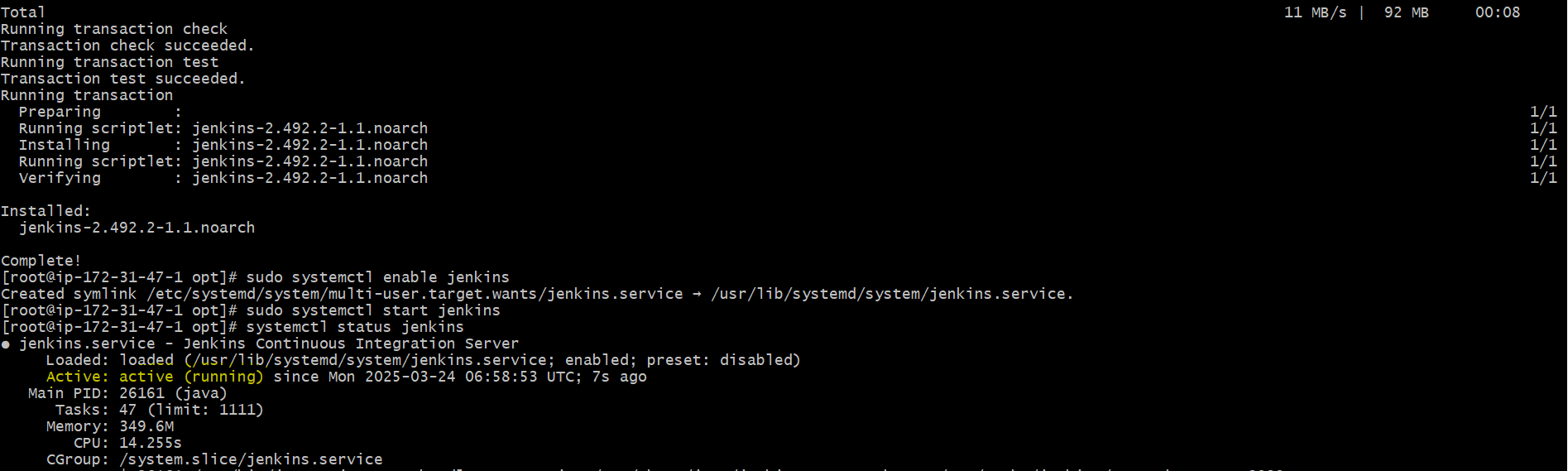
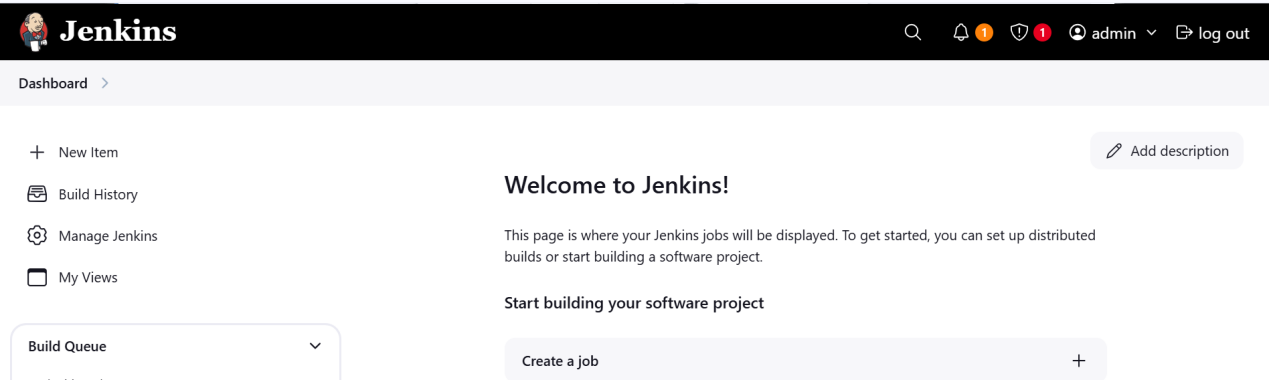
**JENKINS SCRIPTED PIPELINE**

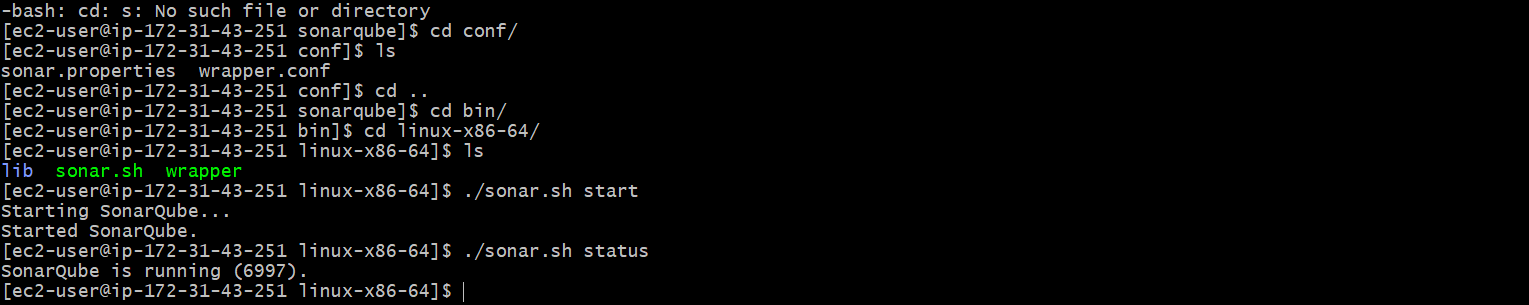
**STEP-1:** Launched an EC2 Instance and installed Jenkins



