3PM, 113

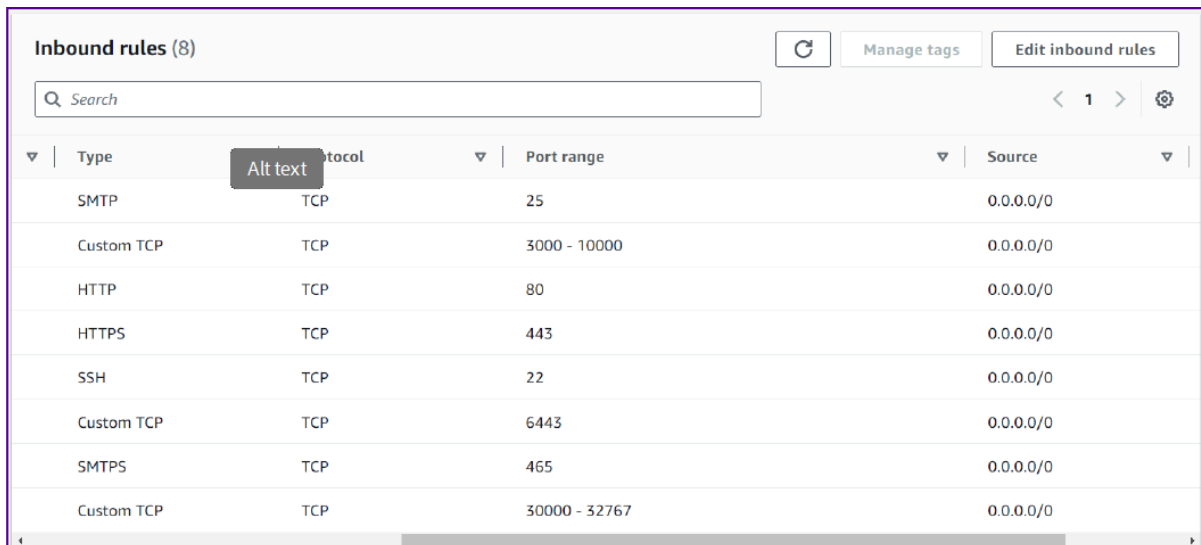
COURSE: AWS AND DEVOPS



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:

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 → Set Up all these

Dashboard > Manage Jenkins > Tools

JDK installations ^ Edited

Add JDK

≡ JDK

Name

jdk17

☒ Install automatically ?

≡ Install from adoptium.net ?

Version ?

jdk-17.0.9+9

Add Installer ▾

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



☒ Install automatically 

 **Install from github.com** 



Version

Dashboard > Manage Jenkins > Tools


Docker installations



Docker installations   Edited


Add Docker

 **Docker** 

Name

☒ Install automatically 



 **Download from docker.com** 

Docker version 

Go To Jenkins → Manage Jenkins → Systems → Set Up all these

SonarQube installations

List of SonarQube installations

 **Name** 

Server URL

Default is http://localhost:9000

Server authentication token

SonarQube authentication token. Mandatory when anonymous access is disabled.

Dashboard > Manage Jenkins > System >

Extended E-mail Notification

SMTP server
smtp.gmail.com

SMTP Port
465

Advanced Edited

Credentials
singareddyv91@gmail.com/***** (Email-cred)

