Contents

Jenkins Pipeline as Code – Jenkinsfile for Static Code Analysis

1. Following is the Pipeline that uses docker image 'sonarsource/sonar-scanner-cli:latest' as an agent and performs static code analysis.

```
2. pipeline {
3.   agent any
4.
5.     stages {
6.    stage('SCA') {
7.      agent {
8.        docker {
9.          image 'sonarsource/sonar-scanner-cli:latest'
10.       }
11.     }
12.     steps {
13.       echo "Steps to execute SCA"
14.       withSonarQubeEnv(installationName: 'SonarQube', credentialsId:
      'SonarToken') {
15.         sh 'sonar-scanner -Dsonar.projectVersion=1.0 -
      Dsonar.projectKey=react-bmi-app -Dsonar.sources=src'
16.       }
17.       waitForQualityGate(abortPipeline: true, credentialsId: 'SonarT
      oken')
18.     }
19.   }
20. }
21.}
```

2. You can also use Dockerfile as an Agent.

Dockerfile.sonar

FROM sonarsource/sonar-scanner-cli:latest

Jenkinsfile

```
1. pipeline {
2.   agent any
3.
4.   stages {
5.   stage('SCA') {
6.   agent {
7.   dockerfile {
8.   filename 'Dockerfile.sonar'
9.   }
10.   }
11.   steps {
12.   echo "Steps to execute SCA"
13.   withSonarQubeEnv(installationName: 'SonarQube', credentialsId:
      'SonarToken') {
```

```
14.        sh 'sonar-scanner -Dsonar.projectVersion=1.0 -
   Dsonar.projectKey=react-bmi-app -Dsonar.sources=src'
15.     }
16.     waitForQualityGate(abortPipeline: true, credentialsId: 'SonarT
   oken')
17.    }
18.  }
19. }
20. }
```

3. Execute Pipeline in Jenkins. Verify SonarQube Quality Gate status.
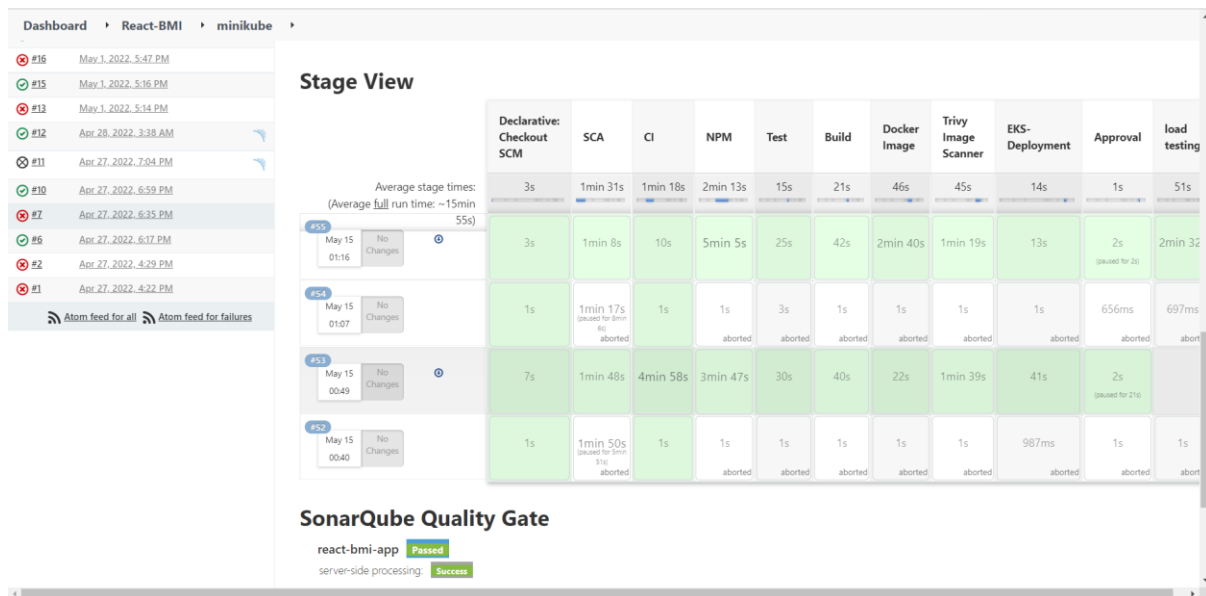
4. Verify the Console logs for following:

```
Checking status of SonarQube task 'AX4hm1OcKzRTh-f6frYb' on server
'SonarQube'
SonarQube task 'AX4hm1OcKzRTh-fxxxxx' status is 'IN_PROGRESS'
SonarQube task 'AX4hm1OcKzRTh-fxxxxx' status is 'SUCCESS'
SonarQube task 'AX4hm1OcKzRTh-fxxxxx' completed. Quality gate is 'OK'
```

5. Verify SonarQube Dashboard.

Done!