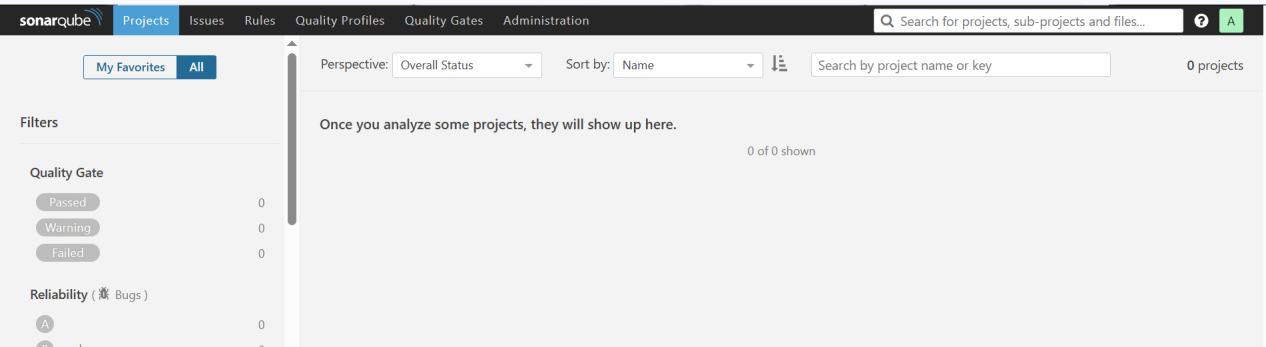
<http://public>IPOfec2:8080/



**STEP-2:** Launched an EC2 Instance and installed MySQL, sonarqube and integrated MySQL with sonarqube by configuring MySQL credentials, url and required information in sonarqube.properties file



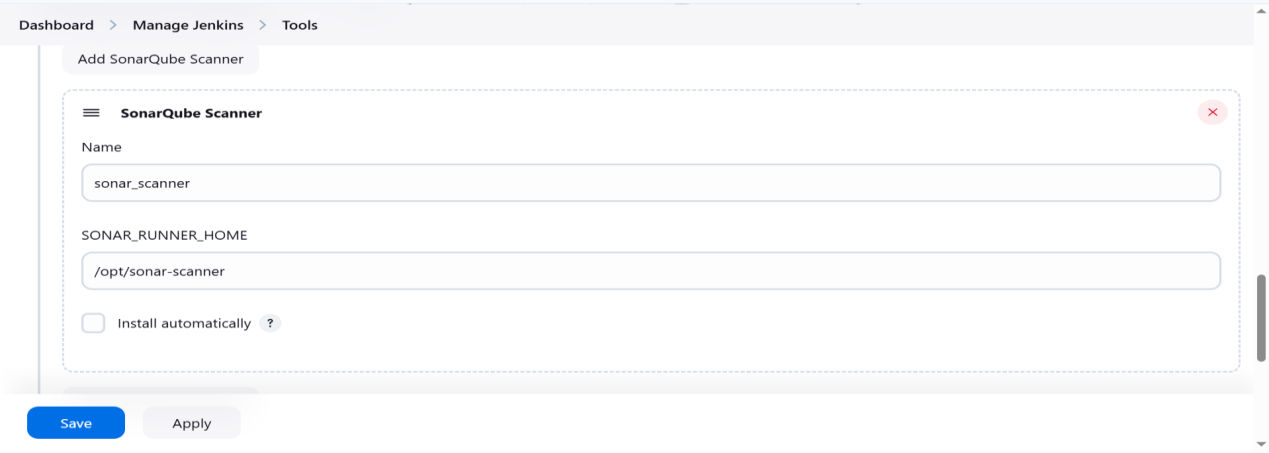
Logged into sonarqube using default Userid:passwrd ~ admin:admin



---Installed sonarqube-scanner in jenkins instance and configured sonarqube url in sonarqubescanner.properties file.

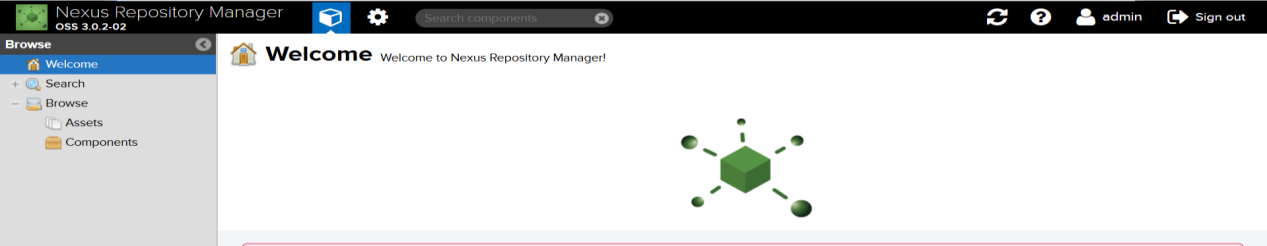
---installed sonarqube-scanner plugin in jenkins web app and configured home path of sonarScanner installed in jenkins server in tools at sonarqube scanner field.

--- Configured Url and credentials( token) of sonarqube in system configurations of jenkins at sonarqube installations.