+ Add

☒ Use SSL
☐ Use TLS
☐ Use OAuth 2.0

Dashboard > Manage Jenkins > System >

E-mail Notification

SMTP server
smtp.gmail.com

Default user e-mail suffix

Advanced Edited

☒ Use SMTP Authentication

User Name
singareddyv91@gmail.com

Password
 Concealed

☒ Use SSL
☐ Use TLS

SMTP Port
465

Reply-To Address

Go To Jenkins → Manage Jenkins → Credentials → Set Up all these

← → ↻ ⚠ Not secure 18.189.32.64:8080/manage/credentials/ ☆ 📁 📱 🔍

Jenkins 🔍 Search (CTRL+K) 🔔 1 🛡 2 👤 Venkatesh ▾ 🚪 log out

Dashboard > Manage Jenkins > Credentials

Credentials

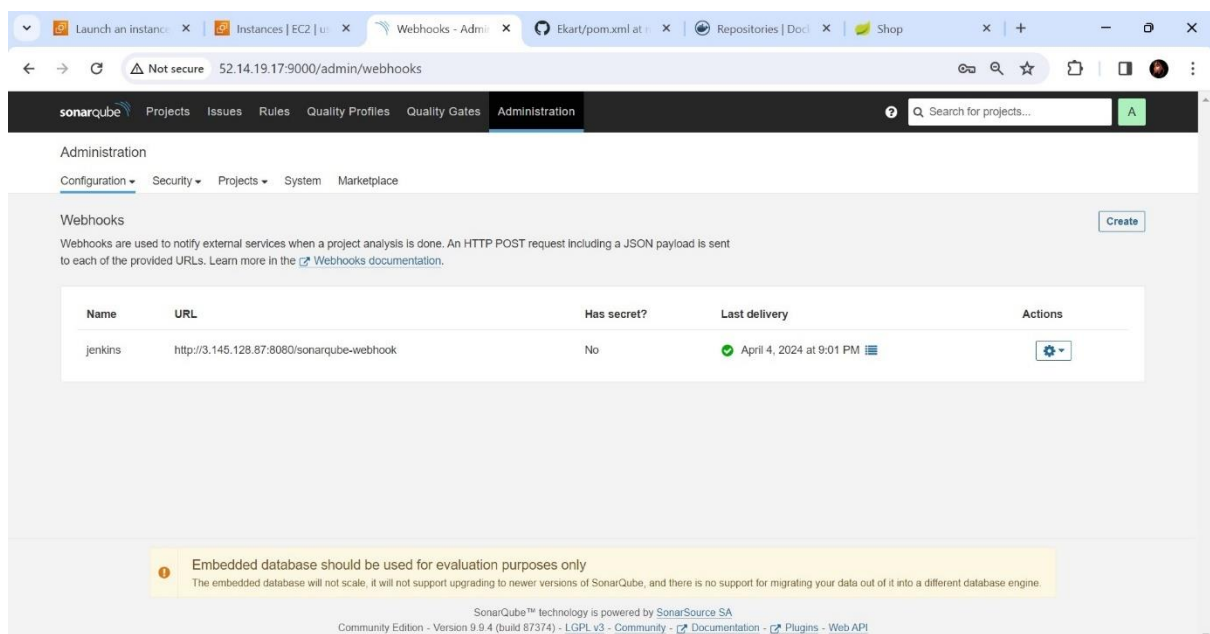
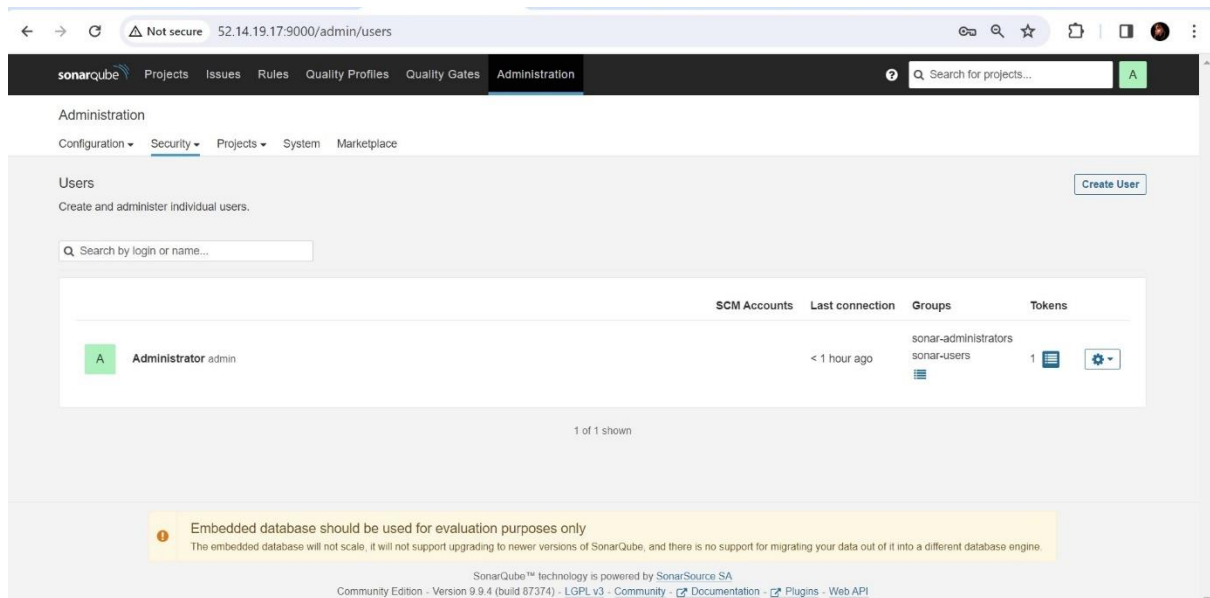
T	P	Store	Domain	ID	Name
		System	(global)	sonar-token	sonar-token
		System	(global)	docker-cred	venkatesh09/***** (docker-cred)
		System	(global)	k8-cred	k8-cred
		System	(global)	Email-cred	singareddyv91@gmail.com/***** (Email-cred)

Stores scoped to Jenkins

P	Store	Domains
---	-------	---------

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 → Set Up all these



→ Go to Kubernetes Master node Create RBAC (RoleBased Access Control)

Pipeline:

Boardgame Ultimate CI/CD Pipeline

```
pipeline {
```

```
    agent any
```

```
    tools {
```

```
        jdk 'jdk17'
```

```
        maven 'maven3'
```

```
    }
```

```
    environment {
```

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

stages {
    stage('Git Checkout') {
        steps {
            checkout scmGit(branches: [[name: '*/main']],
extensions: [], userRemoteConfigs: [[url:
'https://github.com/Singareddy-Venkatesh/Boardgame.git']])
        }
    }

    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-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'
    }
}

stage('OWASP Depencency Check') {
    steps {
        dependencyCheck additionalArguments: '--scan ./ --
format XML', odciInstallation: '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"
        }
    }
}

```

```
    }  
  }  
}  
  
stage('Build & Docker Image') {  
  steps {  
    script {  
      withDockerRegistry(credentialsId: 'docker-cred',  
toolName: 'docker') {  
        sh "docker build -t  
venkatesh09/boardgame:latest ."  
      }  
    }  
  }  
}  
  
stage('Trivy') {  
  steps {  
    sh "trivy image --format table -o trivy-image-  
report.html venkatesh09/boardgame:latest"  
  }  
}
```

```
stage('Docker Push Image') {  
    steps {  
        script {  
            withDockerRegistry(credentialsId: 'docker-cred',  
toolName: 'docker') {  
                sh "docker push  
venkatesh09/boardgame: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 deployment-service.yaml -n  
webapps"  
            sh "kubectl get pods -n webapps"  
            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>
```

```
""""
```

```
emailtext (
```

```
    subject: "${jobName} - Build ${buildNumber} -
${pipelineStatus.toUpperCase()}",
```

```
    body: body,
```

```
    to: 'singareddyv91@gmail.com',
```

```
    from: 'jenkins@example.com',
```

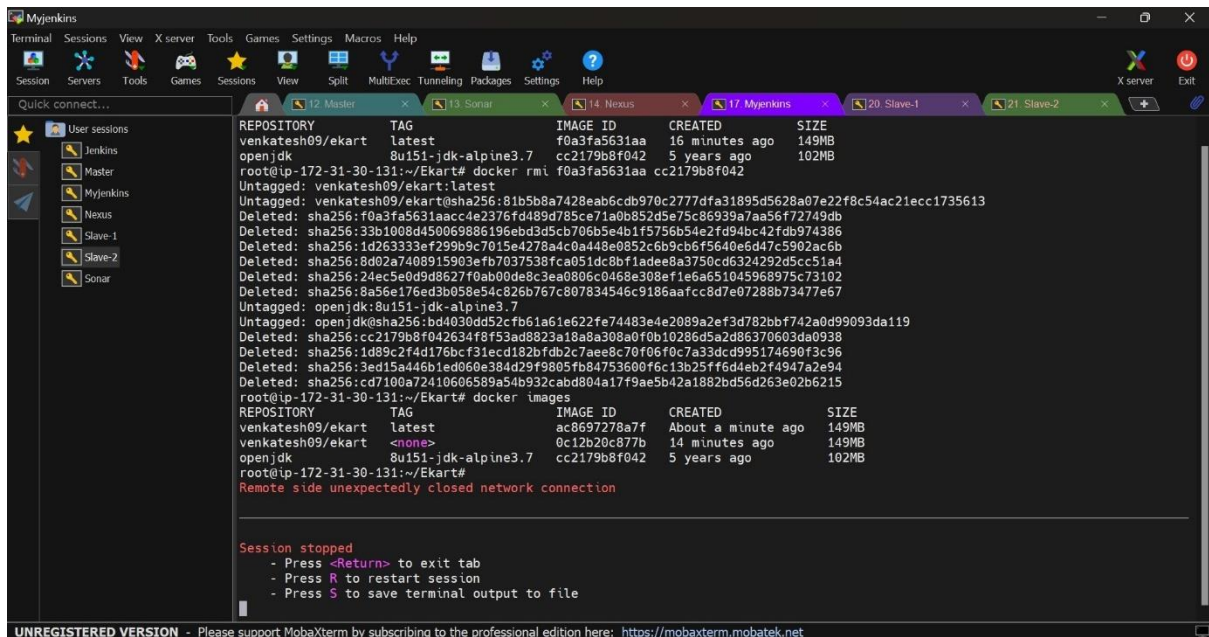
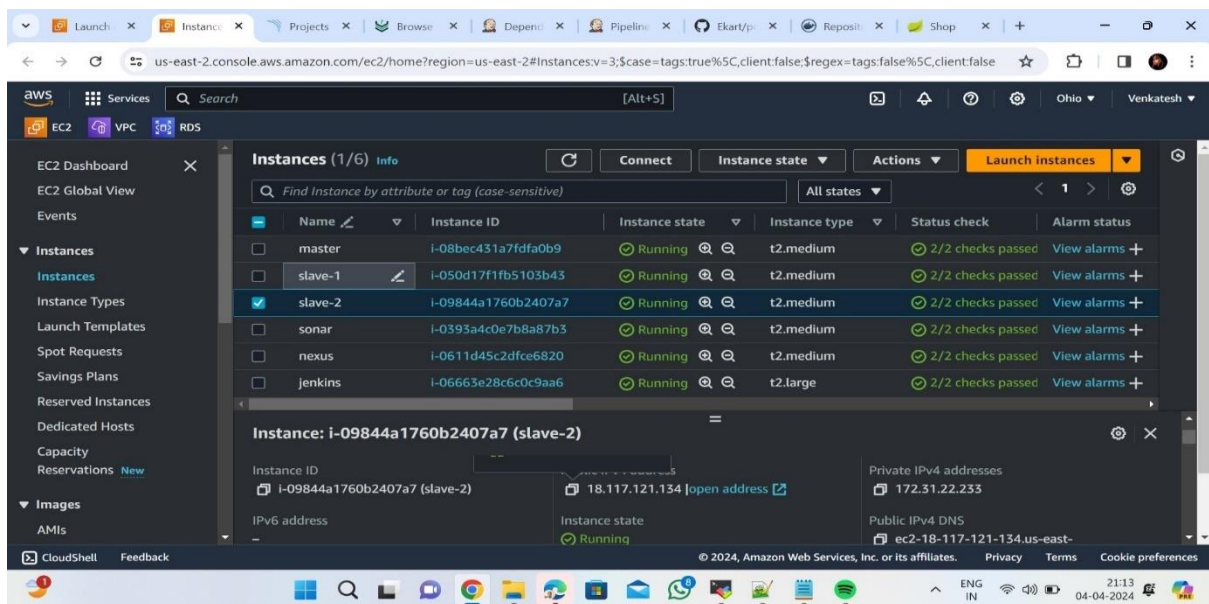
```
    replyTo: 'jenkins@example.com',
```

```
    mimeType: 'text/html',
```

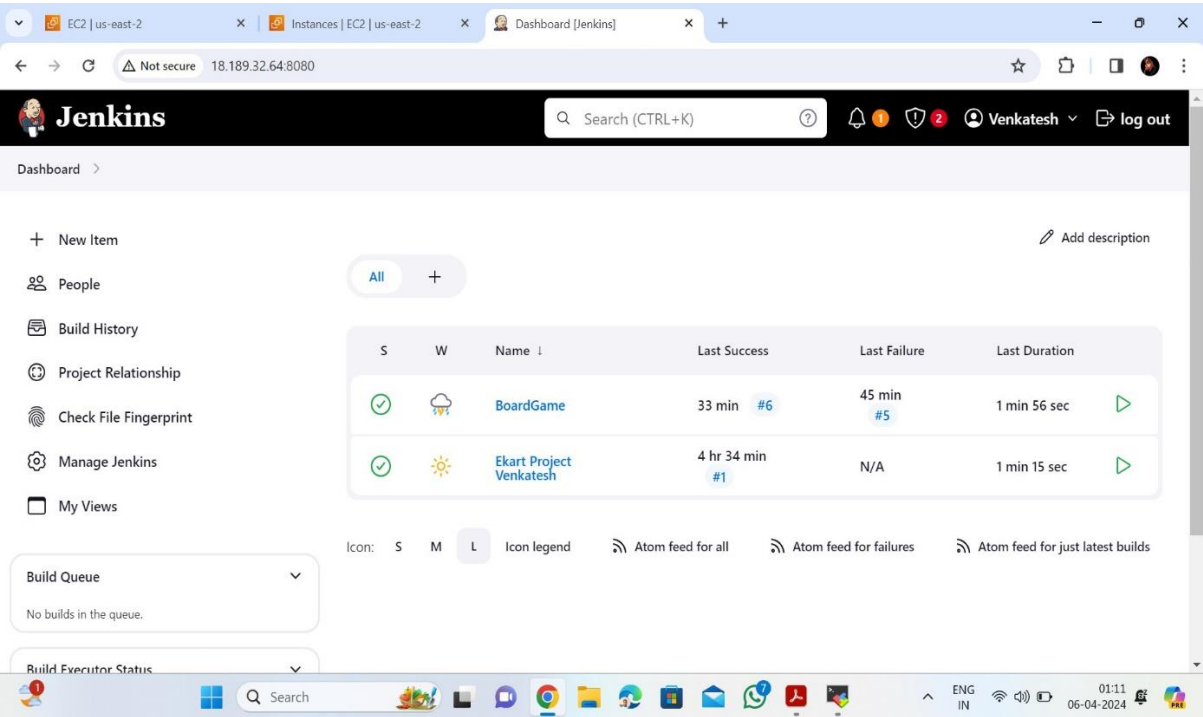
```
    attachmentsPattern: 'trivy-image-report.txt'
```

```
)
```

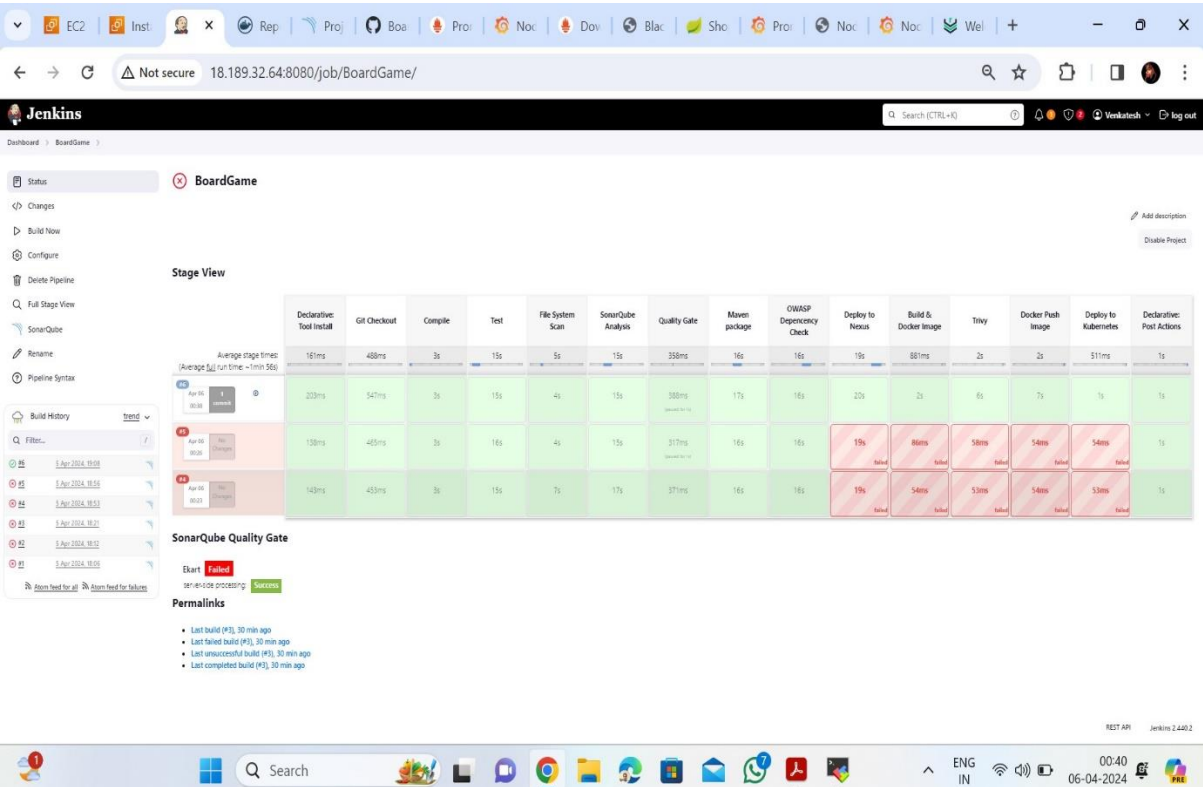
Launch EC2 Instances:



Created A NEW JOB:



Successfully Runs the Pipeline:



Console Output:

[illegible]

```
Dashboard > BoardGame > #6

[Pipeline] sh
+ kubectl get pods -n webapps

NAME                                READY   STATUS    RESTARTS   AGE
boardgame-deployment-5d6f95fd46-54bt2 0/1     ContainerCreating 0          1s
boardgame-deployment-5d6f95fd46-917rb 0/1     ContainerCreating 0          1s
ekart-deployment-5f67d8b0c4-16fsj     1/1     Running    1 (8h ago) 27h
ekart-deployment-5f67d8b0c4-q5k7c     1/1     Running    1 (8h ago) 27h

[Pipeline] sh
+ kubectl get svc -n webapps

NAME      TYPE        CLUSTER-IP      EXTERNAL-IP  PORT(S)          AGE
boardgame-svc LoadBalancer 10.106.100.94   <pending>    8080:10097/TCP   1s
ekart-svc   NodePort     10.100.67.1     <none>       8070:10241/TCP   27h

[Pipeline] }
[kubernetes-cli] kubectl configuration cleaned up
[Pipeline] // withKubeConfig
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] stage
[Pipeline] { [Declarative: Post Actions]
[Pipeline] script
[Pipeline] {
[Pipeline] emailText
Sending email to: singareddy91@gmail.com
[Pipeline] }
[Pipeline] // script
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // node
[Pipeline] end of Pipeline
Finished: SUCCESS
```

SonarQube Output:

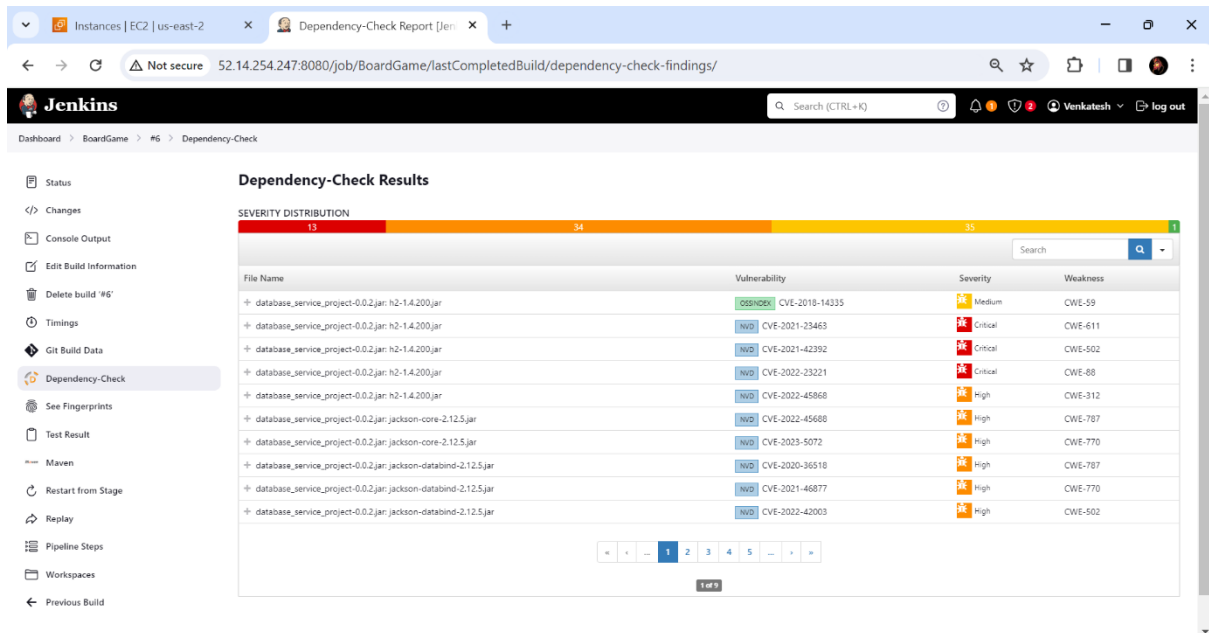
The screenshot displays the SonarQube web interface in a browser. The address bar shows the URL `52.14.19.17:9000/projects`. The interface includes a top navigation bar with tabs for Projects, Issues, Rules, Quality Profiles, Quality Gates, and Administration. A search bar is present on the right. On the left, there are filters for Quality Gate (Passed: 1, Failed: 0) and Reliability (A rating: 0, B rating: 0, C rating: 1, D rating: 0, E rating: 0). The main content area shows a search bar and a list of projects. The first project, 'Ekart', is highlighted with a 'Passed' status. Below the project name, a table displays various metrics: Bugs (25, C), Vulnerabilities (0, A), Hotspots Reviewed (0.0%, E), Code Smells (6, A), Coverage (0.0%, E), Duplications (0.0%, E), and Lines (1.3k, Java, HT...). A yellow warning box at the bottom states: '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.'

Nexus:

The screenshot displays the Sonatype Nexus Repository web interface in a browser. The address bar shows the URL `3.22.164.229:8081/#browse/browse:maven-releases`. The interface includes a top navigation bar with tabs for Browse, Search, and Upload. A search bar is present on the right. The main content area shows a tree view of the Maven releases. The tree structure is as follows:

- com
 - javaproject
 - database_service_project
 - 0.0.2
 - maven-metadata.xml
 - maven-metadata.xml.md5
 - maven-metadata.xml.sha1

Dependency Check:

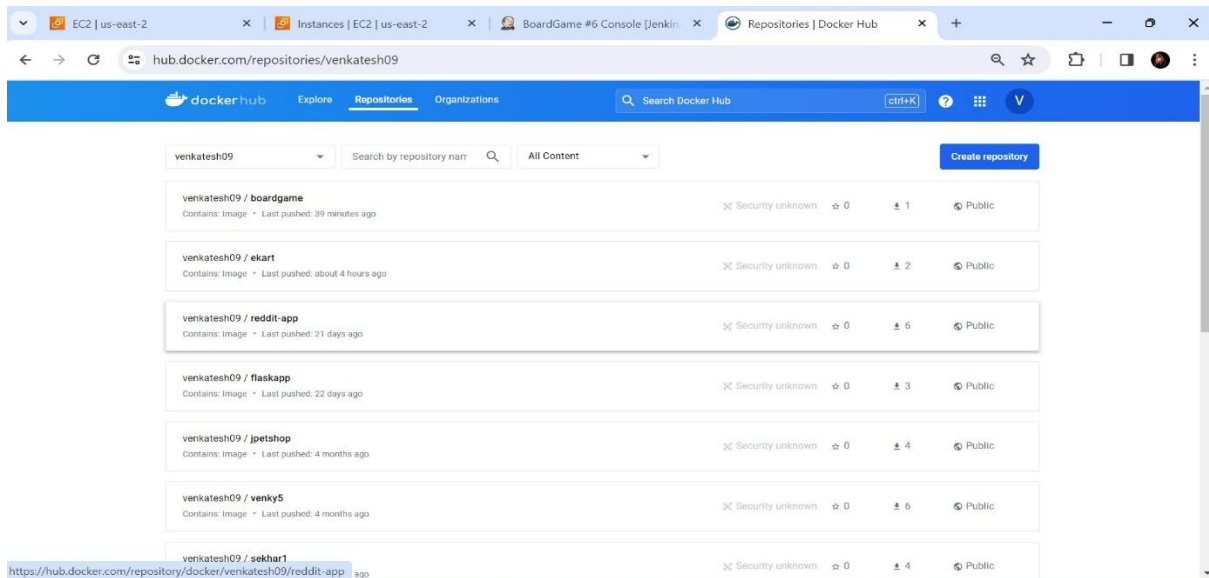


The screenshot shows the Jenkins web interface for a 'Dependency-Check' build. The left sidebar contains navigation links: Status, Changes, Console Output, Edit Build Information, Delete build '#6', Timings, Git Build Data, Dependency-Check (selected), See Fingerprints, Test Result, Maven, Restart from Stage, Replay, Pipeline Steps, Workspaces, and Previous Build. The main content area is titled 'Dependency-Check Results' and features a 'SEVERITY DISTRIBUTION' bar chart with 13 Critical, 34 High, and 35 Medium vulnerabilities. Below the chart is a table of findings.

File Name	Vulnerability	Severity	Weakness
+ database_service_project-0.0.2.jar:h2-1.4.200.jar	OSINDEX CVE-2018-14335	Medium	CWE-59
+ database_service_project-0.0.2.jar:h2-1.4.200.jar	NVD CVE-2021-23463	Critical	CWE-611
+ database_service_project-0.0.2.jar:h2-1.4.200.jar	NVD CVE-2021-42392	Critical	CWE-502
+ database_service_project-0.0.2.jar:h2-1.4.200.jar	NVD CVE-2022-23221	Critical	CWE-88
+ database_service_project-0.0.2.jar:h2-1.4.200.jar	NVD CVE-2022-45868	High	CWE-312
+ database_service_project-0.0.2.jar:jackson-core-2.12.5.jar	NVD CVE-2022-45688	High	CWE-787
+ database_service_project-0.0.2.jar:jackson-core-2.12.5.jar	NVD CVE-2023-5072	High	CWE-770
+ database_service_project-0.0.2.jar:jackson-databind-2.12.5.jar	NVD CVE-2020-36518	High	CWE-787
+ database_service_project-0.0.2.jar:jackson-databind-2.12.5.jar	NVD CVE-2021-46877	High	CWE-770
+ database_service_project-0.0.2.jar:jackson-databind-2.12.5.jar	NVD CVE-2022-42003	High	CWE-502

At the bottom of the table is a pagination control showing '1 of 9' results.

Docker:

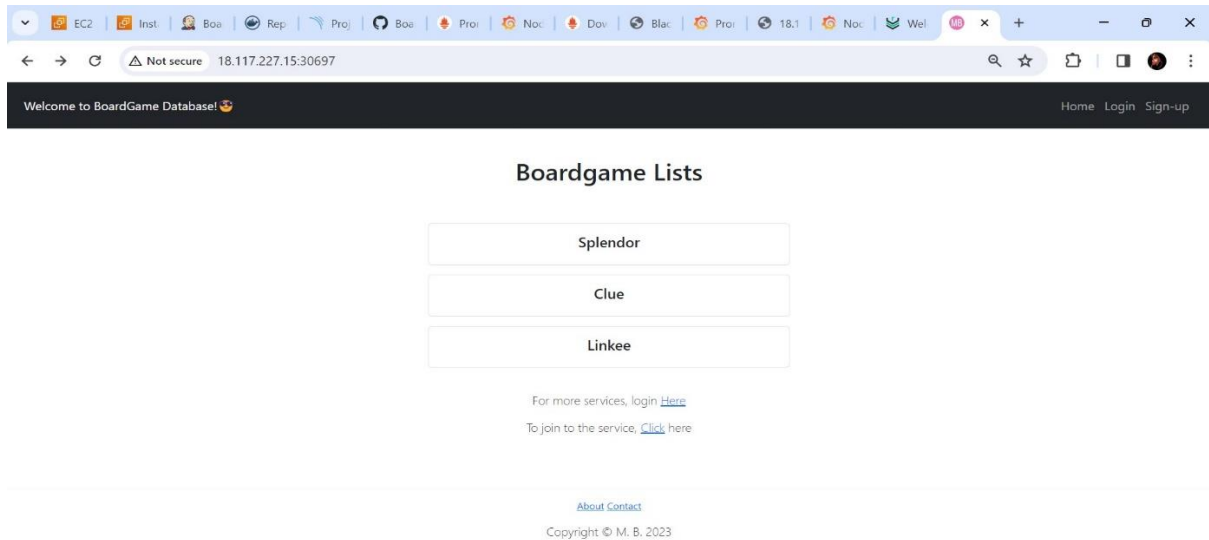


The screenshot shows the Docker Hub repository page for the user 'venkatesh09'. The page lists several public Docker images:

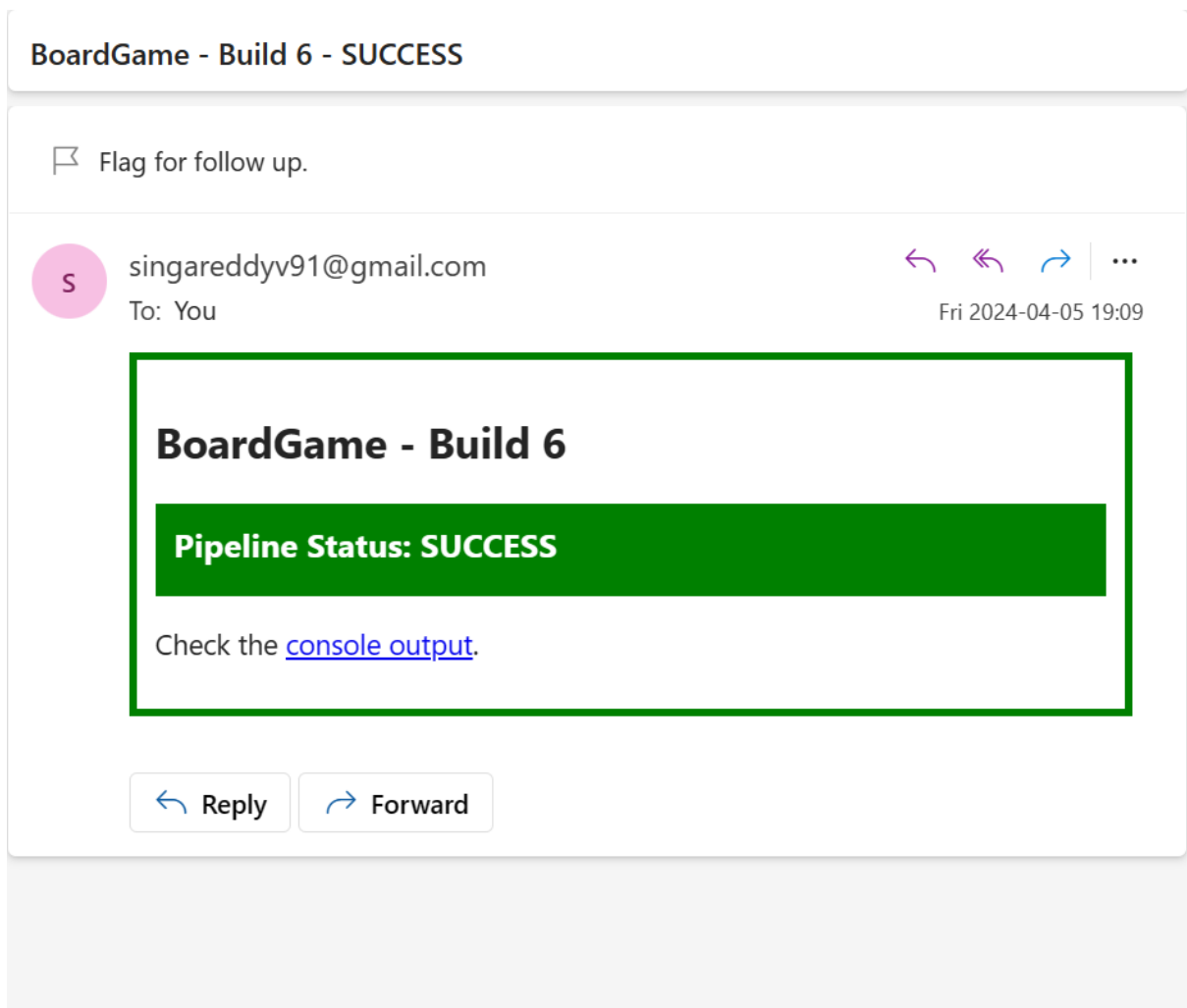
- venkatesh09 / boardgame**: Contains: Image • Last pushed: 39 minutes ago • Security unknown • 0 stars • 1 pull • Public
- venkatesh09 / ekart**: Contains: Image • Last pushed: about 4 hours ago • Security unknown • 0 stars • 2 pulls • Public
- venkatesh09 / reddit-app**: Contains: Image • Last pushed: 21 days ago • Security unknown • 0 stars • 6 pulls • Public
- venkatesh09 / flaskapp**: Contains: Image • Last pushed: 22 days ago • Security unknown • 0 stars • 3 pulls • Public
- venkatesh09 / jpetshop**: Contains: Image • Last pushed: 4 months ago • Security unknown • 0 stars • 4 pulls • Public
- venkatesh09 / venky5**: Contains: Image • Last pushed: 4 months ago • Security unknown • 0 stars • 6 pulls • Public
- venkatesh09 / sekhari**: Contains: Image • Last pushed: 4 months ago • Security unknown • 0 stars • 4 pulls • Public

The URL at the bottom of the page is <https://hub.docker.com/repository/docker/venkatesh09/reddit-app>.

Final Output:



Email Notification



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

Highly recommended 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