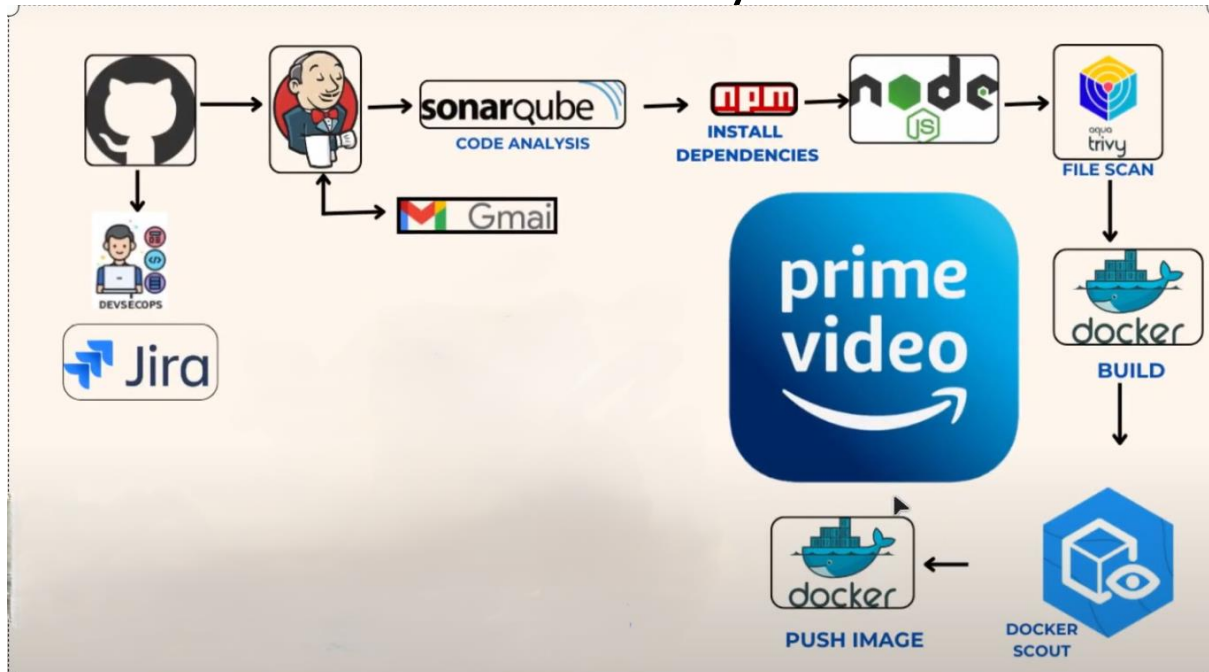


# AWS Deployment Project:

Project Name: **Starbucks Coffee Shop Project**

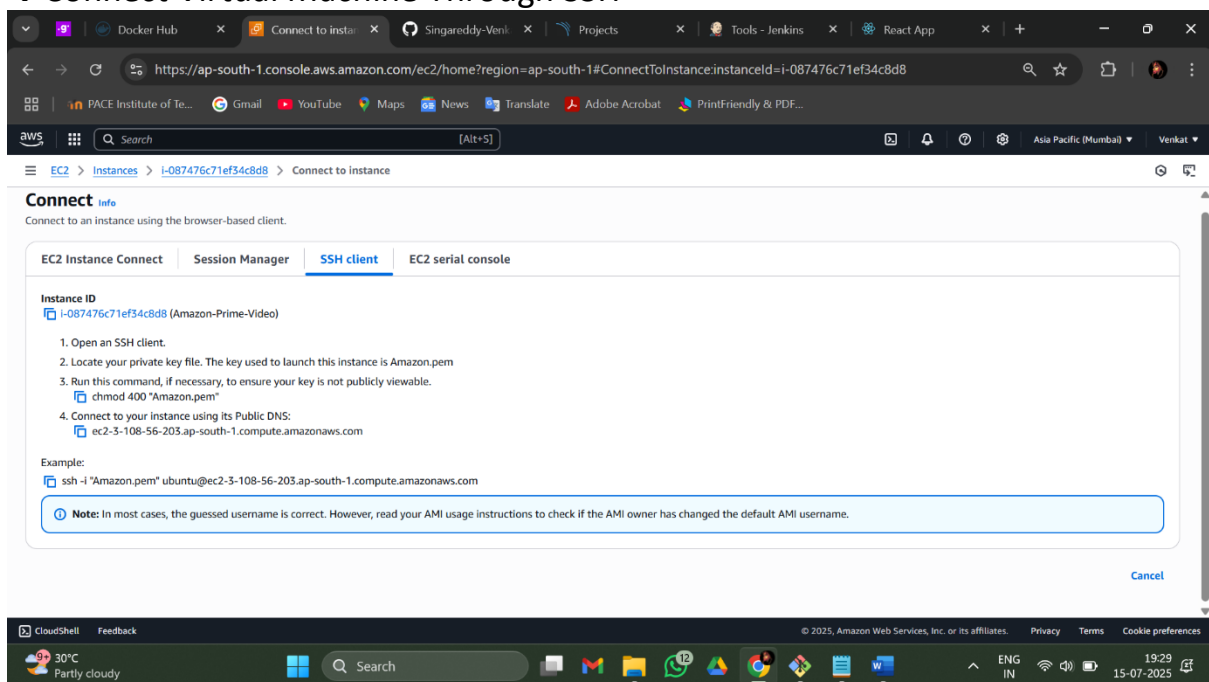
Name : S.Venkateswara Reddy



→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

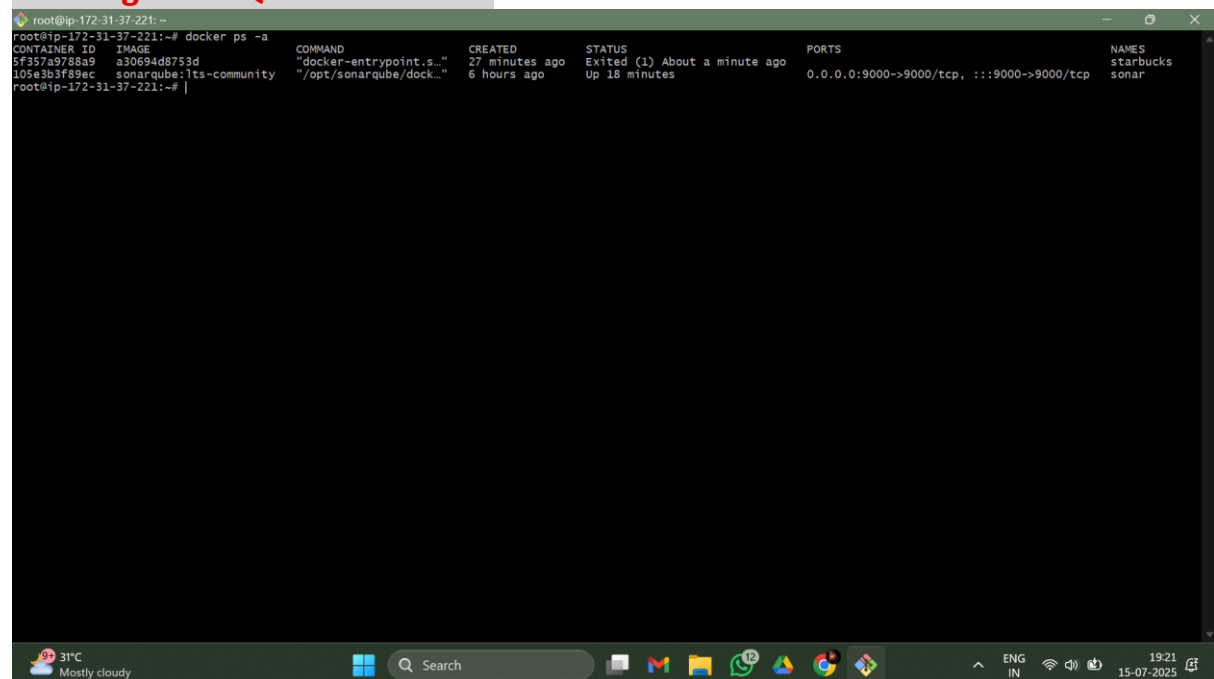
### Install Docker Packages:

→ 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:its-community`)

### Running SonarQube Container:



```
root@ip-172-31-37-221: ~
root@ip-172-31-37-221:~# docker ps -a
CONTAINER ID   IMAGE               COMMAND                  CREATED    STATUS      PORTS                               NAMES
5f357a9788a9   a30694d8753d       "docker-entrypoint.s..." 27 minutes ago  Exited (1) About a minute ago  0.0.0.0:9000->9000/tcp, :::9000->9000/tcp  starbucks
105e3b3f89ec   sonarqube:its-community  "/opt/sonarqube/dock..." 6 hours ago   Up 18 minutes                               sonar
```

→ **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:

The screenshot shows the Jenkins Dashboard in a web browser. The browser's address bar displays the URL `http://3.108.56.203:8080`. The Jenkins logo and name are visible at the top left of the dashboard. On the right side of the header, there are links for 'log out' and a user profile for 'Venkatesh'. The main content area features a sidebar on the left with navigation options: 'New Item', 'Build History', 'Manage Jenkins', 'My Views', and 'Open Blue Ocean'. The central part of the dashboard displays a table of pipeline jobs. The table has columns for 'S' (Success), 'W' (Warning), 'Name', 'Last Success', 'Last Failure', 'Last Duration', and 'F' (Failure). A single job, 'Amazon\_Deploy', is listed with a success status, a warning icon, a last success time of '19 min', a last failure time of '23 min', and a last duration of '2 min 27 sec'. Below the table, there are sections for 'Build Queue' (showing 'No builds in the queue') and 'Build Executor Status' (showing '0 of 7 executors busy'). At the bottom of the dashboard, there is a footer with 'REST API' and 'Jenkins 2.504.3'. The Windows taskbar is visible at the very bottom of the image, showing the system clock as 19:07 on 15-07-2025.