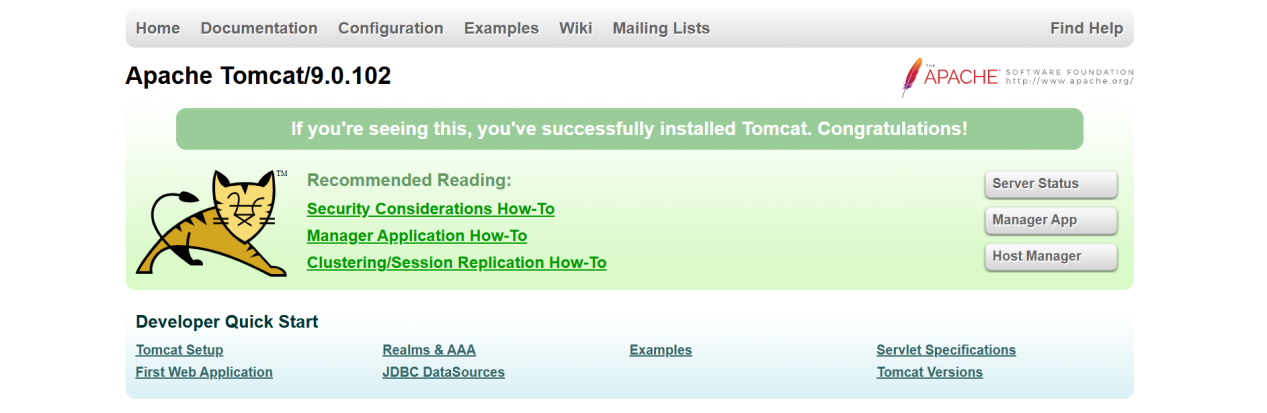




**STEP-3:** Launched an EC2 Instance and installed Nexus (nexus-3.0.2-02)



**STEP-4:** Launched an EC2 Instance and installed Tomcat

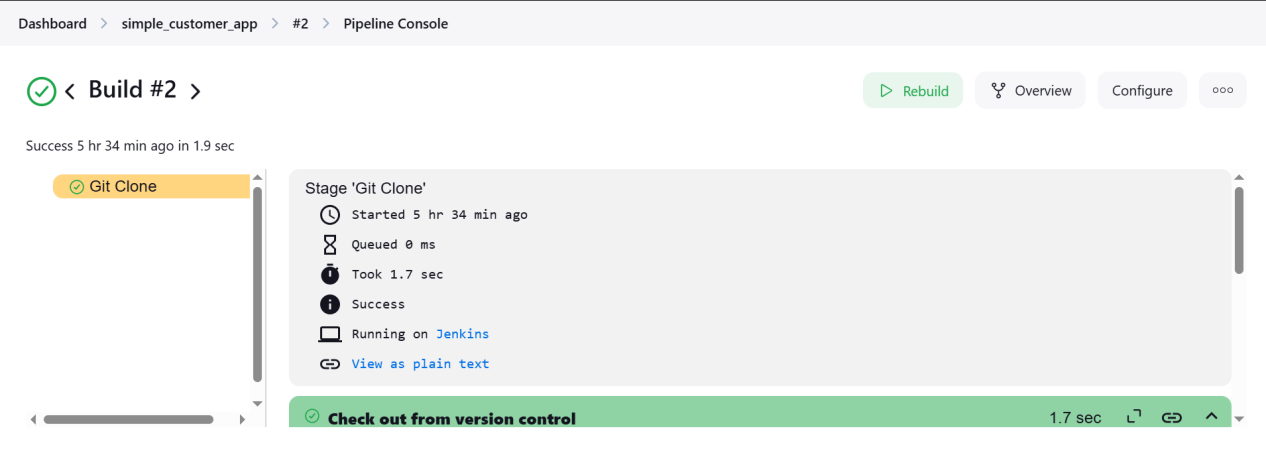


**STEP-5:** Cloned feature-1.1 branch from git repository.

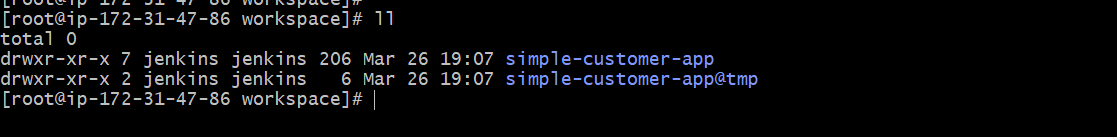
---For this we need git installed in our jenkins server



---Build

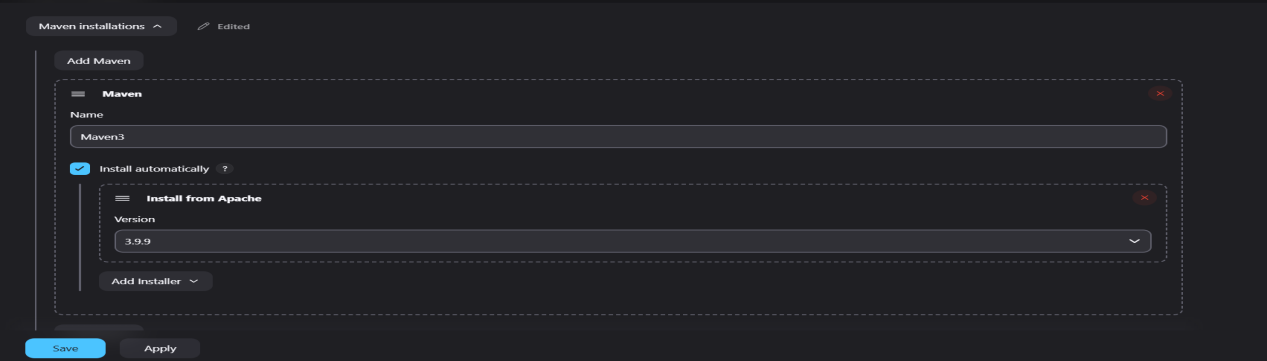


---Cloned feature-1.1 branch from git repository.

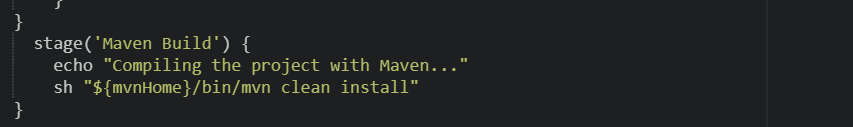


**STEP-6:** Use Maven to compile and package the Java application

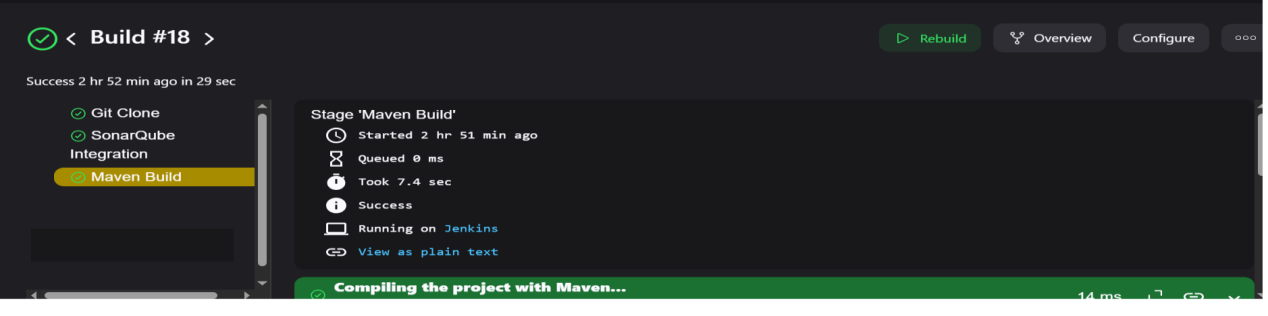
---Install maven-integration plugin and configure in global tool configuration

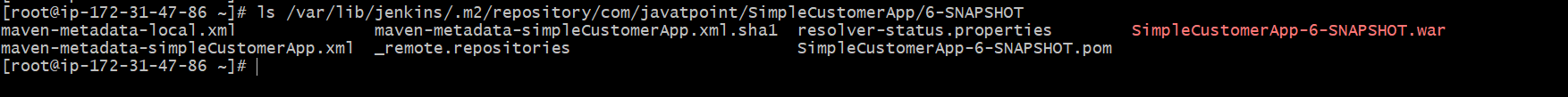


---Snippet for maven execution



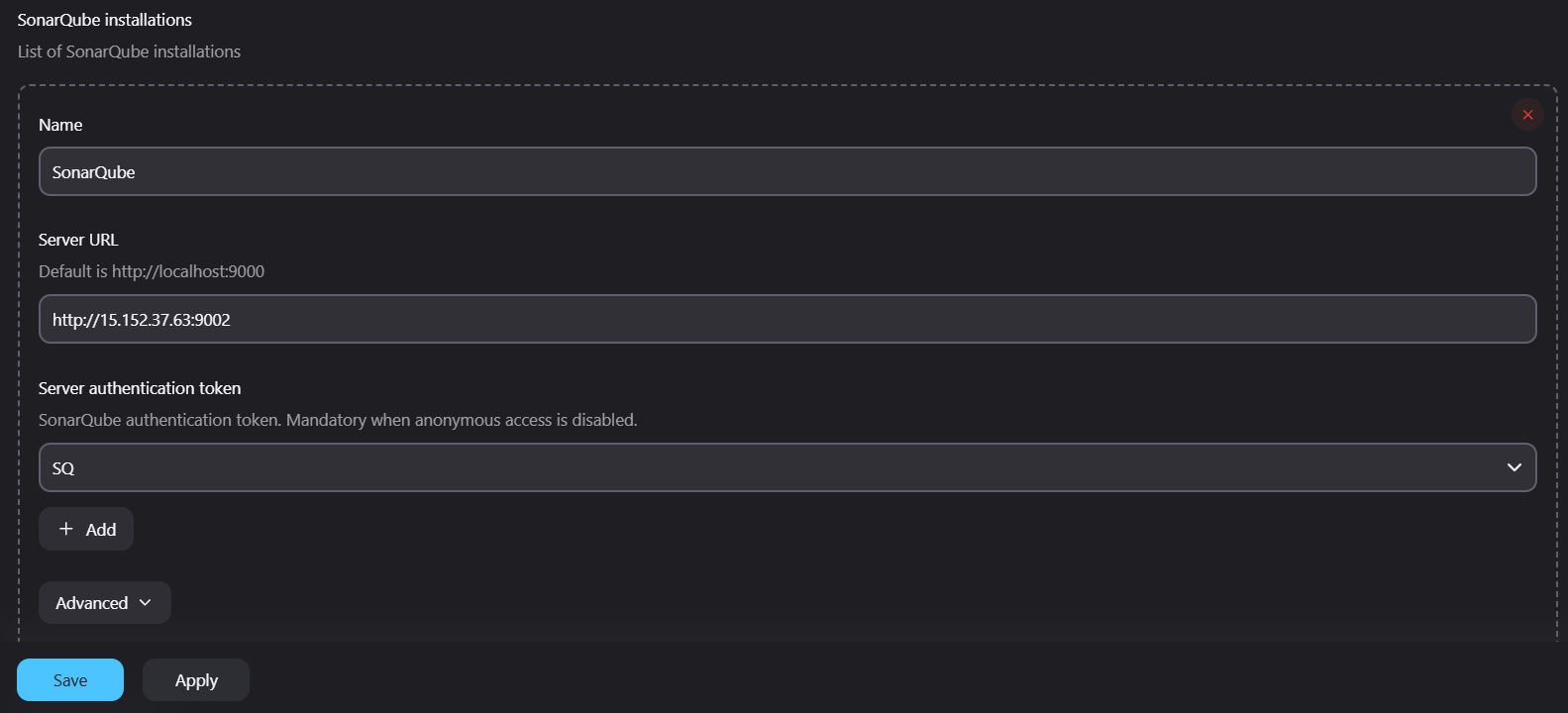
---Build



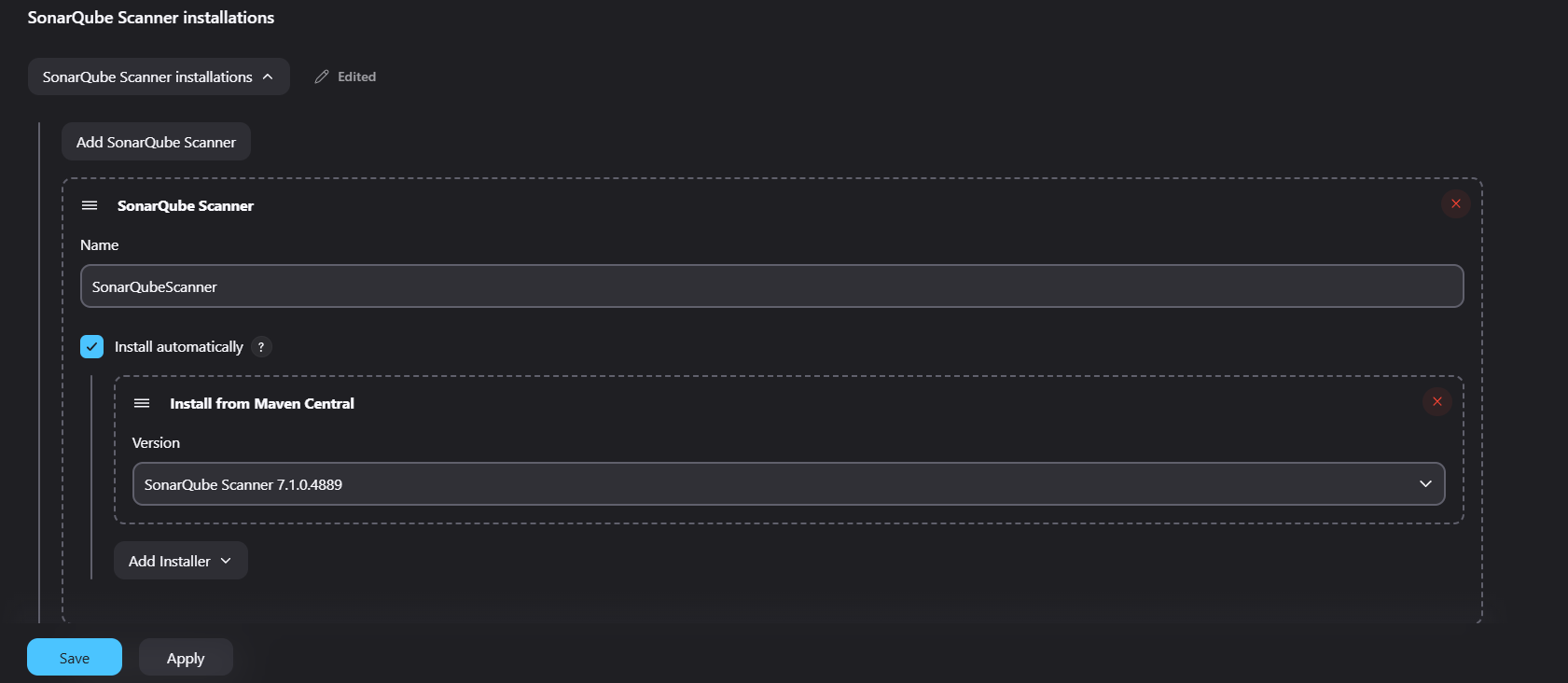


**STEP-7:** Integrating SonarQube with jenkins to Perform Code Analysis

--- configure sonarqube url and credentials in system configurations



---install sonarqube-scanner plugin and comfigure sonarqube installations for the server in global tool configuration

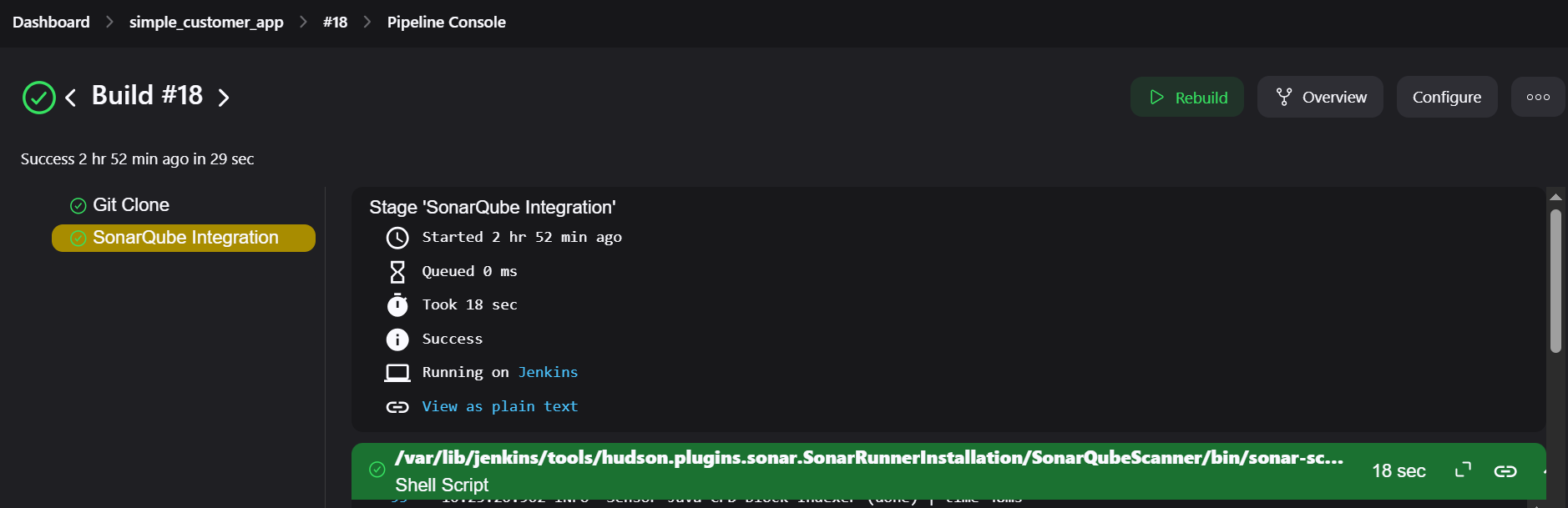


--- script for static code analysis

---This stage scans code for bugs, vulnerabilities, code smells, and duplications.



--- build job





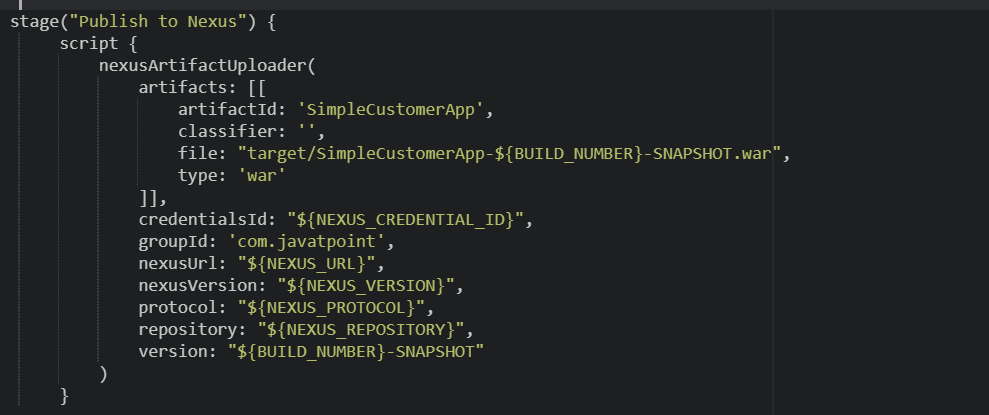
**STEP-8:** Upload the built .war file to Nexus for storage.

--- install nexus-artifact uploader plugin

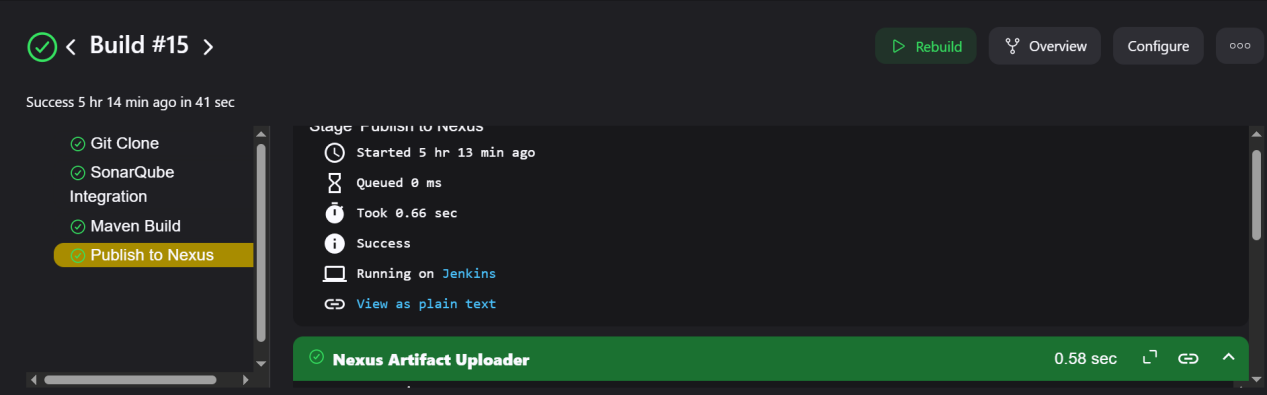
--- create a secret(credential) with username and password

--- create a Repository in nexus

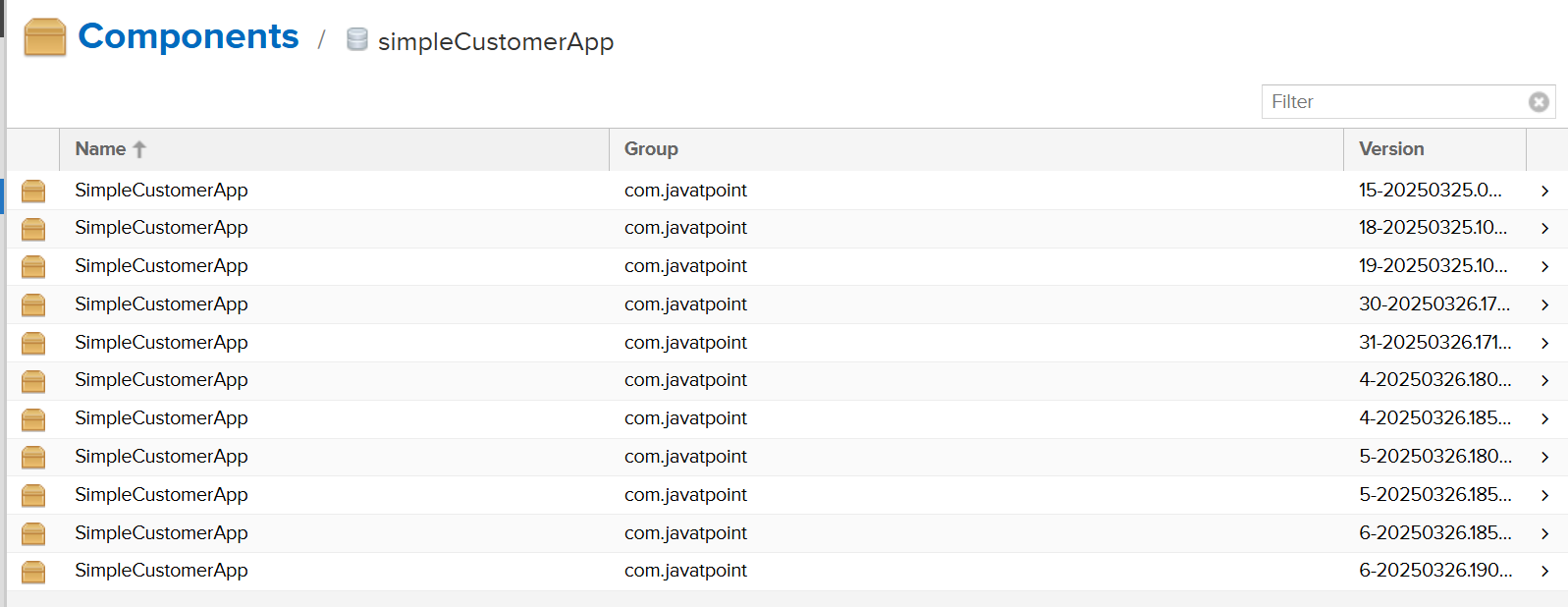
--- Script for Nexus configuration



---Build job



---nexus repository



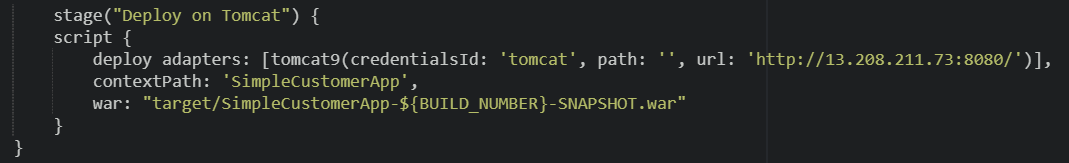
**STEP-9:** Deploy the .war file to the Tomcat application server.

--- Install deploy to container plugin

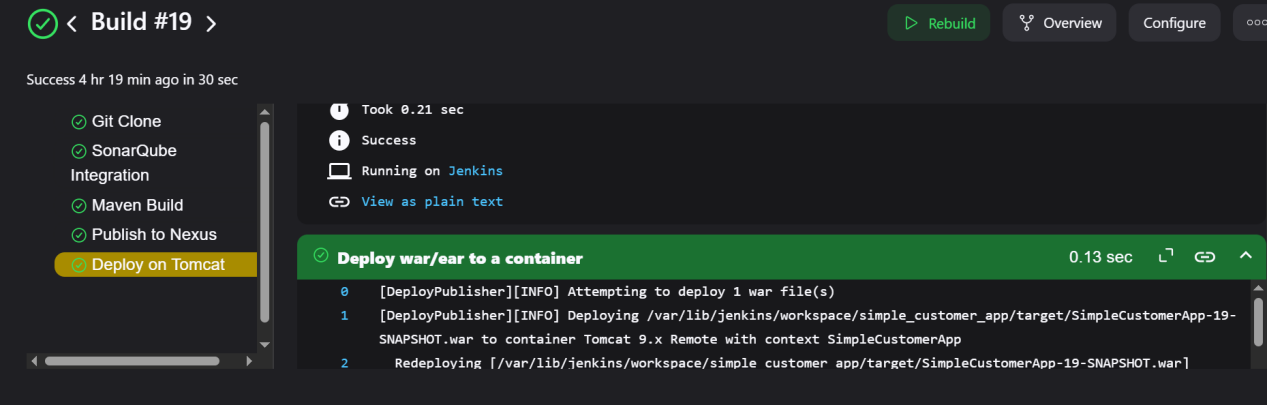
--- In Tomcat create a user with required permissons and roles

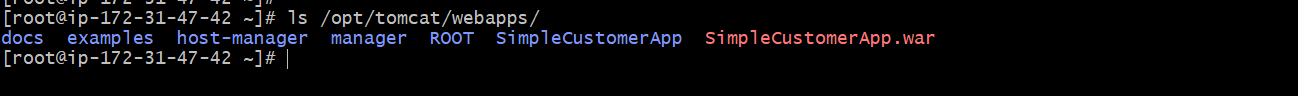
--- Create a secret(Credential) with tomcat user id and password

---Script for tomcat deployment



---Build job

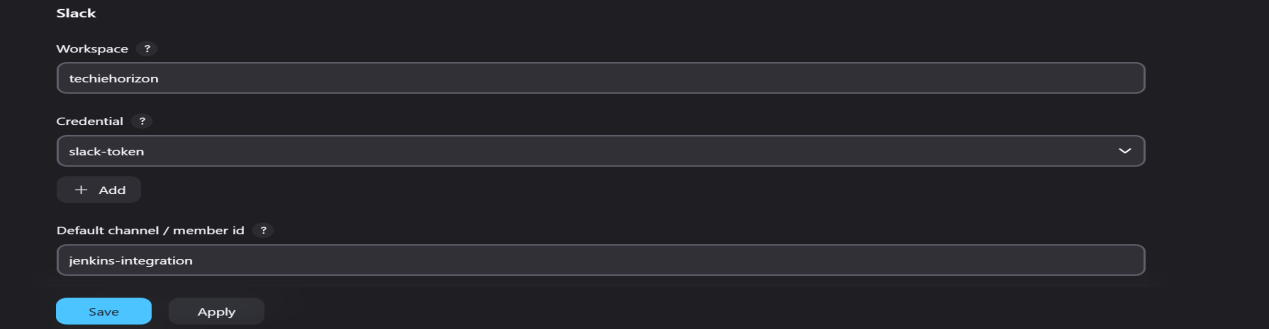




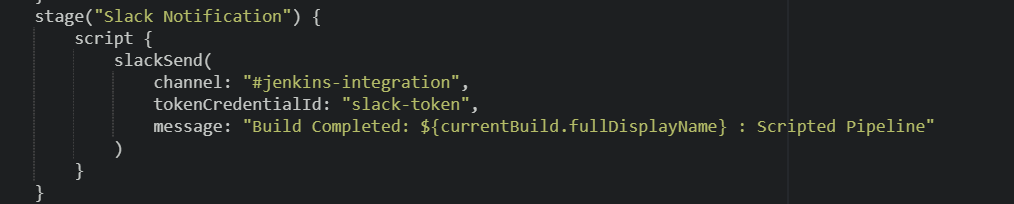
**STEP-10:** Sending pipeline status notifications to Slack.

---Install slack notification plugin

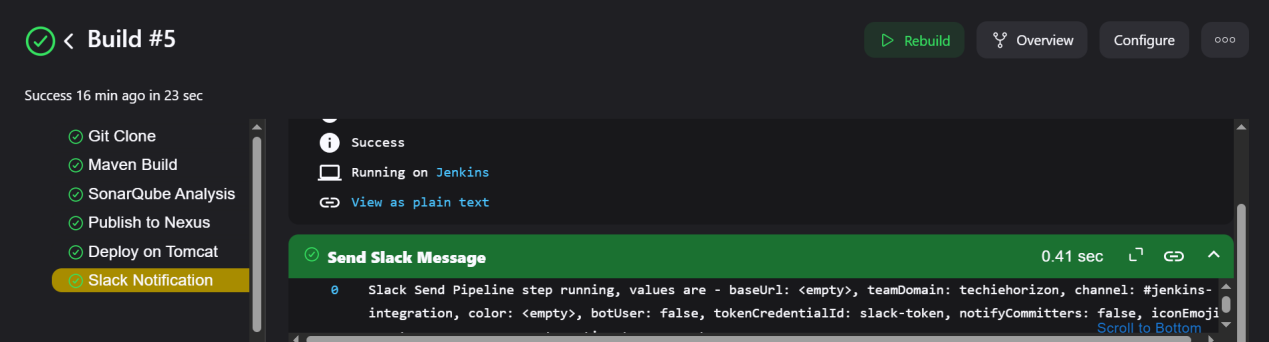
---Set the Slack workspace, channel, and token in system configurations

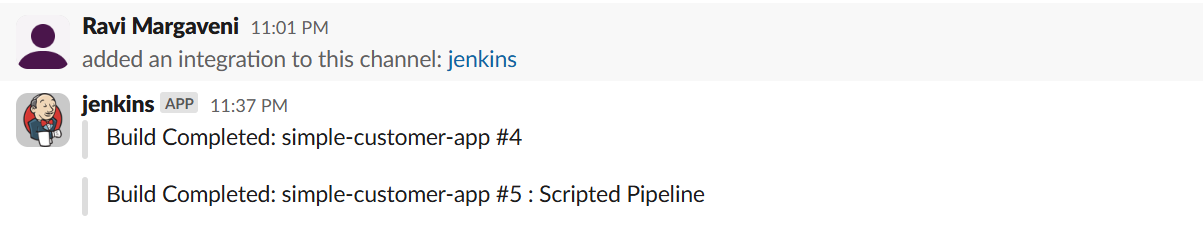


---Script for slack configuration



---Build job





Script

node {

  def NEXUS\_VERSION = "nexus3"

  def NEXUS\_PROTOCOL = "http"

  def NEXUS\_URL = "56.155.36.211:8081/"

  def NEXUS\_REPOSITORY = "simpleCustomerApp"

  def NEXUS\_CREDENTIAL\_ID = "nexus"

  def mvnHome = tool name: 'Maven3'

  def SCANNER\_HOME = tool 'SonarQubeScanner'

  stage('Git Clone') {

      checkout([$class: 'GitSCM',

          branches: [[name: 'feature-1.1']],

          userRemoteConfigs: [[url: 'https://github.com/betawins/sabear\_simplecutomerapp.git']]

      ])

  }

  stage('Maven Build') {

      echo "Compiling the project with Maven..."

      sh "${mvnHome}/bin/mvn clean install"

  }

  stage('SonarQube Analysis') {

        withSonarQubeEnv('SonarQube') {

        sh "${mvnHome}/bin/mvn sonar:sonar"

    }

}

  stage("Publish to Nexus") {

      script {

          nexusArtifactUploader(

              artifacts: [[

                  artifactId: 'SimpleCustomerApp',

                  classifier: '',

                  file: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war",

                  type: 'war'

              ]],

              credentialsId: "${NEXUS\_CREDENTIAL\_ID}",

              groupId: 'com.javatpoint',

              nexusUrl: "${NEXUS\_URL}",

              nexusVersion: "${NEXUS\_VERSION}",

              protocol: "${NEXUS\_PROTOCOL}",

              repository: "${NEXUS\_REPOSITORY}",

              version: "${BUILD\_NUMBER}-SNAPSHOT"

          )

      }

}

      stage("Deploy on Tomcat") {

      script {

          deploy adapters: [tomcat9(credentialsId: 'tomcat', path: '', url: 'http://13.208.211.73:8080/')],

          contextPath: 'SimpleCustomerApp',

          war: "target/SimpleCustomerApp-${BUILD\_NUMBER}-SNAPSHOT.war"

      }

  }

stage("Slack Notification") {

        script {

            slackSend(

                channel: "#jenkins-integration",

                tokenCredentialId: "slack-token",

                message: "Build Completed: ${currentBuild.fullDisplayName} : Scripted Pipeline"

            )

        }

    }

}