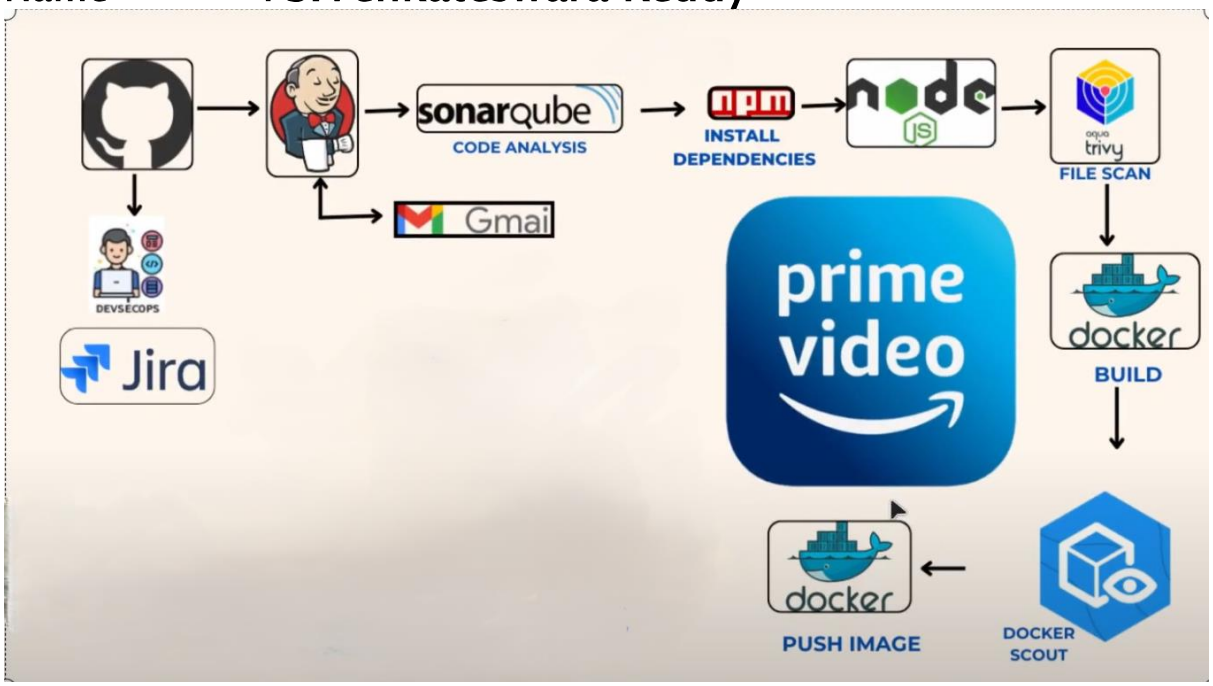
AWS Deployment Project:

Project Name: Starbucks Coffee Shop Project

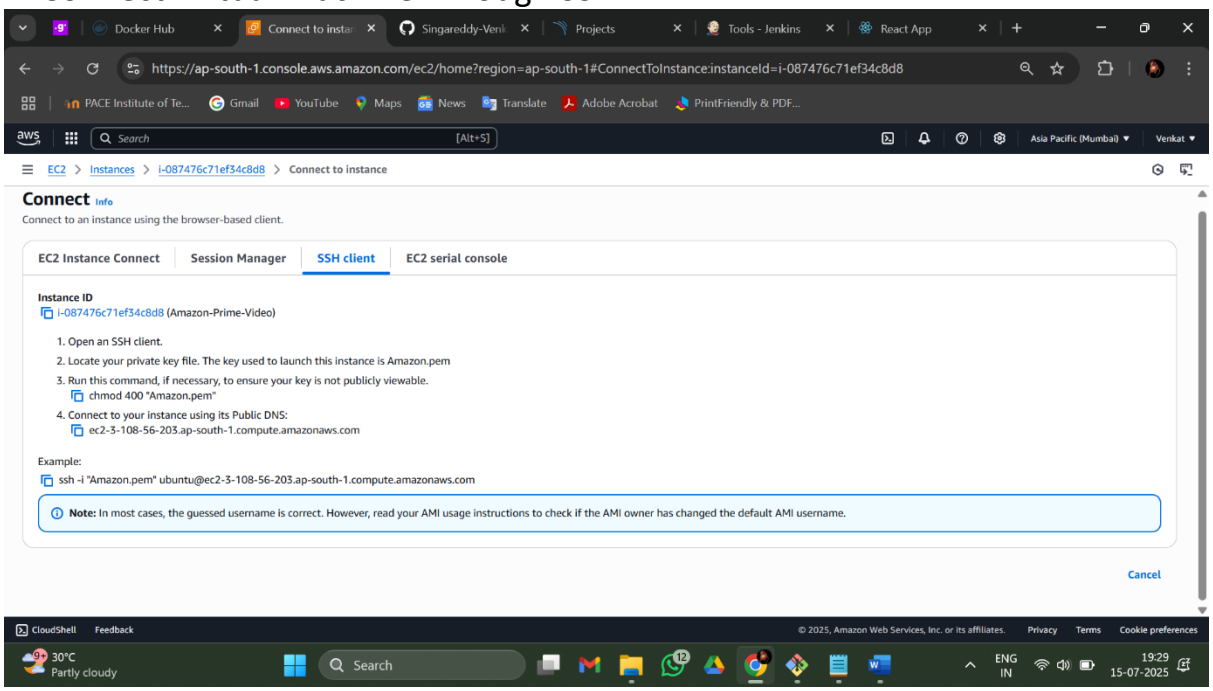
Name : S.Venkateswara Reddy



→Launch EC2 Instance with t2.xlarge

Security Group: Allow All Traffic (For Practice Purpose)

→Connect Virtual Machine Through SSH



→Connect through Git Bash or MobaXterm
→Sudo apt update -y
→Connect the server and git clone https://github.com/Singareddy-Venkatesh/starbucks-kubernetes.git

→cd scripts install all Packages Jenkins, Docker, SonarQube, Trivy, Docker Scout
→cat permissionexecut.sh (chmod +x *.sh)
→chmod +x permissionexecut.sh (All get Execute Permissions)
→sh permissionexecute.sh

Jenkins Packages:

```
#!/bin/bash
# jenkins installation on ubuntu
sudo apt update -y
sudo apt install fontconfig openjdk-17-jre -y
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 -y
sudo apt-get install jenkins -y
sudo systemctl enable jenkins
sudo systemctl start Jenkins
```

=>sh Jenkins.sh

⇨ <Public IP>8080 (You Will Open Jenkins Dashboard)
⇨ In Your Terminal (cat /var/lib/jenkins/secrets/initialAdminPassword)
Here generates a password for to Login into the Jenkins Dashboard and Install Plugins

Install Docker Packages:

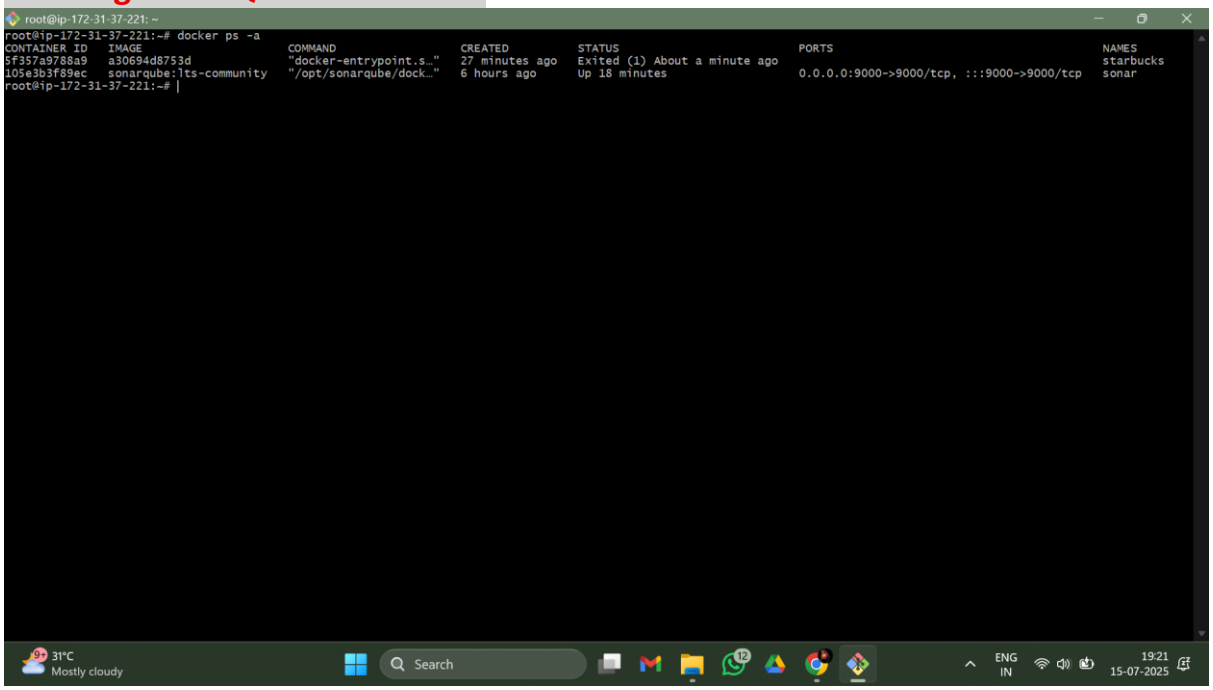
→Install docker (sh docker.sh) by apt install docker.io -y

```
#!/bin/bash
sudo apt-get update -y
sudo apt-get install docker.io -y
sudo usermod -aG docker ubuntu
```

sudo usermod -aG docker jenkins
newgrp docker
sudo chmod 660 /var/run/docker.sock
sudo chown root:docker /var/run/docker.sock
sudo systemctl restart docker
docker -version

→Install SonarQube (docker run --name sonar -d -p 9000:9000 sonarqube:lts-community

Running SonarQube Container:



→Install trivy (sh trivy.sh)

#!/bin/bash
# Install necessary dependencies
sudo apt-get install wget apt-transport-https gnupg lsb-release -y

Add the Trivy repository key
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key | gpg --dearmor | sudo tee /usr/share/keyrings/trivy.gpg > /dev/null

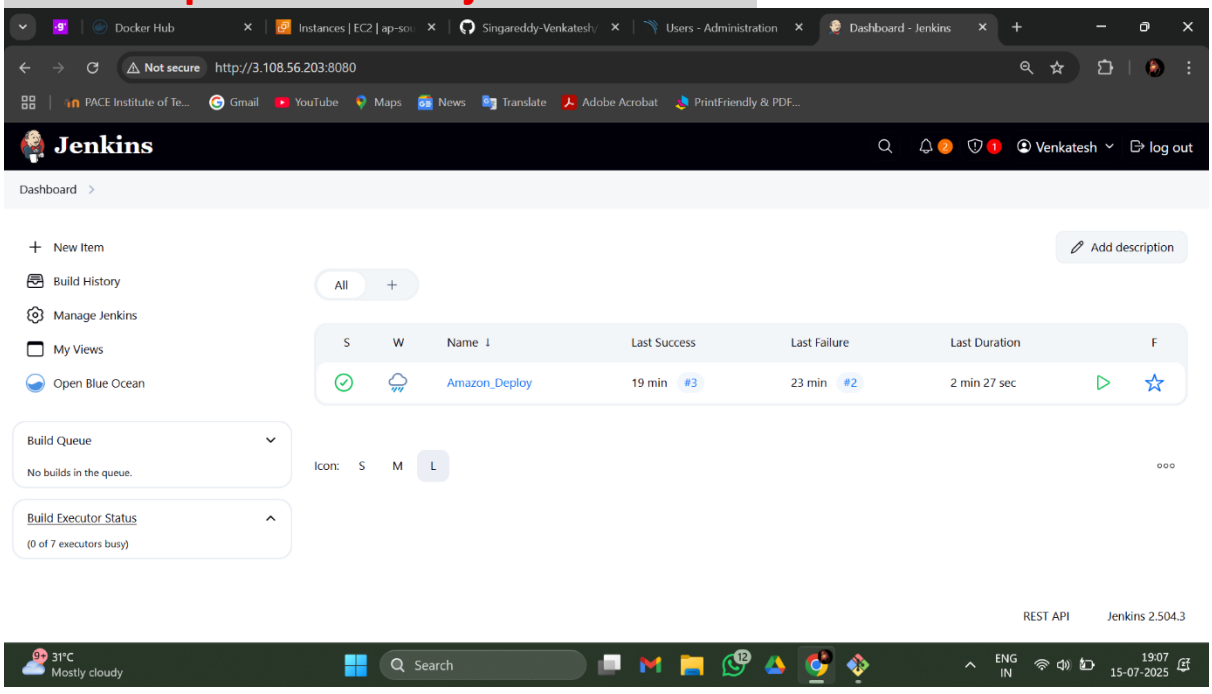
echo "deb [signed-by=/usr/share/keyrings/trivy.gpg] https://aquasecurity.github.io/trivy-repo/deb $(lsb_release -sc) main" | sudo tee -a /etc/apt/sources.list.d/trivy.list

```
sudo apt-get update -y
sudo apt-get install trivy -y
```

In Jenkins Install Plugins:

Eclipse Temurin Installer
SonarQube Scanner
Sonar Quality Gates
Pipeline Stage View
NodeJs
Docker
Docker Compose
Docker Pipeline
Docker API
Kubernetes
Kubernetes Client API
Kubernetes Credentials
Kubernetes CLI
Kubernetes Credentials Provider
Kubernetes:Pipeline:DevOps Steps
Blue Ocean (We Start Pipeline in Different Stages)

Create Pipeline Job Project in Jenkins:



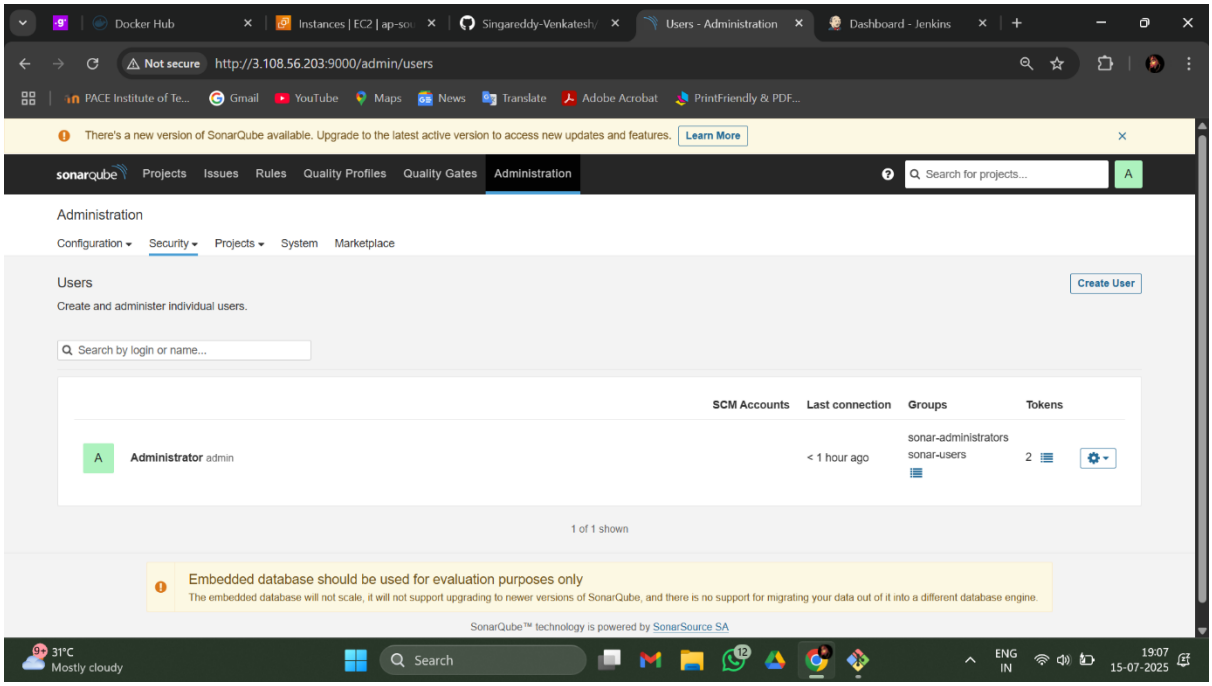
⇨ In Browser <Public IP>:9000 You can Open SonarQube

⇨ Username: admin

⇨ Password: admin

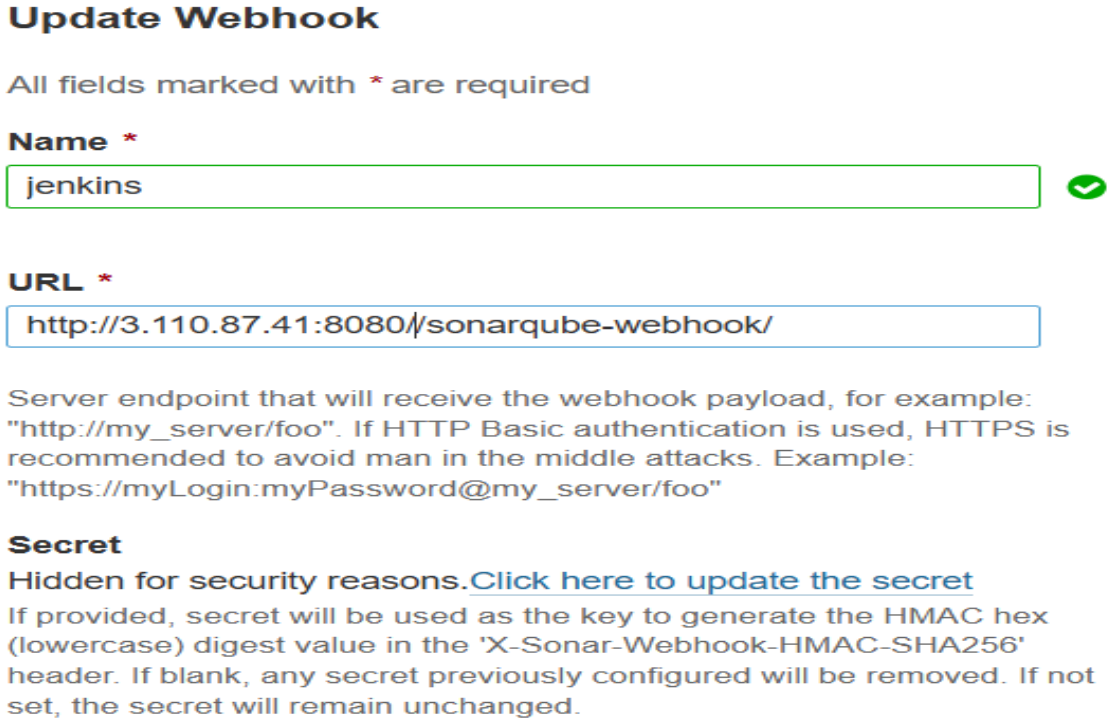
Integrate SonarQube with Jenkins:

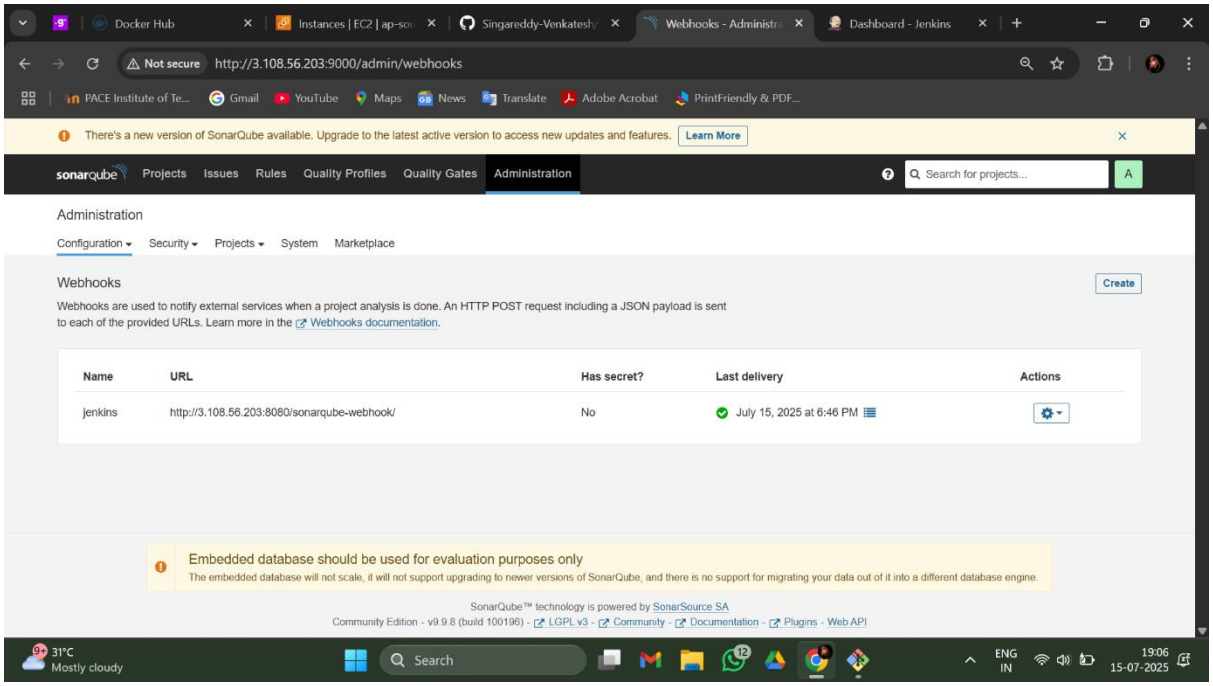
In SonarQube →Go to Administrator →User →Create Token