S	W	Name	Last Success	Last Failure	Last Duration	F
✓	⚠	Amazon_Deploy	19 min #3	23 min #2	2 min 27 sec	▶ ☆

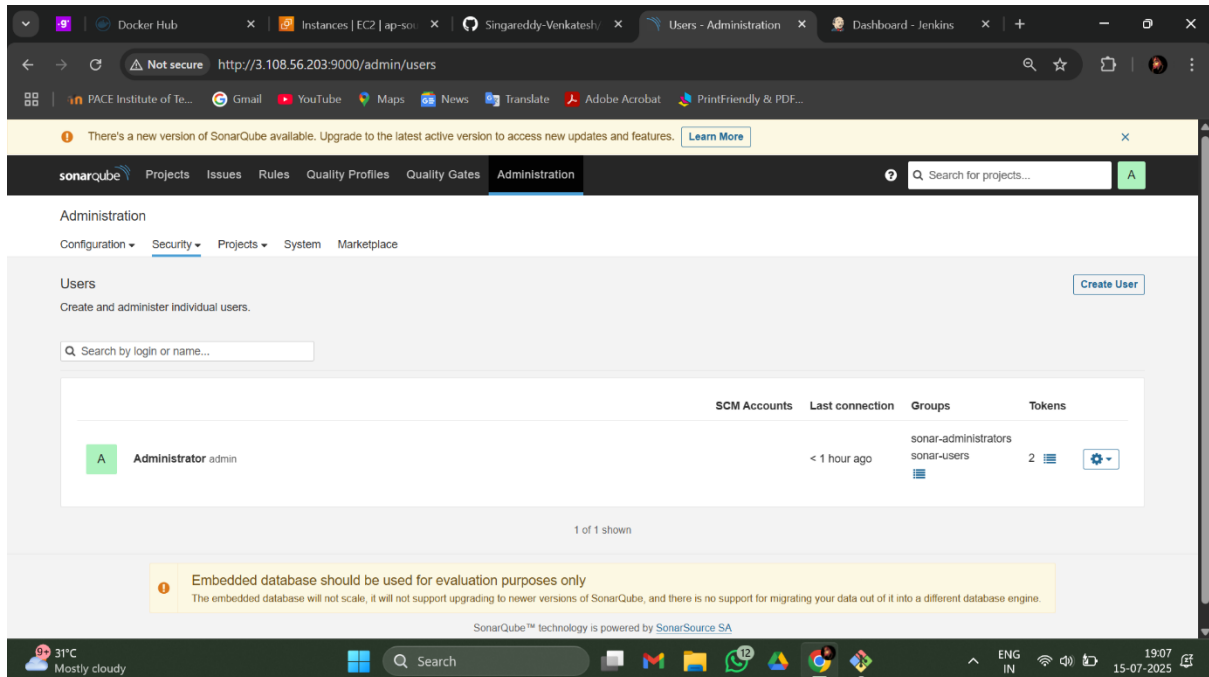
⇒ In Browser <Public IP>:9000 You can Open SonarQube

⇒ 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/>

### Update Webhook

All fields marked with \* are required

**Name \***

jenkins



**URL \***

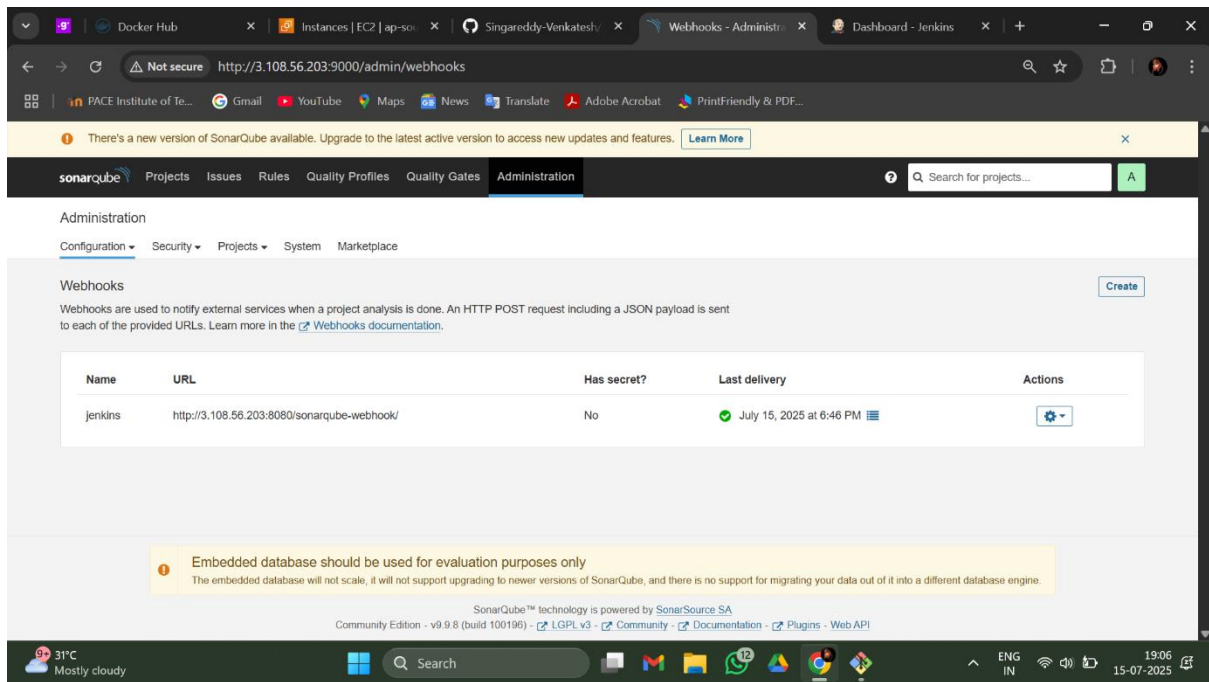
<http://3.110.87.41:8080/sonarqube-webhook/>

Server endpoint that will receive the webhook payload, for example: "http://my\_server/foo". If HTTP Basic authentication is used, HTTPS is recommended to avoid man in the middle attacks. Example: "https://myLogin:myPassword@my\_server/foo"

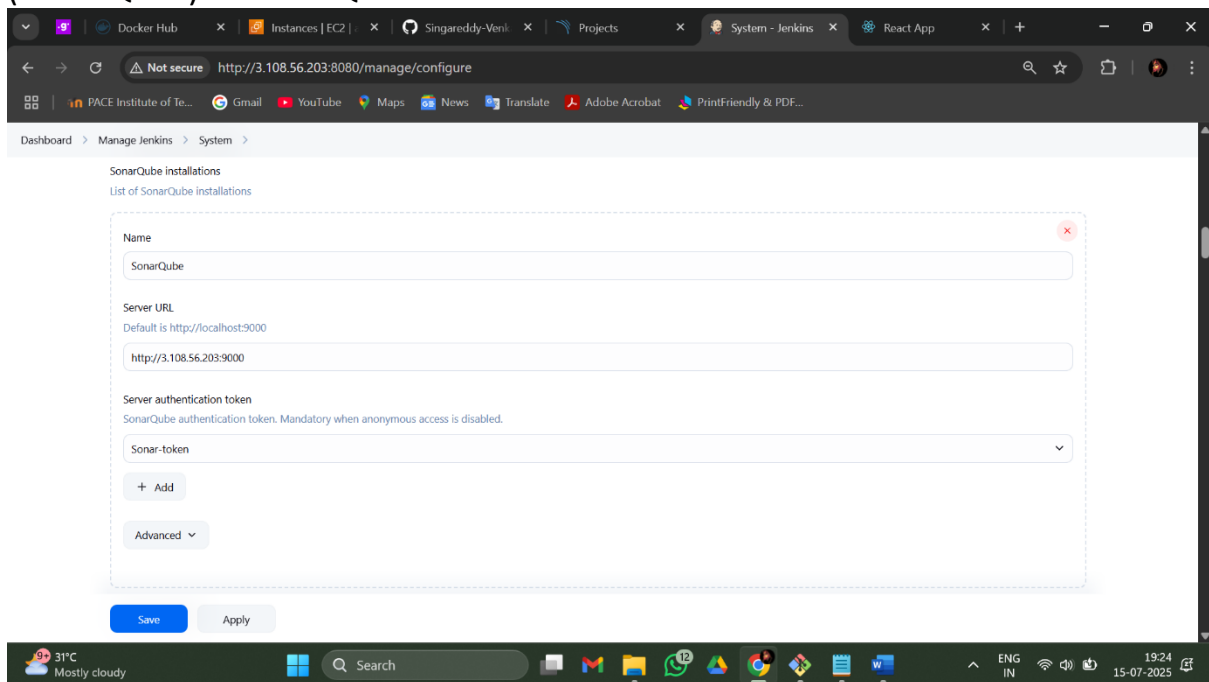
**Secret**

Hidden for security reasons. [Click here to update the secret](#)

If provided, secret will be used as the key to generate the HMAC hex (lowercase) digest value in the 'X-Sonar-Webhook-HMAC-SHA256' header. If blank, any secret previously configured will be removed. If not set, the secret will remain unchanged.

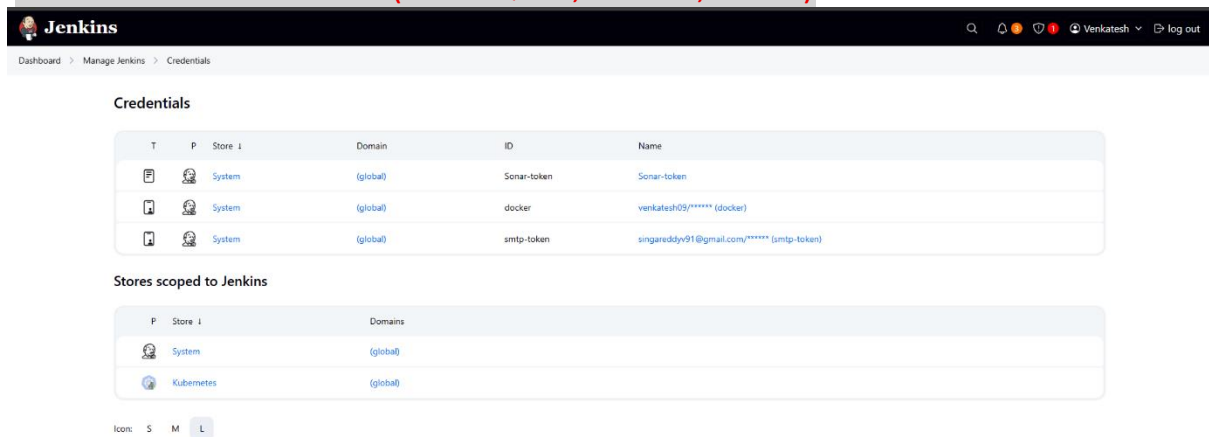


⇒ To Integrate SonarQube with Jenkins → Go to Manage Jenkins → System → SonarQube Server → Select SonarQube Installations → Name (SonarQube) → SonarQube URL → Select Server AuthenticationToken



⇒ 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)



The screenshot shows the Jenkins 'Credentials' page. At the top, there's a navigation bar with 'Dashboard', 'Manage Jenkins', and 'Credentials'. Below this, the 'Credentials' section displays a table of existing credentials:

T	P	Store	Domain	ID	Name
		System	(global)	Sonar-token	Sonar-token
		System	(global)	docker	venkatesh09/***** (docker)
		System	(global)	smtp-token	singareddy91@gmail.com/***** (smtp-token)

Below the table, the 'Stores scoped to Jenkins' section shows two stores:

P	Store	Domains
	System	(global)
	Kubernetes	(global)

At the bottom, there are filters for 'Icon', 'S', 'M', and 'L'.

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

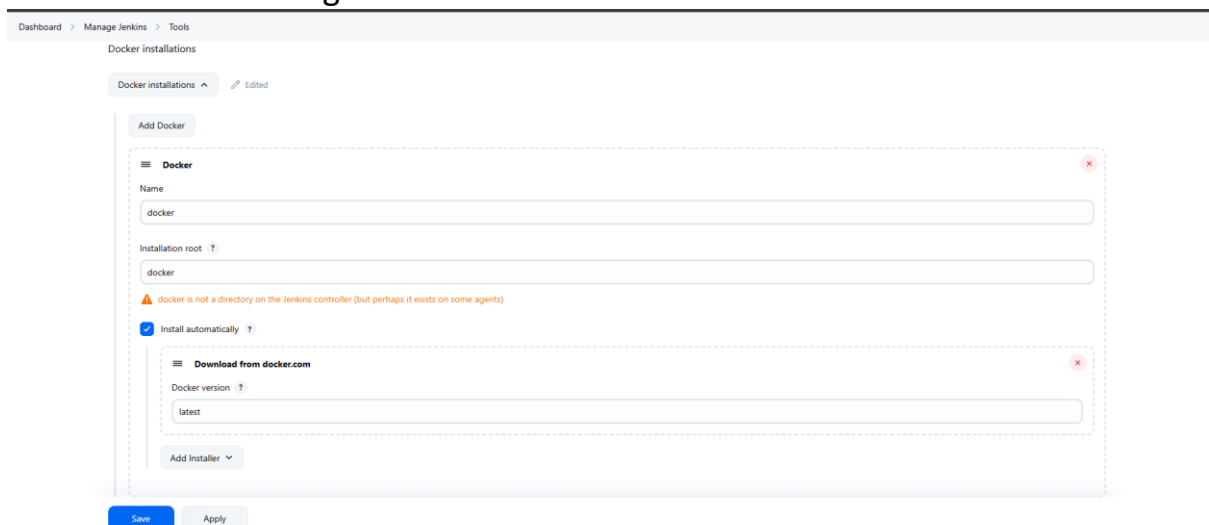
## ⇒ Docker Credentials in Jenkins:

→ 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



The screenshot shows the 'Docker installations' configuration page in Jenkins. It features a 'Docker' section with the following fields:

- Name:** docker
- Installation root:** docker
- A warning message: "docker is not a directory on the Jenkins controller (but perhaps it exists on some agents)"
- Install automatically:** ☒
- Download from docker.com:** A sub-section with a 'Docker version' field set to 'latest'.

At the bottom, there are 'Save' and 'Apply' buttons.

⇒ 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:

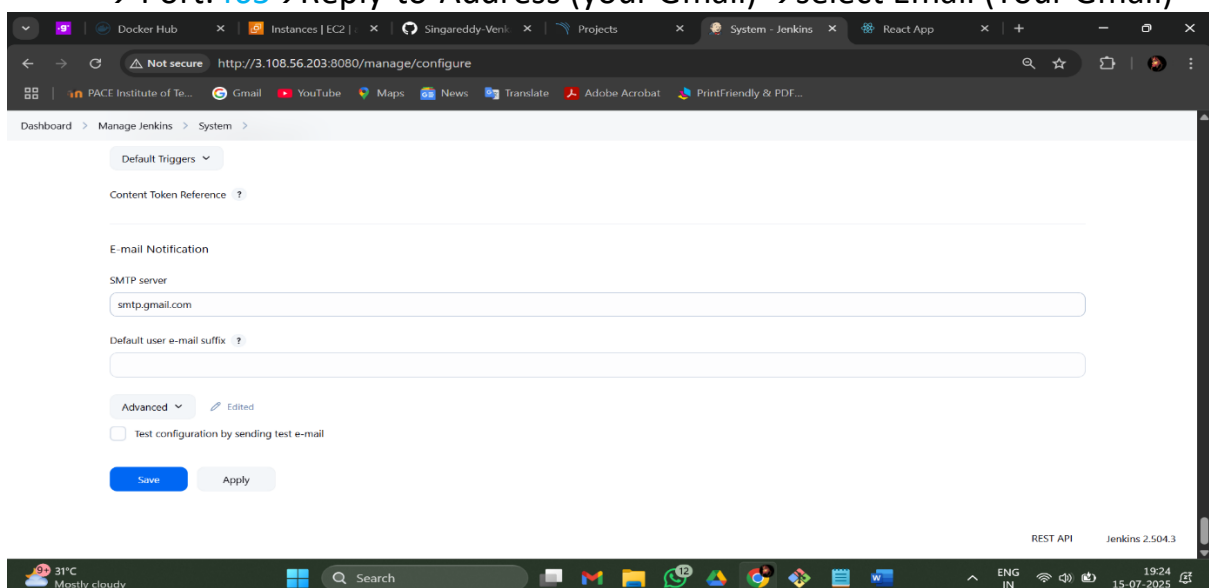
⇒ In Jenkins → Manage Jenkins → NodeJs Installations → node17 → Version (NodeJS 17.0.0)



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

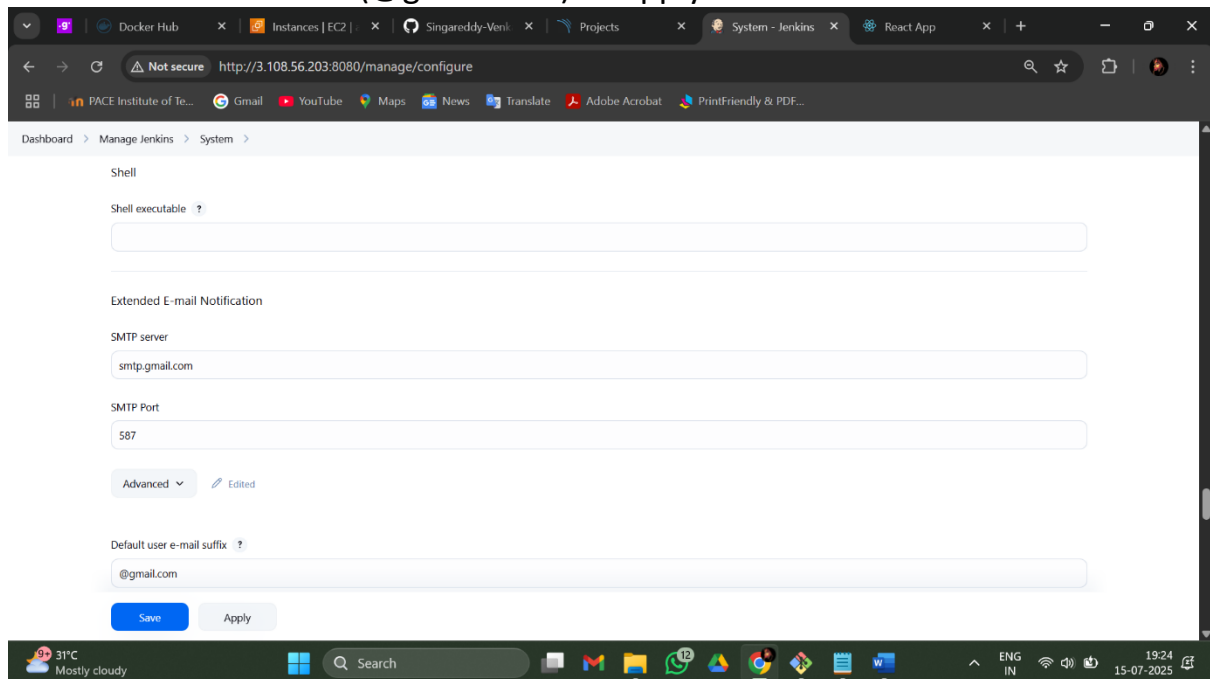
⇒ After go to Jenkins Credentials → Username & Password → Your Gmail & Token which we created → ID (smtp-token) → Create

⇒ 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](mailto:smtp.gmail.com) → Port: **587** → Select Credentials → Select Use **TLS** → Default  
User email Suffix([@gmail.com](mailto:@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'

```

```

        emailx (
            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>

```

## Build The Pipeline Successfully:





## SonarQube Dashboard with Starbucks Project:

The screenshot displays the SonarQube dashboard for the 'starbucks' project. The project status is 'Passed'. The dashboard includes a sidebar with filters for Quality Gate, Reliability, and Security. The main area shows project details and a table of metrics.

Metric	Value	Rating
Bugs	26	D
Vulnerabilities	1	B
Hotspots Reviewed	0.0%	E
Code Smells	8	A
Coverage	0.0%	F
Duplications	13.7%	C
Lines	1.6k	S

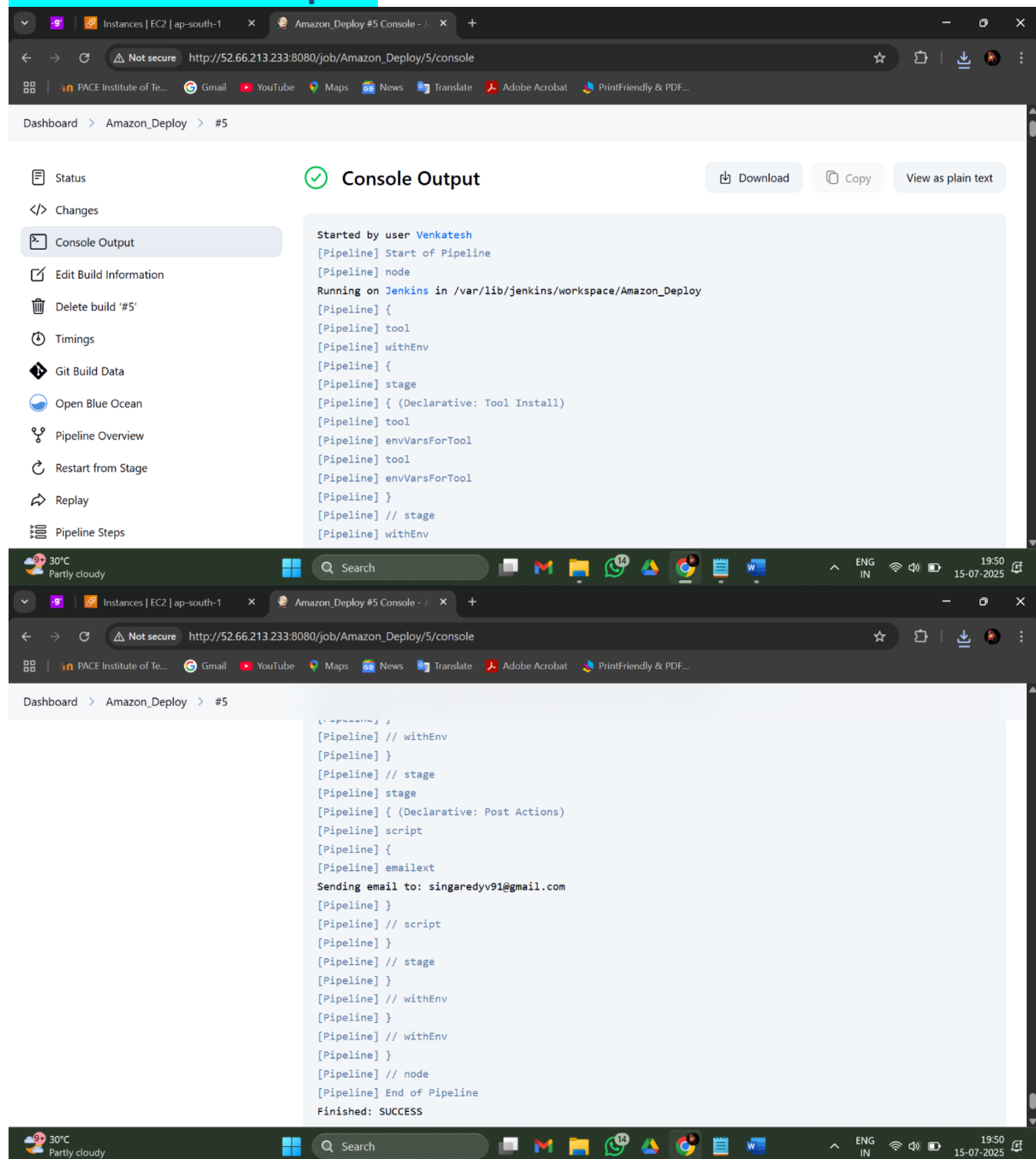
1 of 1 shown

## Docker Hub Image Build:

The screenshot shows the Docker Hub repository page for the user 'venkatesh09'. The page lists several repositories, including 'starbucks', 'recommendationservice', 'frontend', 'loadgenerator', 'productcatalogservice', 'emailservice', and 'checkoutservice'. The 'starbucks' repository is highlighted.

Name	Last Pushed	Contains	Visibility	Scout
venkatesh09/starbucks	2 minutes ago	IMAGE	Public	Inactive
venkatesh09/recommendationservice	over 1 year ago	IMAGE	Public	Inactive
venkatesh09/frontend	over 1 year ago	IMAGE	Public	Inactive
venkatesh09/loadgenerator	over 1 year ago	IMAGE	Public	Inactive
venkatesh09/productcatalogservice	over 1 year ago	IMAGE	Public	Inactive
venkatesh09/emailservice	over 1 year ago	IMAGE	Public	Inactive
venkatesh09/checkoutservice	over 1 year ago	IMAGE	Public	Inactive

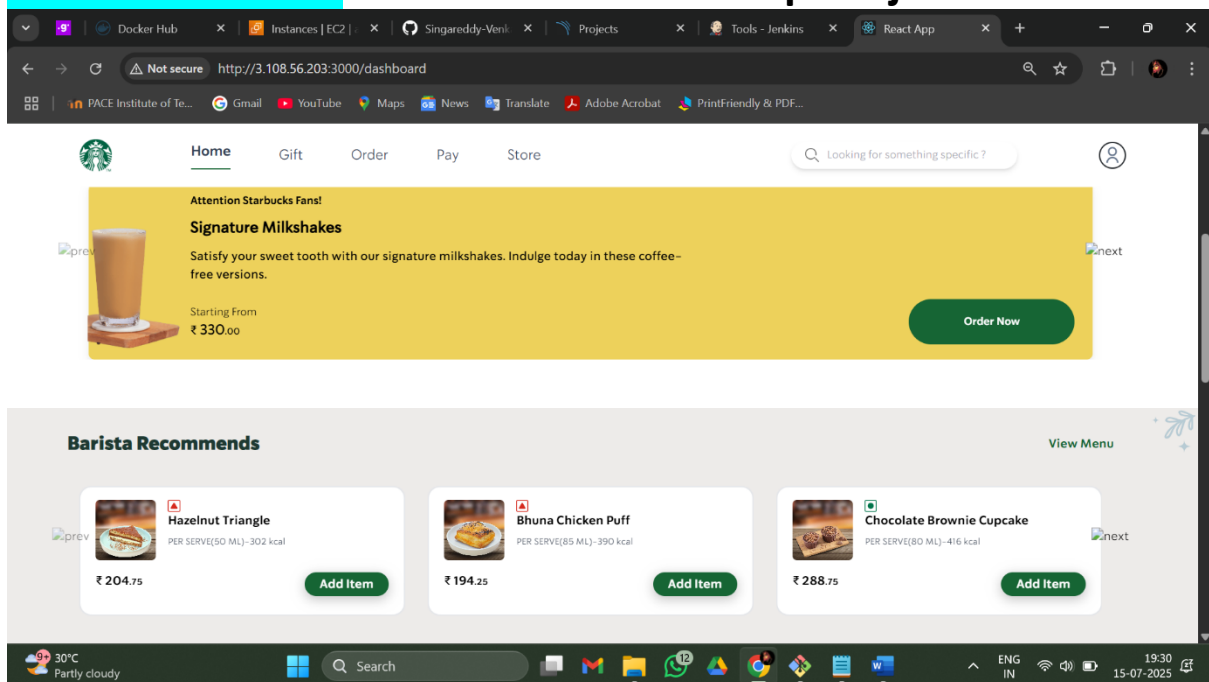
## Console Final Output:



```
Started by user Venkatesh
[Pipeline] Start of Pipeline
[Pipeline] node
Running on Jenkins in /var/lib/jenkins/workspace/Amazon_Deploy
[Pipeline] {
[Pipeline] tool
[Pipeline] withEnv
[Pipeline] {
[Pipeline] stage
[Pipeline] { (Declarative: Tool Install)
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] tool
[Pipeline] envVarsForTool
[Pipeline] }
[Pipeline] // stage
[Pipeline] withEnv
[Pipeline] // withEnv
[Pipeline] }
[Pipeline] // stage
[Pipeline] }
[Pipeline] // node
[Pipeline] End of Pipeline
Finished: SUCCESS
```

→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:

