Name: Shivpratik Hande Class: D15C Rollno.14

Case Study

14. Continuous Integration with Simple Code Analysis

- **Concepts Used**: Jenkins, AWS Cloud9, and SonarQube.
- **Problem Statement**: "Set up a Jenkins pipeline using AWS Cloud9 to perform a simple code analysis on a JavaScript file using SonarQube."
- **Tasks**:
  - Create a Jenkins job using AWS Cloud9.
  - Configure the job to integrate with SonarQube for basic code analysis.
  - Run the Jenkins job with a JavaScript file and review the analysis report.

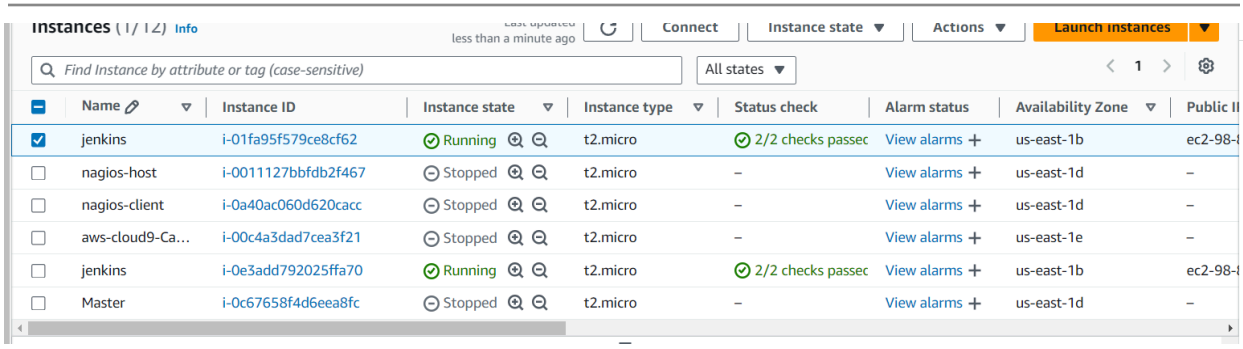
Step-by-Step Explanation

Step 1: Initial Setup and Configuration

- **Launch AWS EC2 Instance** for both **jenkins** and **sonarqube** :
  1. Create an AWS account if you haven't.
  2. Launch a **t2.medium** EC2 instance with **Ubuntu 20.04.**
  3. SSH into the instance using a terminal with the command

Allow the following inbound rules:

- **HTTP (port 80)**: For accessing Jenkins.
- **SSH (port 22)**: For secure shell access.
- **Custom TCP (port 8080)**: For accessing Jenkins.



Step 2: Install Jenkins on EC2 (Ubuntu)

- ssh -i path/to/your-key.pem ubuntu@**<your-EC2-IP>**
- sudo apt update

Name: Shivpratik Hande Class: D15C Rollno.14

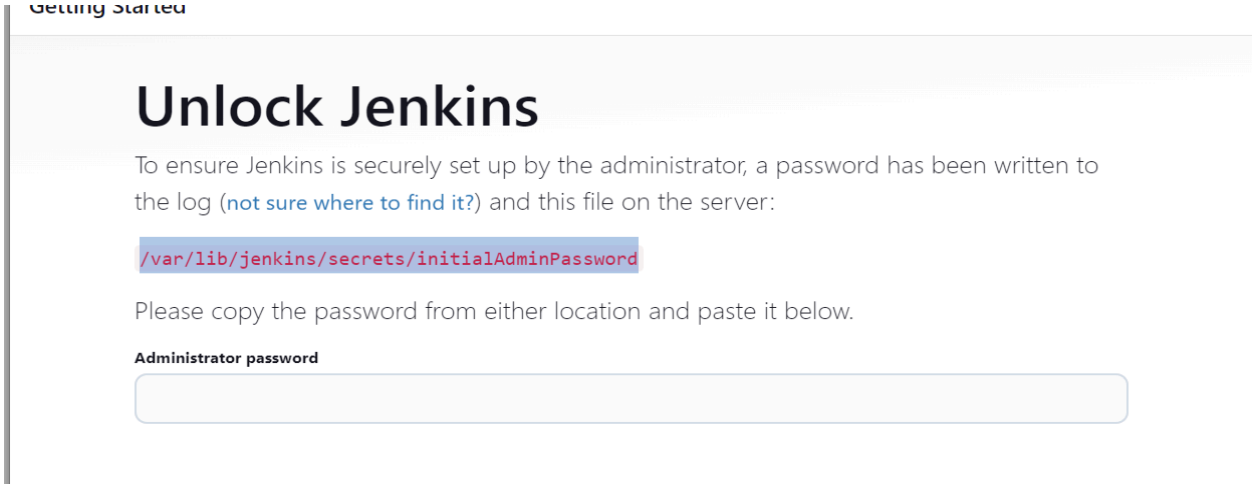
- sudo apt install fontconfig openjdk-17-jre
- java -version

Add the Jenkins repository

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

- sudo systemctl start jenkins
- sudo systemctl enable jenkins
- sudo systemctl status jenkins

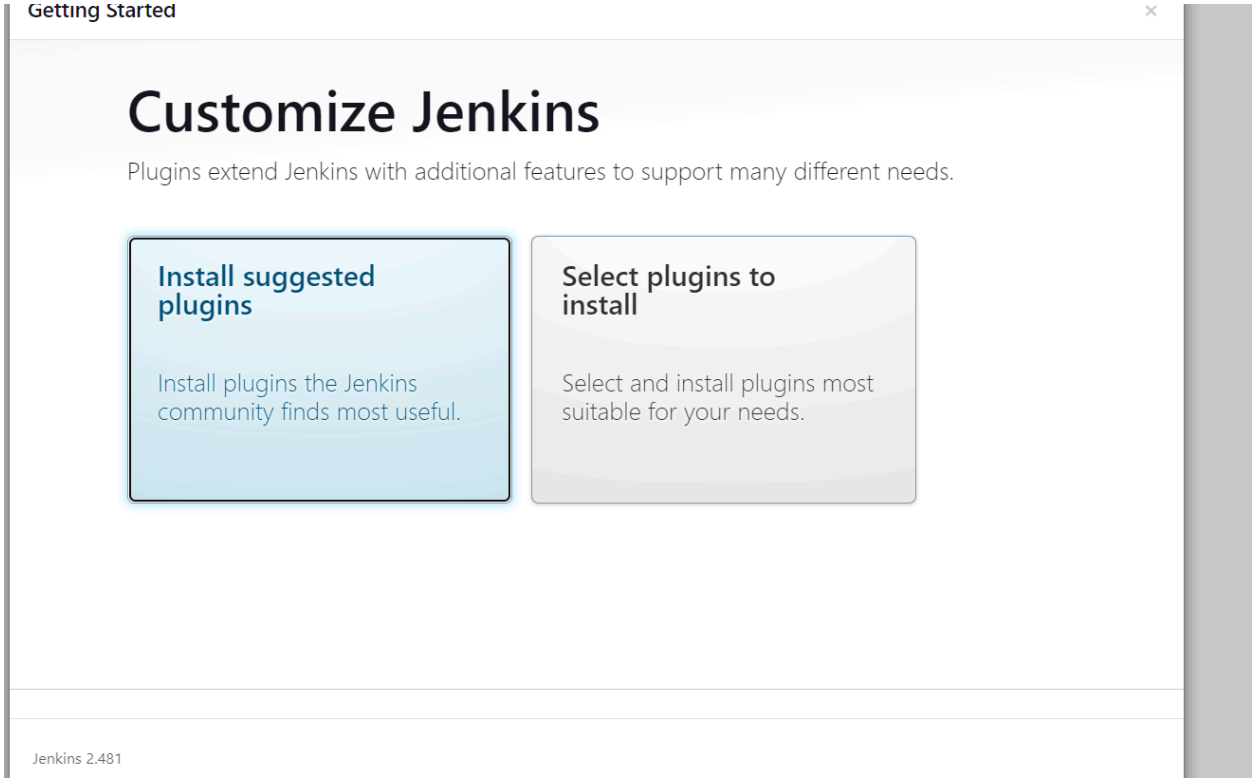
Open a browser and navigate to `http://<your-EC2-IP>:8080`.

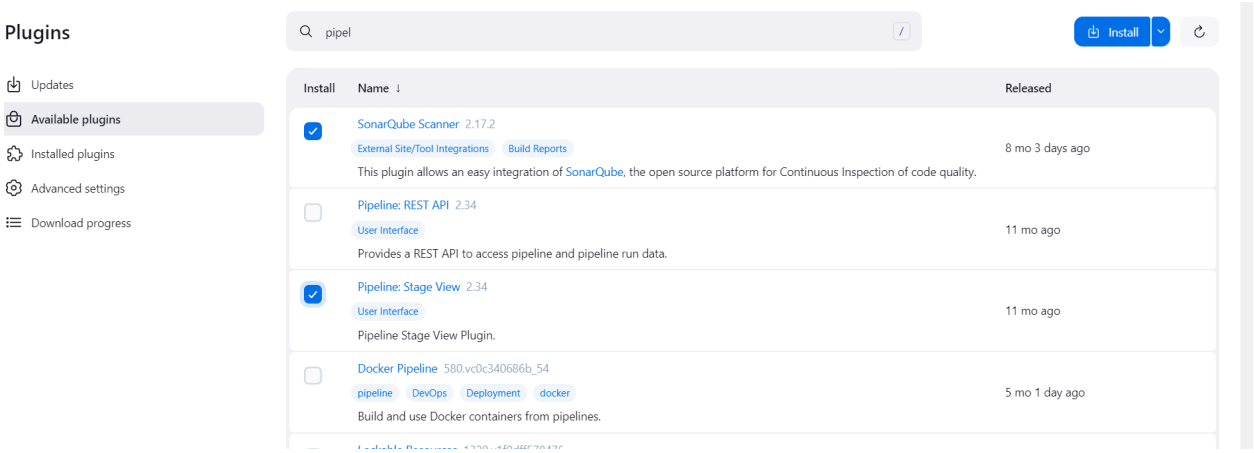



```
sudo cat /var/lib/jenkins/secrets/initialAdminPassword
```

To get Administrator Password

Name: Shivpratik Hande Class: D15C Rollno.14



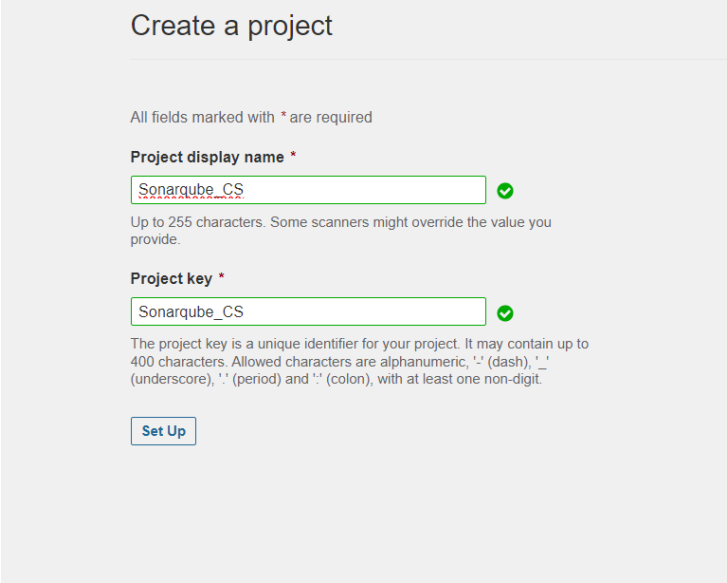


Step 3: To setup sonarqube in ec2

Follow :

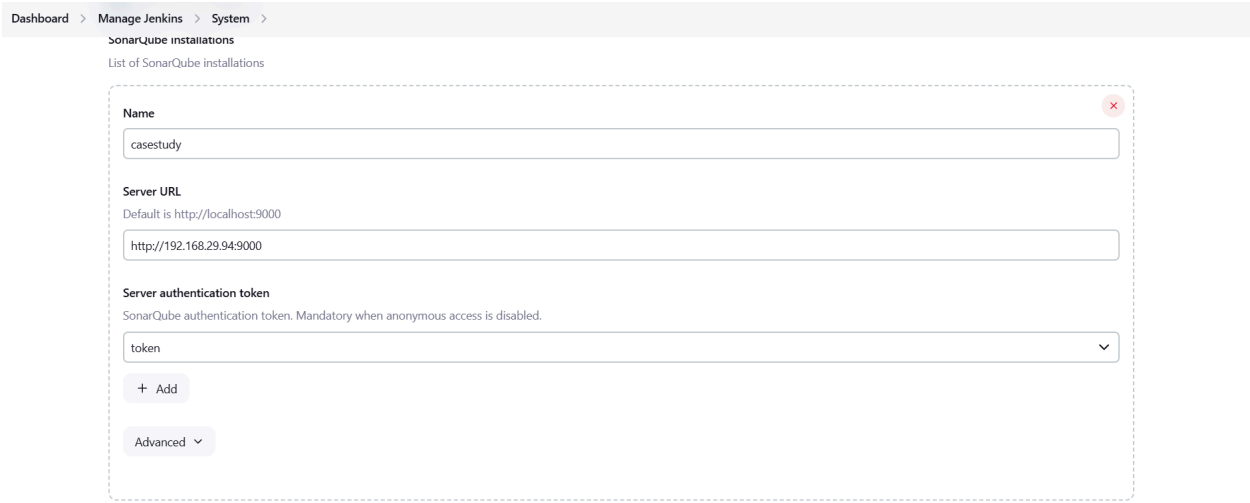
[How To Install SonarQube on Ubuntu | by Joyal Saji | Medium](#) till **Setup Systemd Service**

Open a browser and navigate to `http://<your-EC2-IP>:9000`.

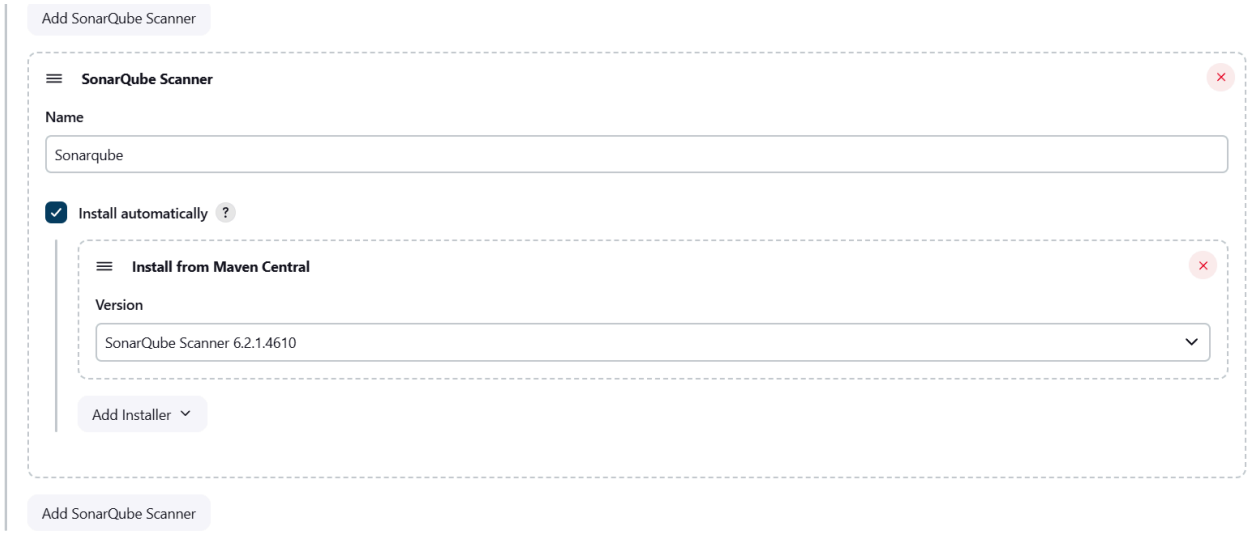


Step 4: Integrate Jenkins with SonarQube

1. **Install SonarQube Scanner Plugin in Jenkins**:
   - Go to **Manage Jenkins** → **Manage Plugins**.
   - Search for **SonarQube Scanner** and install it.
2. **Configure SonarQube Server in Jenkins**:
   - Go to **Manage Jenkins** → **Configure System**.
   - Find the **SonarQube servers** section and click **Add SonarQube**.
   - Enter:
     - **Name**: SonarQube
     - **Server URL**: `http://<your-local-IP>:9000` (use your local machine's IP, not `localhost`).
     - **Server authentication token**: Generate a token in SonarQube by going to **My Account** → **Security** → **Generate Tokens**.
3. **Add Credentials in Jenkins**:
   - Go to **Manage Jenkins** → **Manage Credentials** → **Add a new credential**.
   - Add your SonarQube token as a **Secret Text** credential.



4. **Set sonarqube Scanner**
   Manage Jenkins → Tools

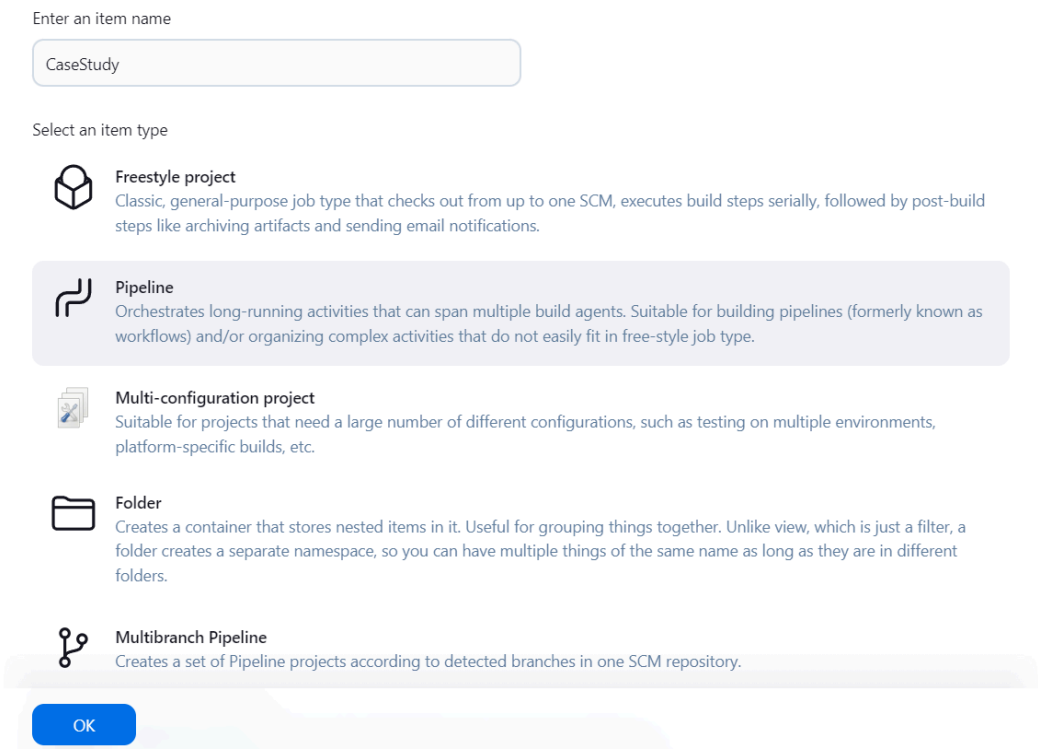


Step 5: Create Pipeline project

Pipeline code:

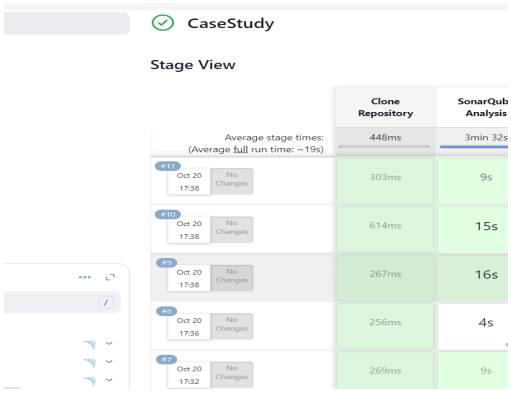
```
pipeline {
   agent any
   stages {
      stage('Clone Repository') {
         steps {
            git 'https://github.com/Ashloneer/adv-devops-casestudy'
         }
```

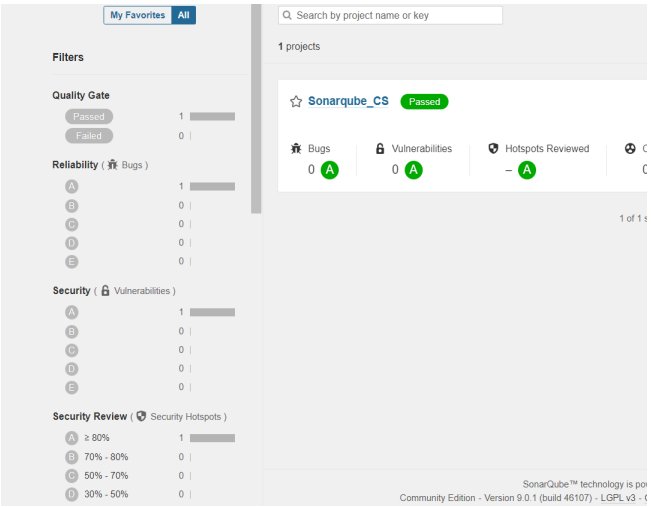
```
        }
        stage('SonarQube Analysis') {
            environment {
                SONAR_TOKEN = '48e56dcd98b809d4b5b4e90b25167a191efd9eca' // Your actual
token
            }
            steps {
                withSonarQubeEnv('Sonarqube') {
                    sh """
                    ${tool 'Sonarqube'}/bin/sonar-scanner \\
                      -Dsonar.projectKey=Sonarqube_CS \\
                      -Dsonar.sources=. \\
                      -Dsonar.host.url=http://54.166.51.117:9000 \\
                      -Dsonar.login=${SONAR_TOKEN}
                    """
                }
            }
        }
    }
}
```

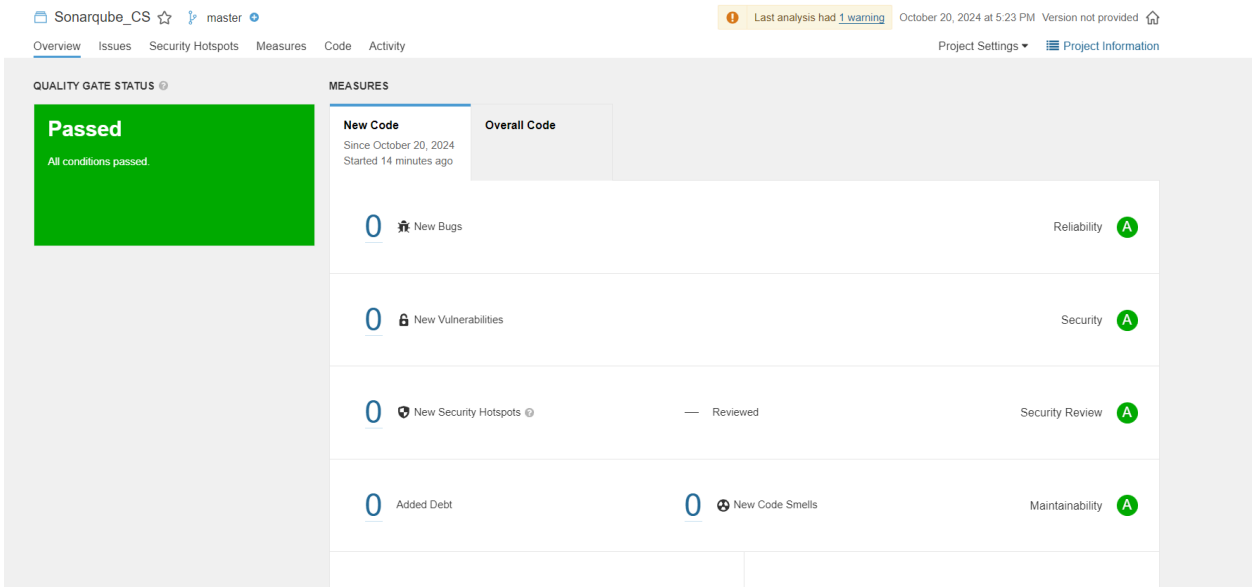


After adding pipeline : Build project by clicking **Build Now**

Name: Shivpratik Hande Class: D15C Rollno.14







done…….