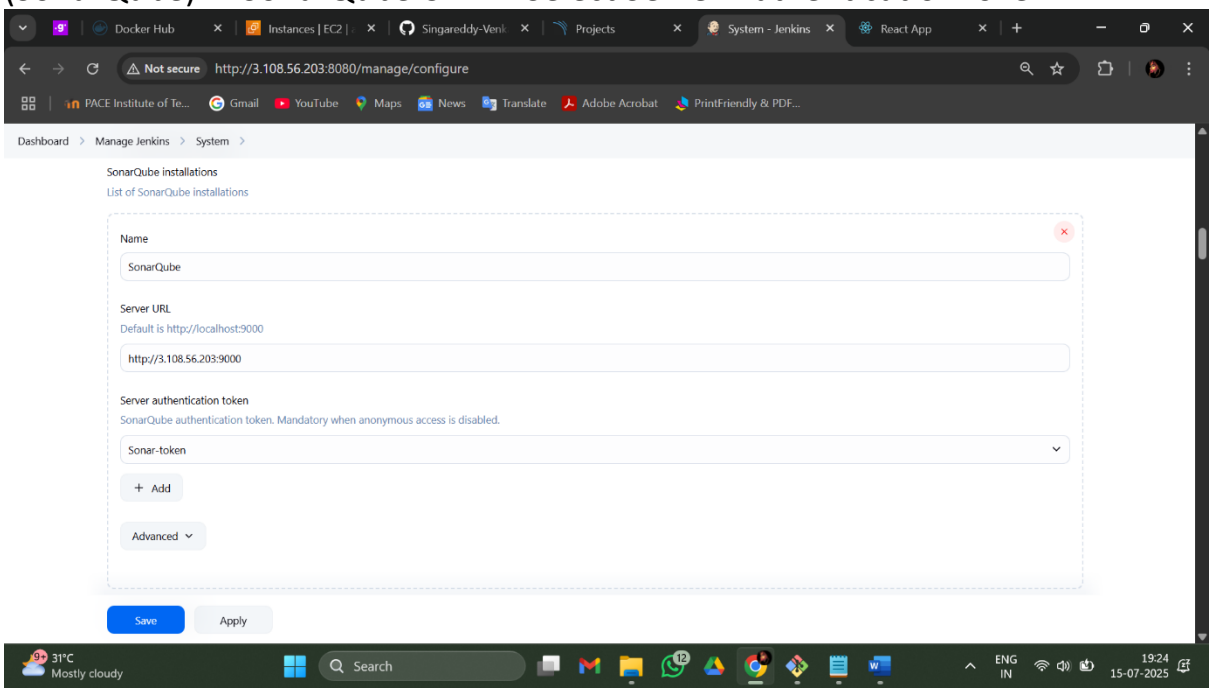
⇨ In Configuration →Webhook→Name (Jenkins)→ <JenkinsURL>sonarqube-webhooks/ →Create

Ex: http://3.108.56.203:8080/sonarqube-webhook/



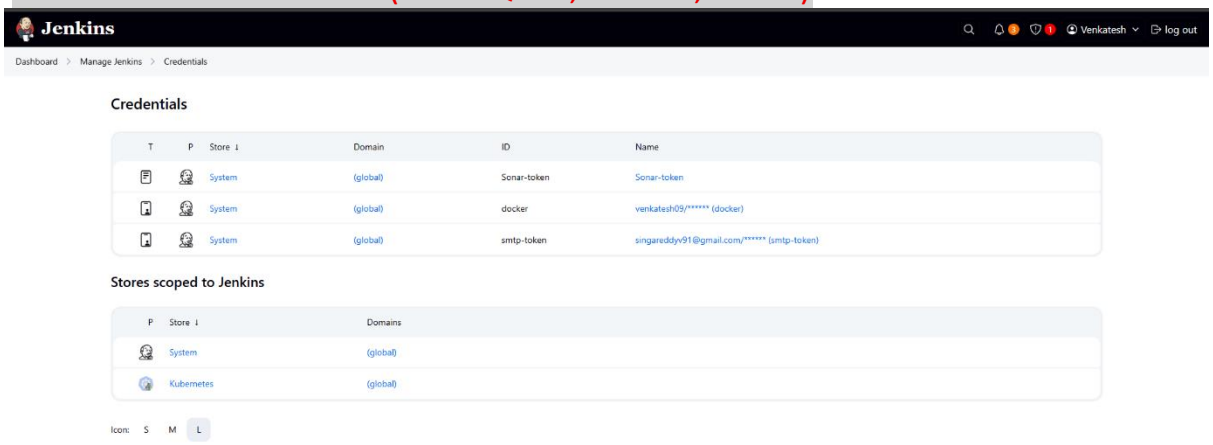


⇨ To Integrate SonarQube with Jenkins →Go to Manage Jenkins →System →SonarQube Server →Select SonarQube Installations →Name (SonarQube) →SonarQube URL →Select Server AuthenticationToken



⇨ In Jenkins→Go to Credentials →In Global →Select Secret Text →Paste Token →Give Same Name As per the Pipeline (Sonar-token) →Create

⇨ Create Credentials of (SonarQube, Docker, Gmail)



⇨ In Jenkins →In tools →SonarQube Scanner Installations→sonar-scanner→Select SonarQube Scanner 7.1.0.4889 →Apply

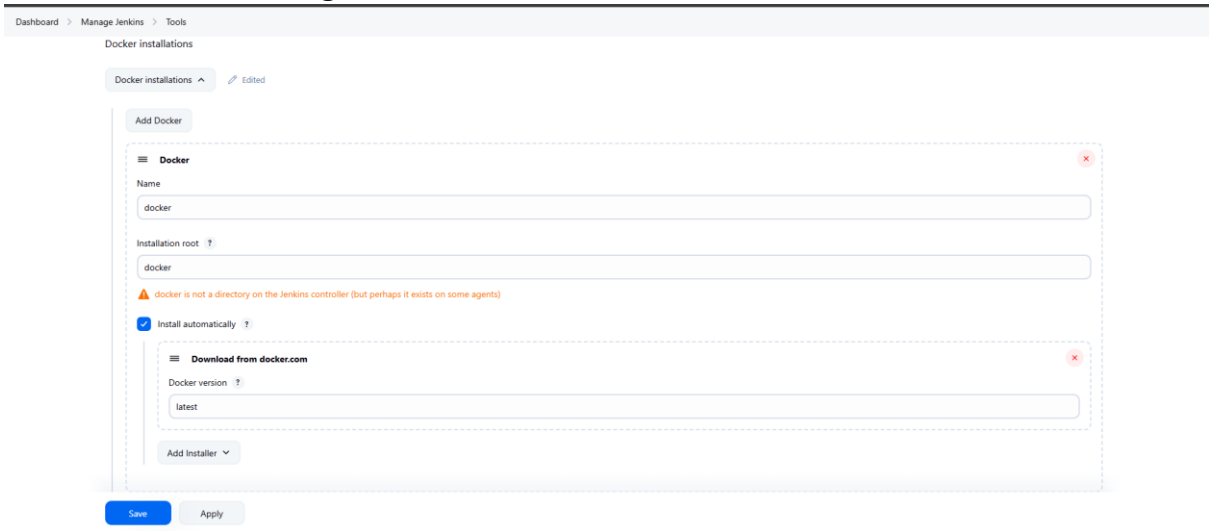
⇨ **Docker Credentials in Jenkins:**

→ Go to Manage Jenkins →System→Global →Select Username & Password →Give Docker Username & Password →Give Same Name As per the Pipeline (docker) →Create

→**Install Docker-Scout** →go to Terminal docker login with Username &Password Install

```
curl -sSfL https://raw.githubusercontent.com/docker/scout-cli/main/install.sh | sh -s -- -b /usr/local/bin
sh install-scout.sh
```

→In Jenkins →Manage Jenkins →tools→Select Docker Installations



⇨ In Jenkins →Go to Manage Jenkins → Tools →JDK Installations →jdk →Select Install from Adoptium.net (Version jdk-17.0.1+12)

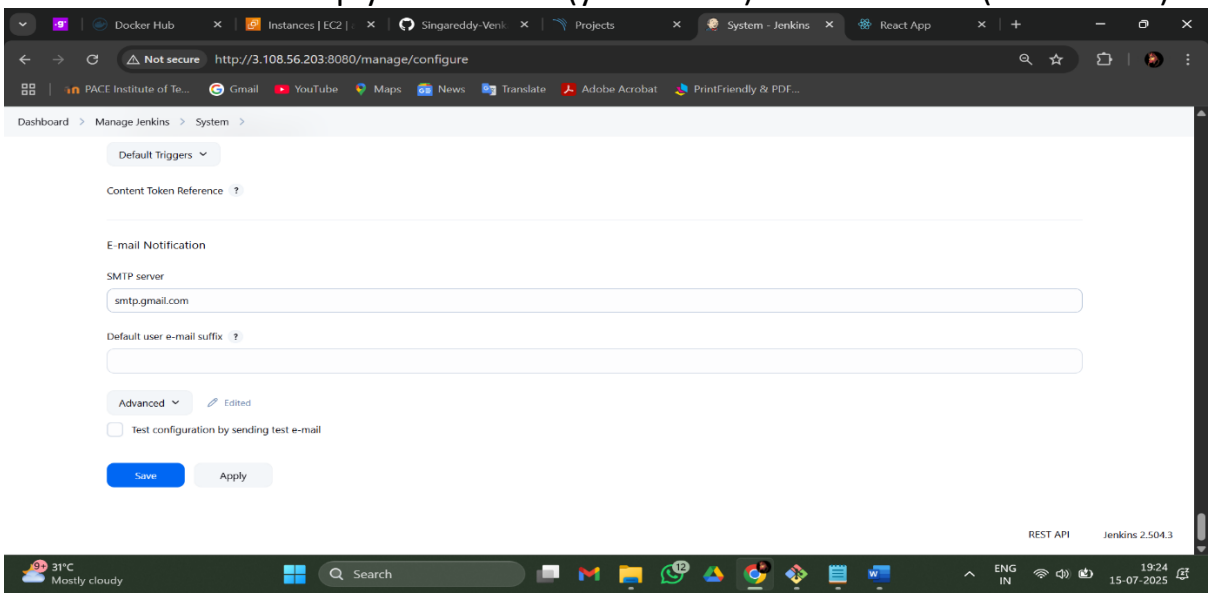
Nodejs Integrate with Jenkins:

⇨ In Jenkins→ Manage Jenkins→NodeJs Installations→node17 →Version(NodeJS 17.0.0)

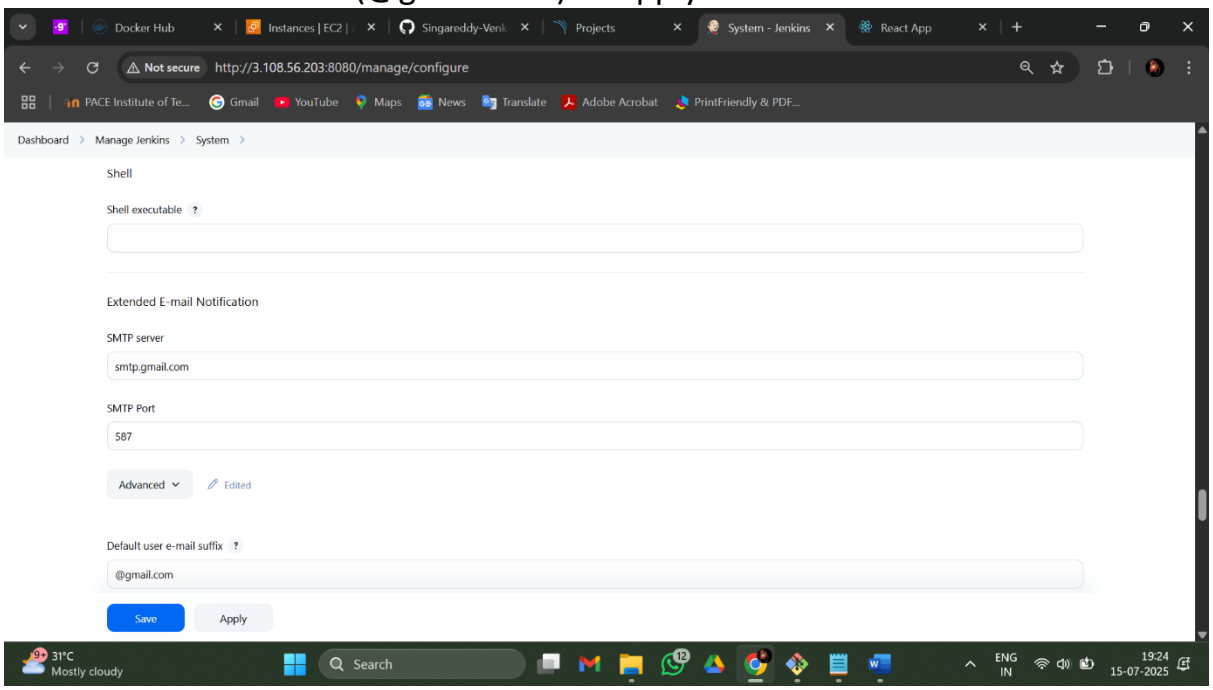
⇨ Email Notification →go to google Account →Manage Google → Security →Search App password →Create Token

⇨ After go to Jenkins Credentials →Username & Password →Your Gmail & Token which we created →ID (smtp-token) →Create

⇨ In Jenkins →Manage Jenkins →System → Email Notification →Name(smtp.gmail.com) →Default user email (@gmail.com) →Advanced →Use SMTP Authentication (Your Gmail) → Give Token →Select Use SSL → Port:465→Reply-to-Address (your Gmail) →select Email (Your Gmail)



⇨ In Jenkins→Manage Jenkins→Go to Extended E-mail Notification →smtp.gmail →Port:587 →Select Credentials →Select Use TLS →Default User email Suffix(@gmail.com) →Apply



→**In Jenkins write Declarative Pipeline**

Pipeline Syntax → Snippet Generator → Select Git →Paste the Git Repository →you will get the Generate Pipeline Script Paste into your Pipeline Groovy Script.

Jenkins Pipeline:

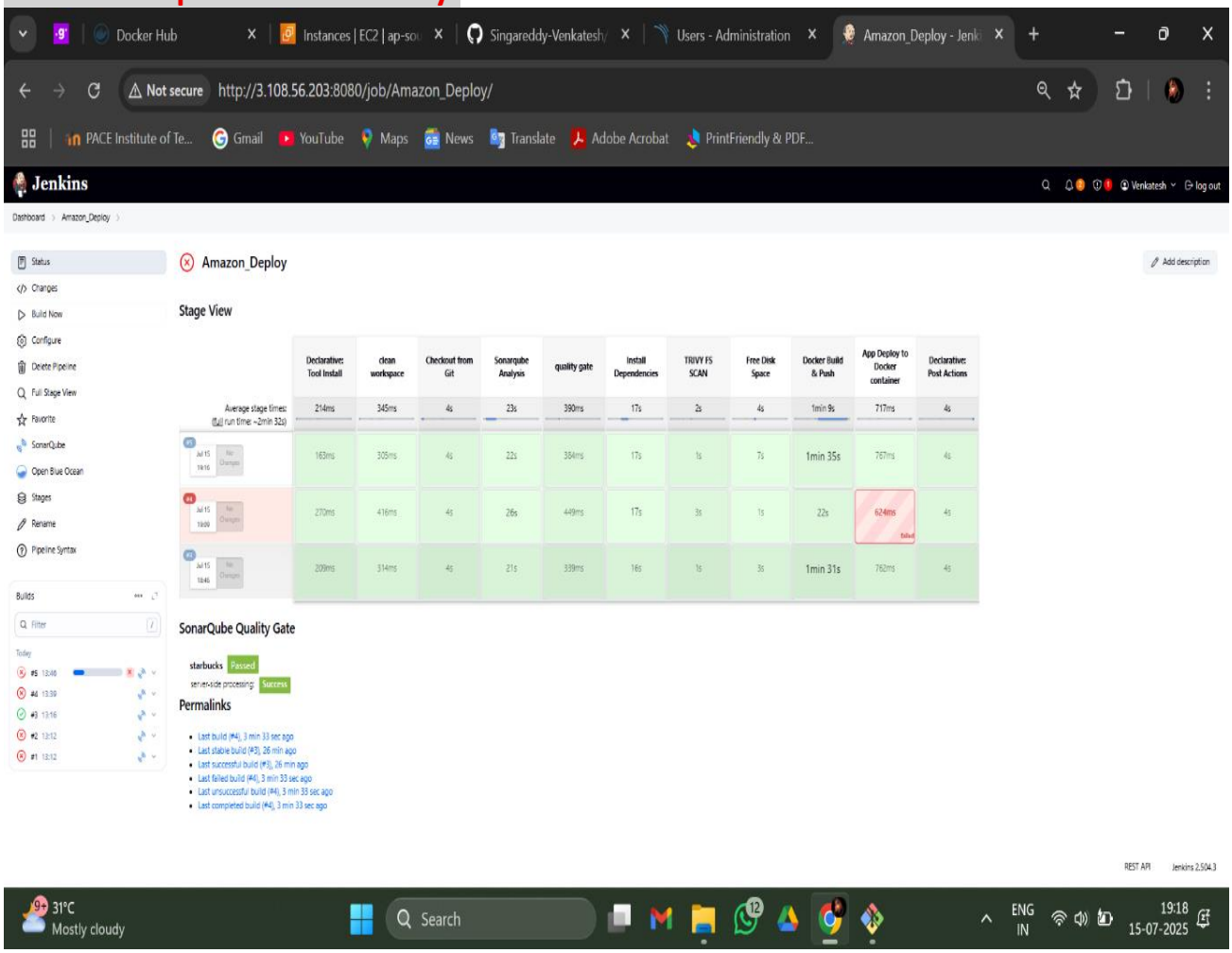
```
pipeline{
  agent any
  tools{
    jdk 'jdk'
    nodejs 'node17'
  }
  environment {
    SCANNER_HOME=tool 'sonar-scanner'
  }
  stages {
    stage('clean workspace'){
      steps{
        cleanWs()
```

```
        }
      }
      stage('Checkout from Git'){
        steps{
          git branch: 'main', url: 'https://github.com/Singareddy-
Venkatesh/starbucks-kubernetes.git'
        }
      }
      stage("Sonarqube Analysis "){
        steps{
          withSonarQubeEnv('SonarQube') {
            sh ''' $SCANNER_HOME/bin/sonar-scanner -
Dsonar.projectName=starbucks \
            -Dsonar.projectKey=starbucks '''
          }
        }
      }
       stage("quality gate"){
         steps {
          script {
            waitForQualityGate abortPipeline: false, credentialsId: 'Sonar-token'
          }
        }
      }
      stage('Install Dependencies') {
        steps {
          sh "npm install"
        }
      }
      stage('TRIVY FS SCAN') {
        steps {
          sh "trivy fs . > trivyfs.txt"
        }
      }
      stage('Free Disk Space') {
        steps {
          echo "Running cleanup to free disk space..."
          sh 'df -h'
          sh 'docker system prune -af || true'
          sh 'docker volume prune -f || true'
```

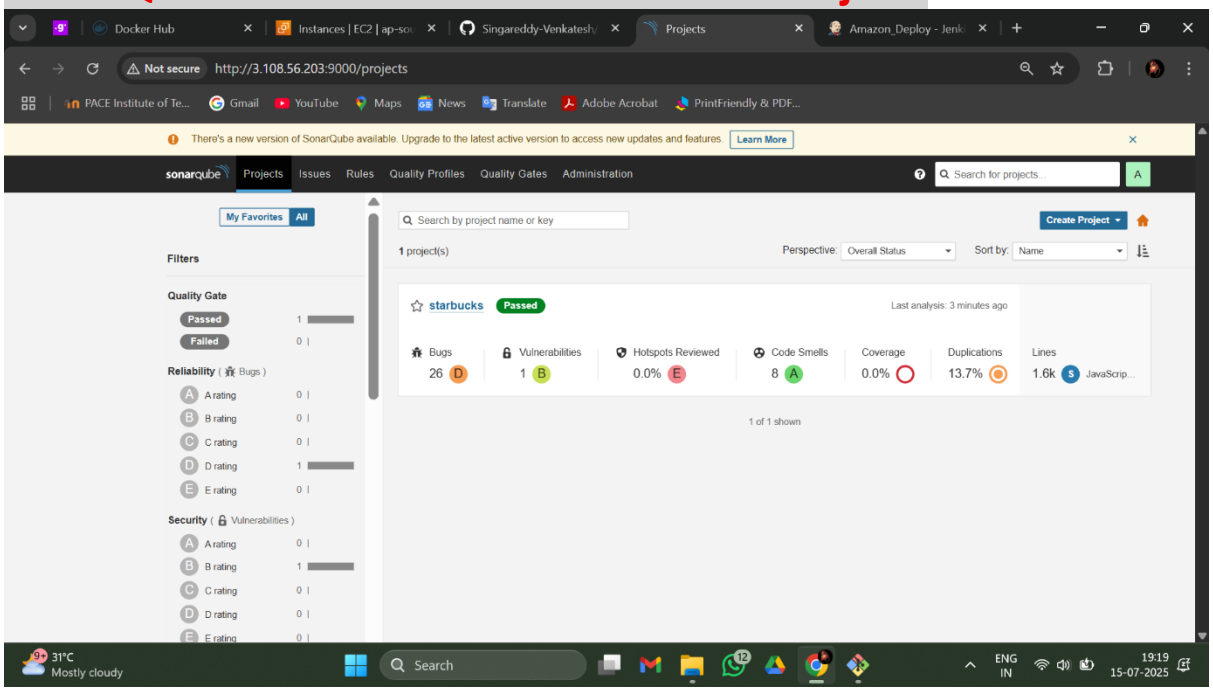
```
          // DON'T delete the current workspace manually here!
          sh 'df -h'
        }
      }
      stage("Docker Build & Push"){
        steps{
          script{
            withDockerRegistry(credentialsId: 'docker', toolName: 'docker') {
              sh "docker build -t starbucks ."
              sh "docker tag starbucks venkatesh09/starbucks:latest "
              sh "docker push venkatesh09/starbucks:latest "
            }
          }
        }
      }
      stage('App Deploy to Docker container'){
        steps{
          sh 'docker run -d --name starbucks -p 3000:3000
venkatesh09/starbucks:latest'
        }
      }

   }
   post {
   always {
      script {
        def buildStatus = currentBuild.currentResult
        def buildUser =
currentBuild.getBuildCauses('hudson.model.Cause$UserIdCause')[0]?.userId ?:
'Github User'

        emailext (
          subject: "Pipeline ${buildStatus}: ${env.JOB_NAME}
#${env.BUILD_NUMBER}",
          body: """
            <p>This is a Jenkins starbucks CI/CD pipeline status.</p>
            <p>Project: ${env.JOB_NAME}</p>
            <p>Build Number: ${env.BUILD_NUMBER}</p>
            <p>Build Status: ${buildStatus}</p>
            <p>Started by: ${buildUser}</p>
```

```
            <p>Build URL: <a
href="${env.BUILD_URL}">${env.BUILD_URL}</a></p>
            """,
            to: 'singaredyv91@gmail.com',
            from: 'singareddyv91@gmail.com',
            replyTo: 'singareddyv91@gmail.com',
            mimeType: 'text/html',
            attachmentsPattern: 'trivyfs.txt,trivyimage.txt'
        )
      }
    }

  }

}
```

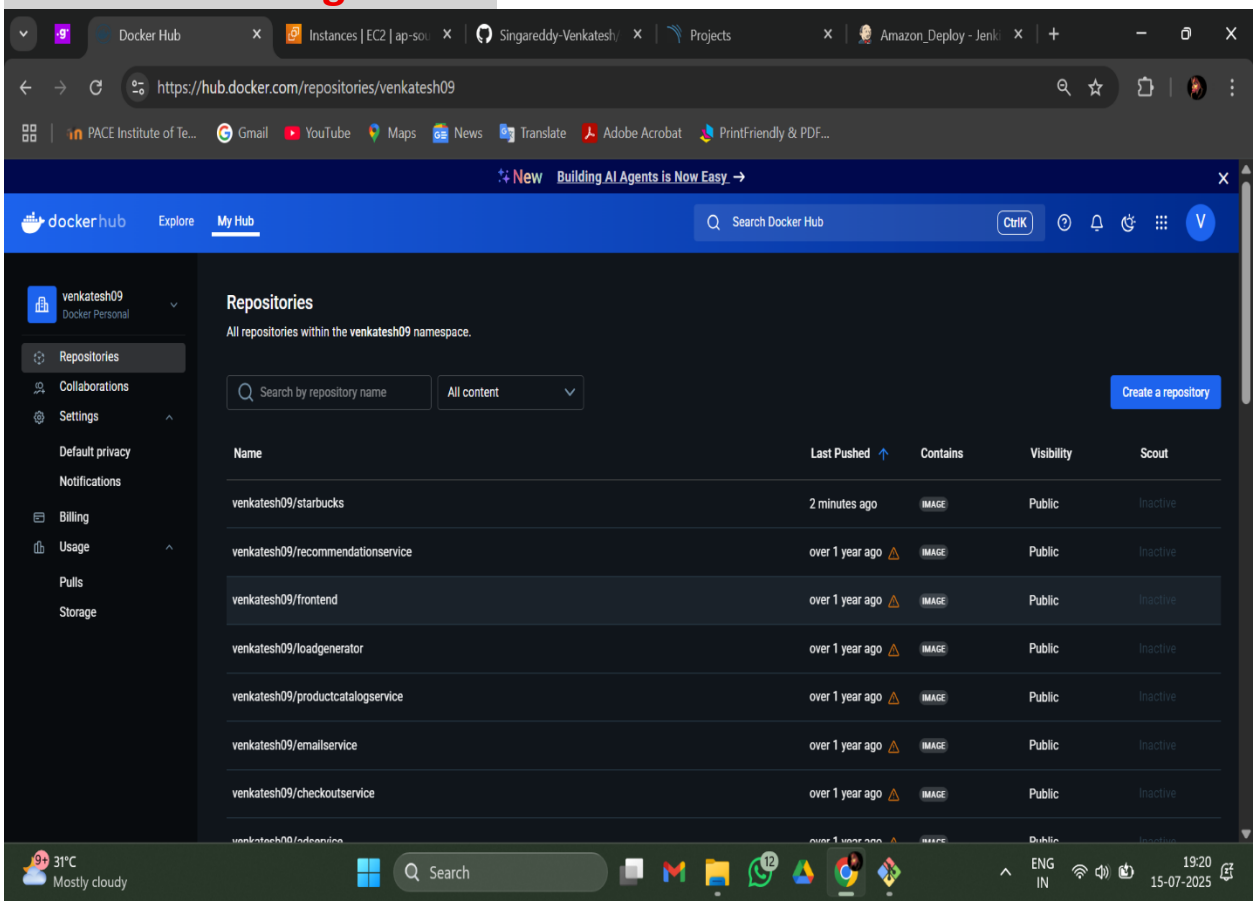
Build The Pipeline Successfully:



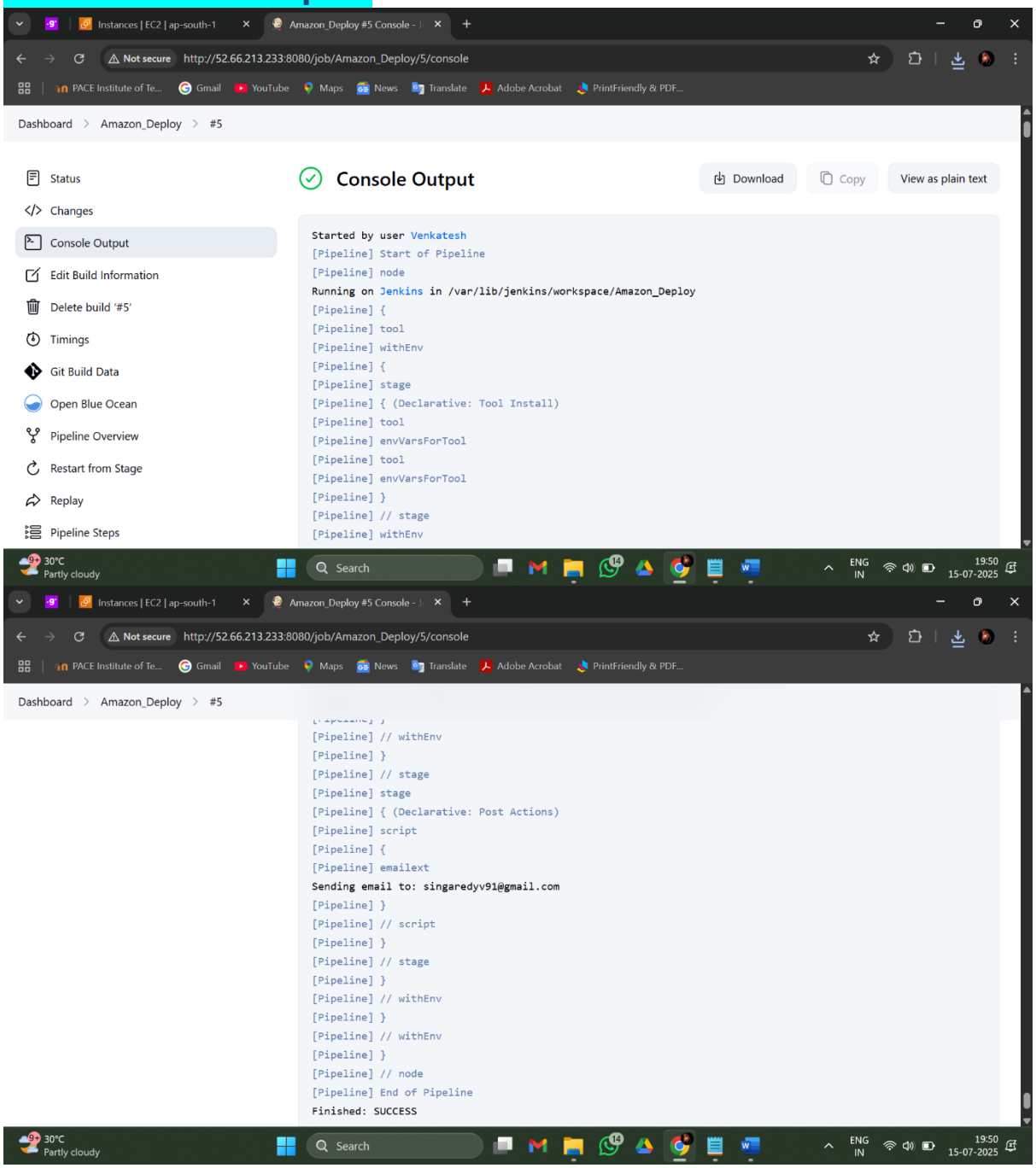
SonarQube Dashboard with Starbucks Project:



Docker Hub Image Build:

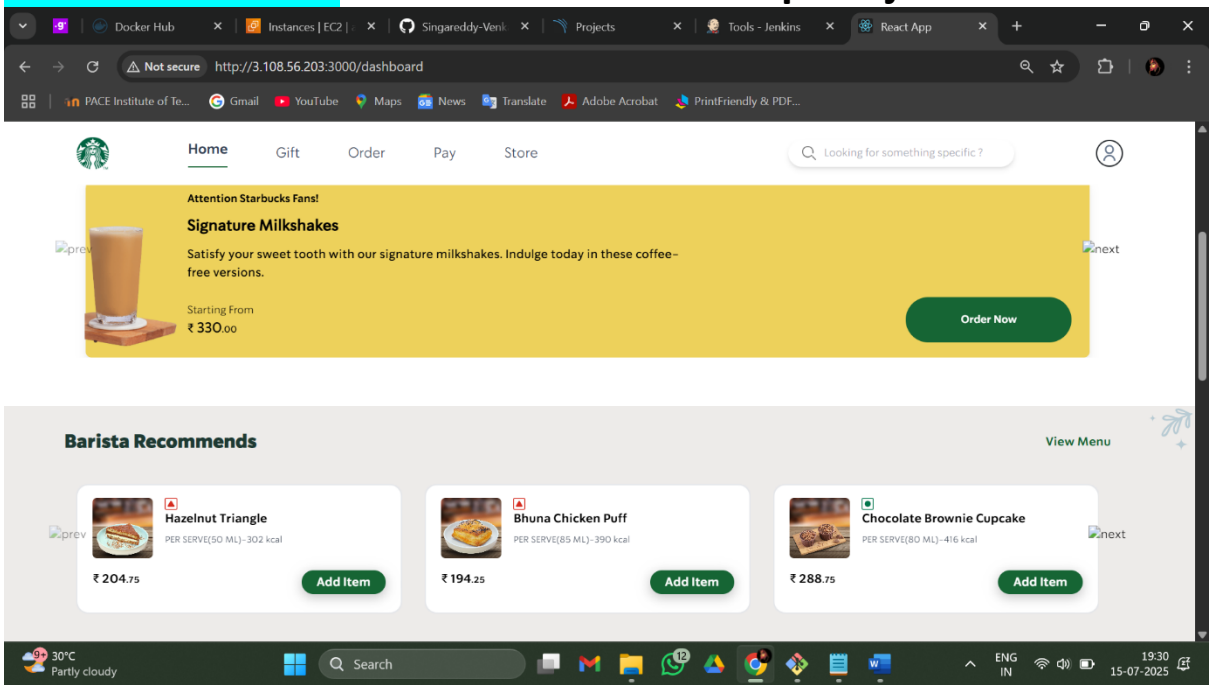


Console Final Output:



→Successfully Build All the Stages take <Public IP:3000> Check in your Browser its working or not and check the docker Hub Image →Check SonarQube →Email

Final Dashboard: Starbucks Coffee Shop Project



Email Notification Successfully Received: