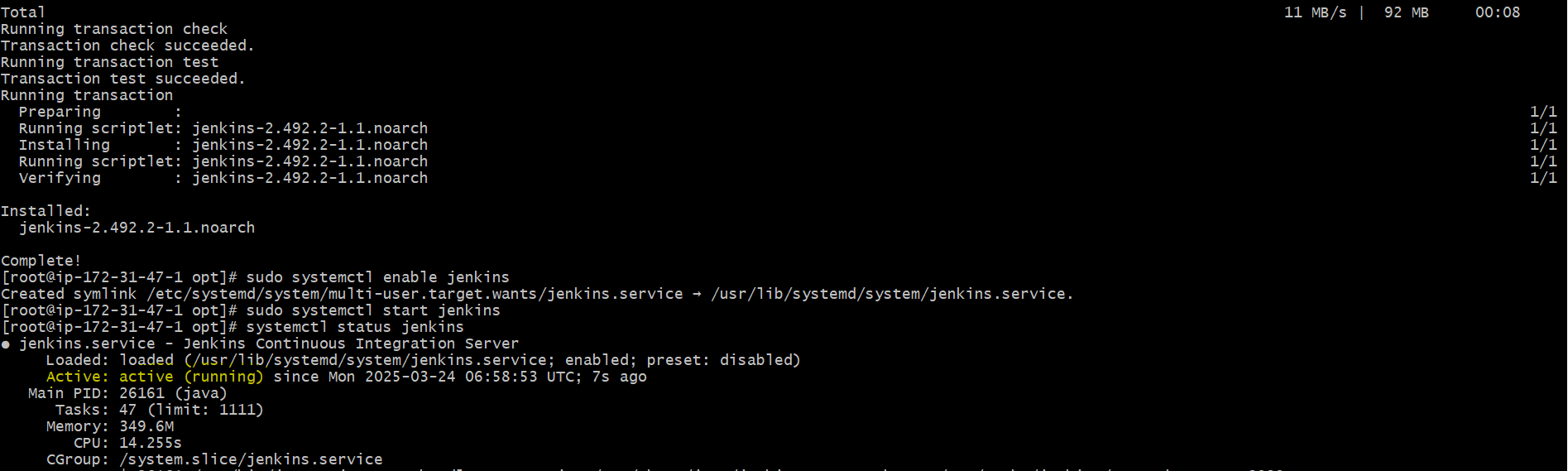
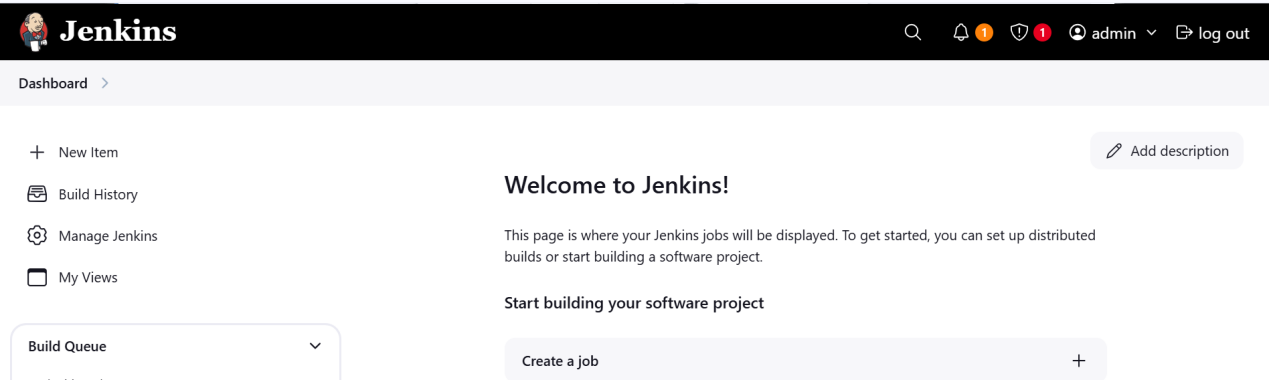
**JENKINS SCRIPTED PIPELINE**

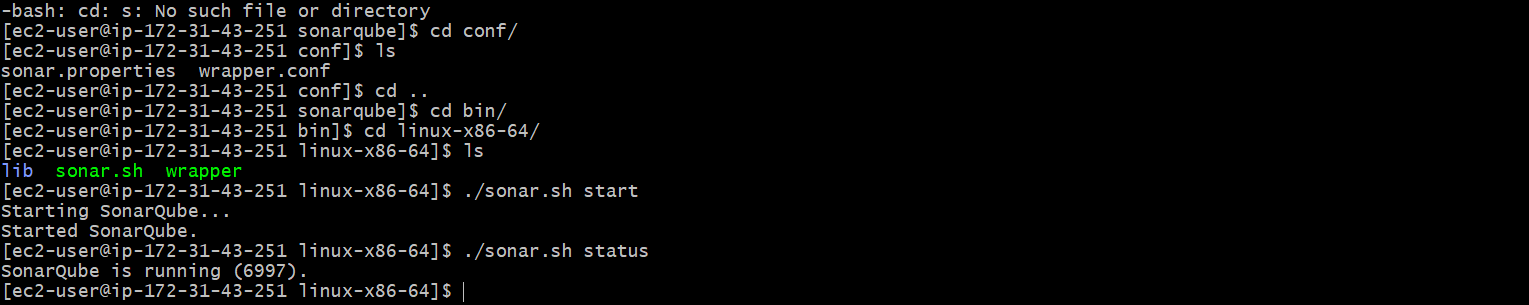
**STEP-1:** Launched an EC2 Instance and installed Jenkins



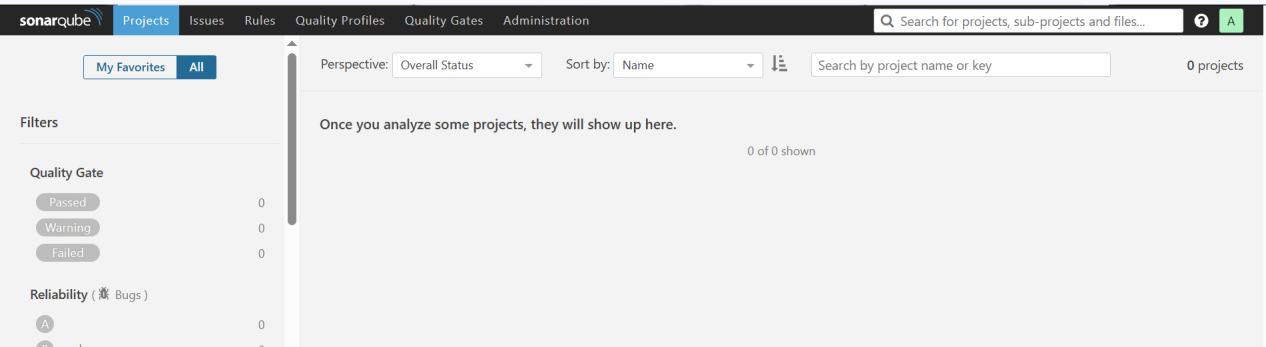
<http://public>IPOfec2:8080/



**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



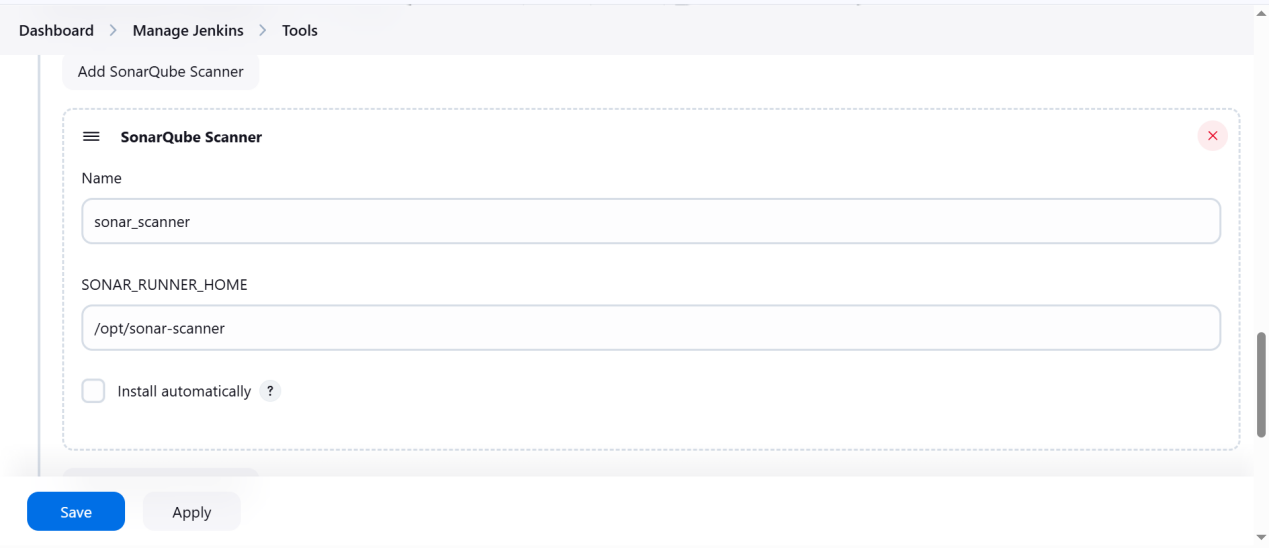
Logged into sonarqube using default Userid:passwrd ~ admin:admin



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

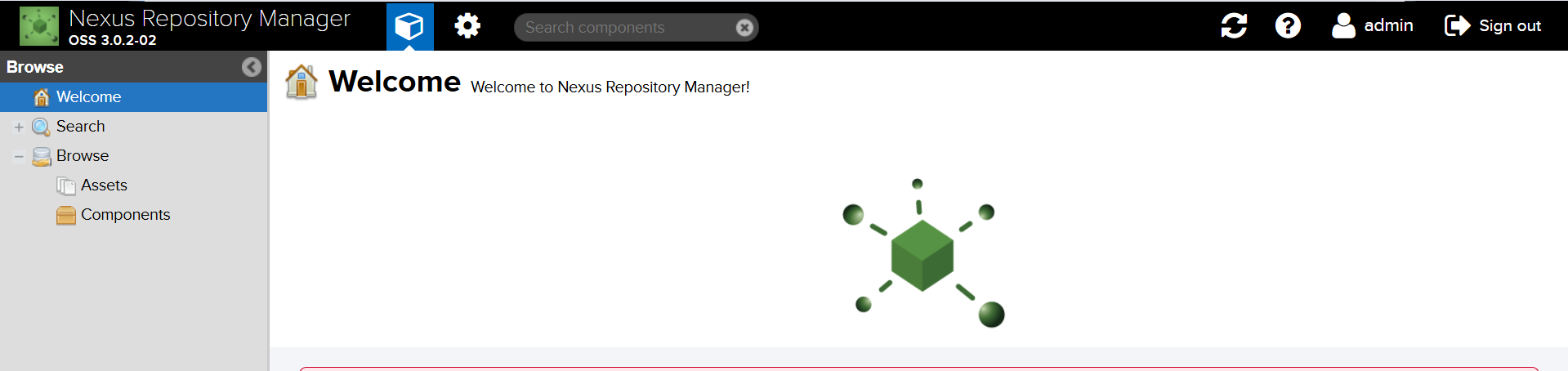
---installed sonarqube-scanner plugin in jenkins web app and configured home path of sonarScanner installed in jenkins server in tools at sonarqube scanner field.

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

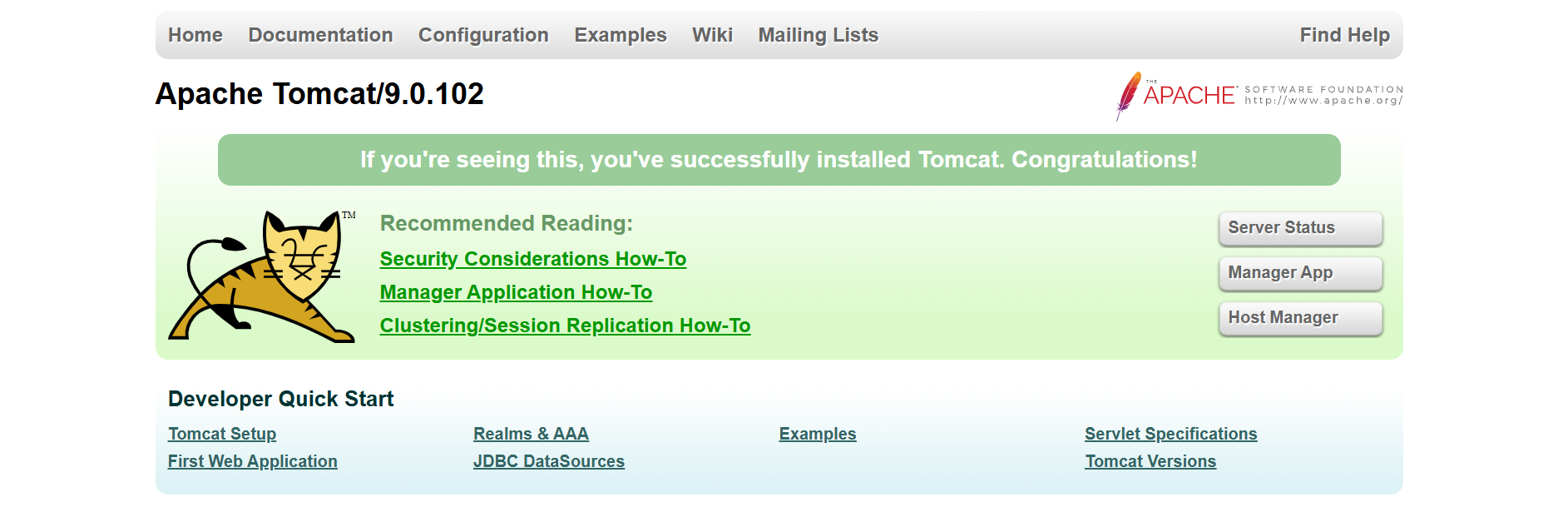




**STEP-3:** Launched an EC2 Instance and installed Nexus (nexus-3.0.2-02)



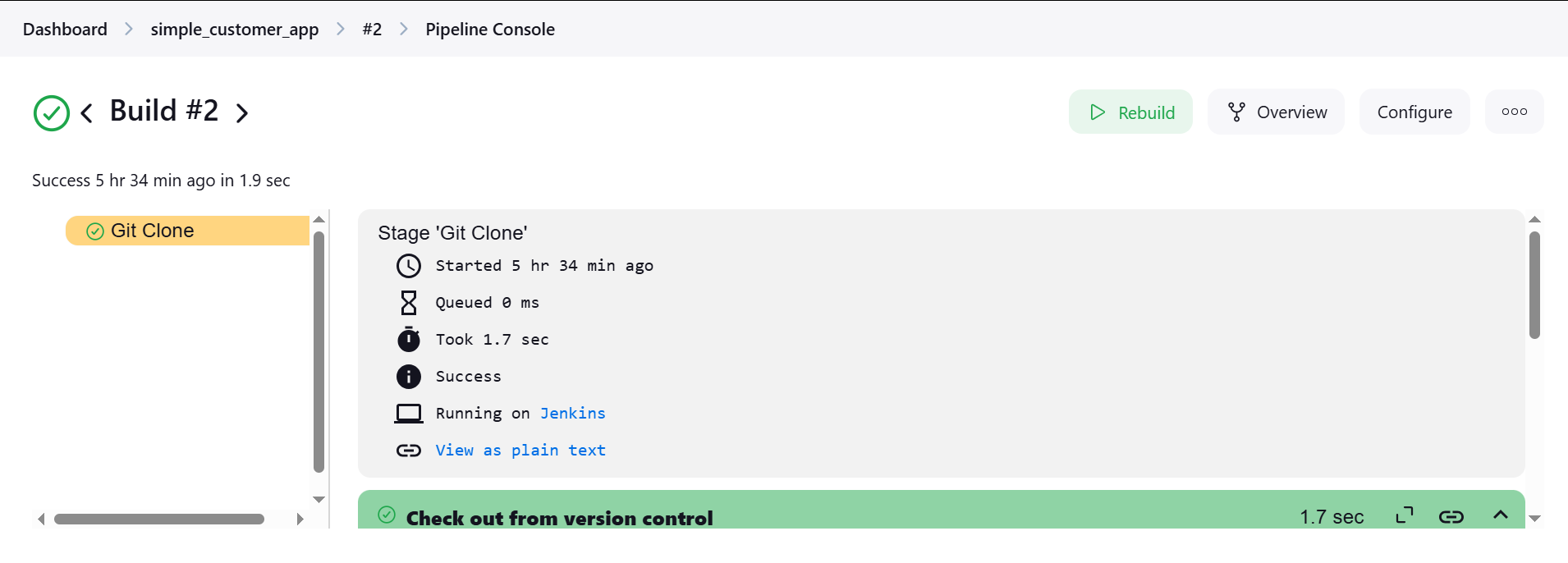
**STEP-4:** Launched an EC2 Instance and installed Tomcat



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

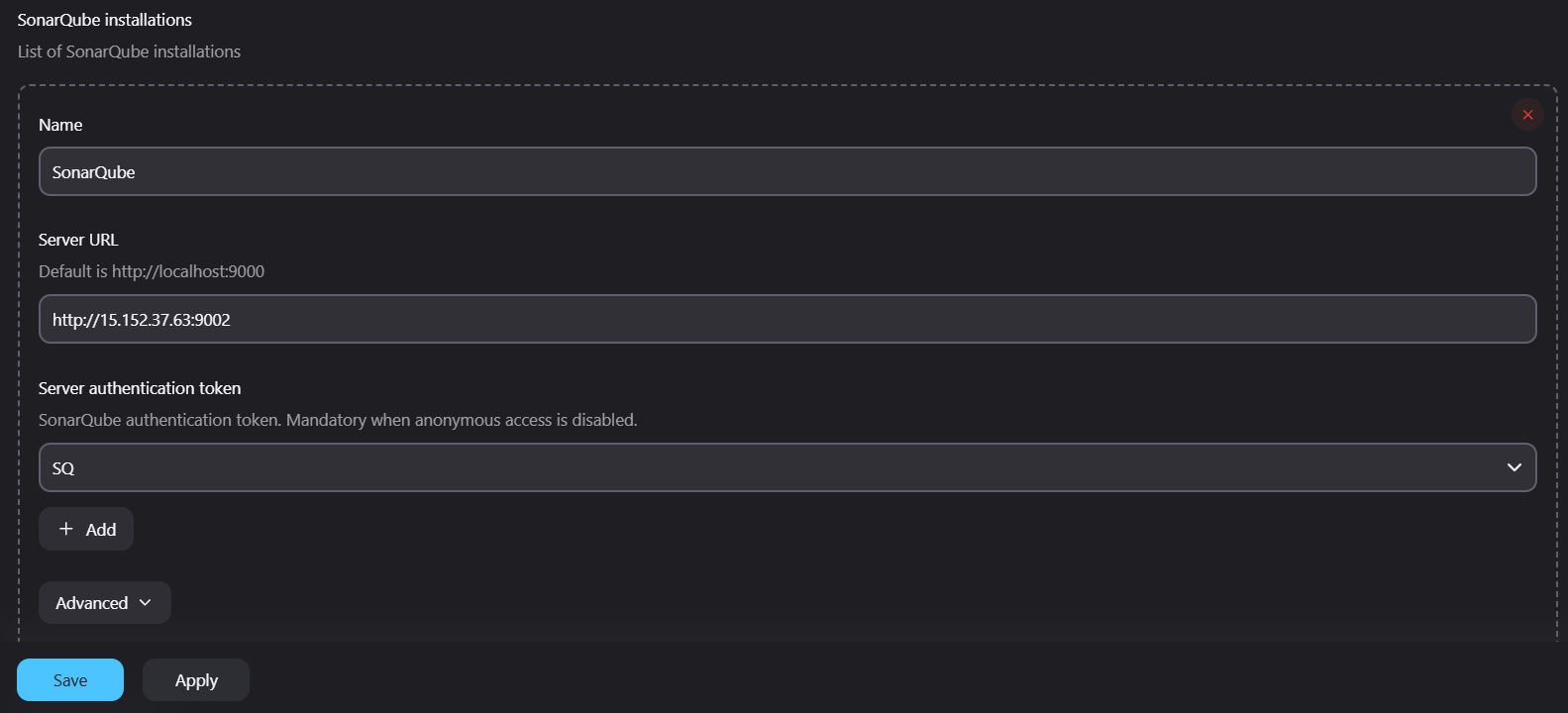
---For this we need git installed in our jenkins server



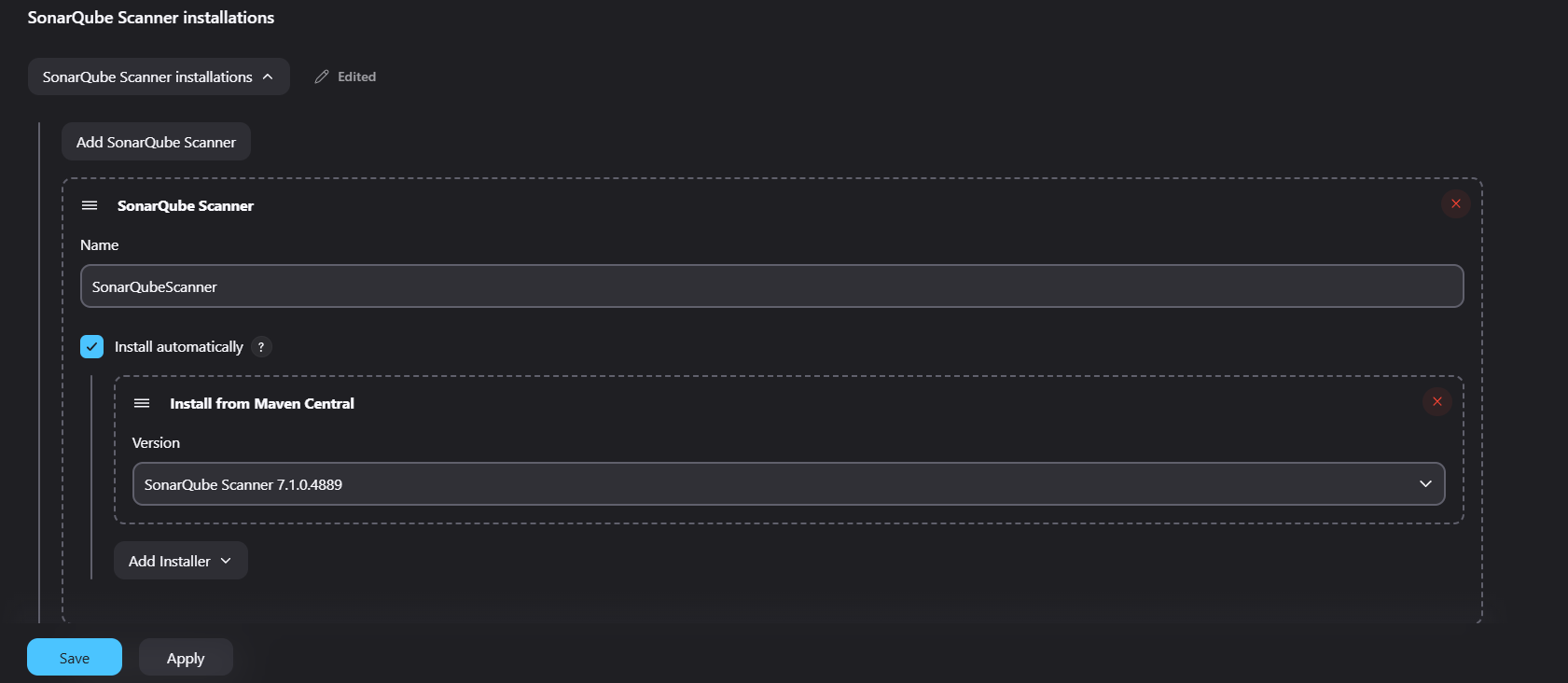


**STEP-6:** Integrating SonarQube with jenkins to Perform Code Analysis

--- configure sonarqube url and credentials in system configurations

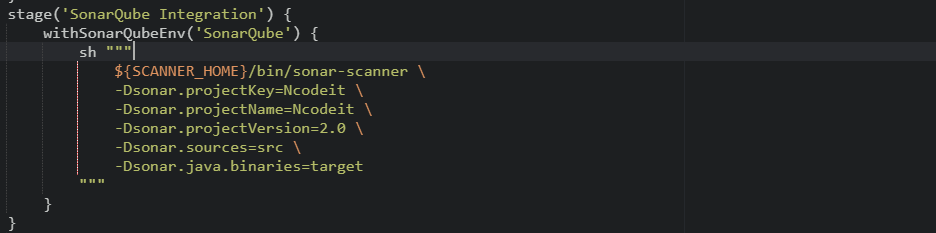


---install sonarqube-scanner plugin and comfigure sonarqube installations for the server in global tool configuration

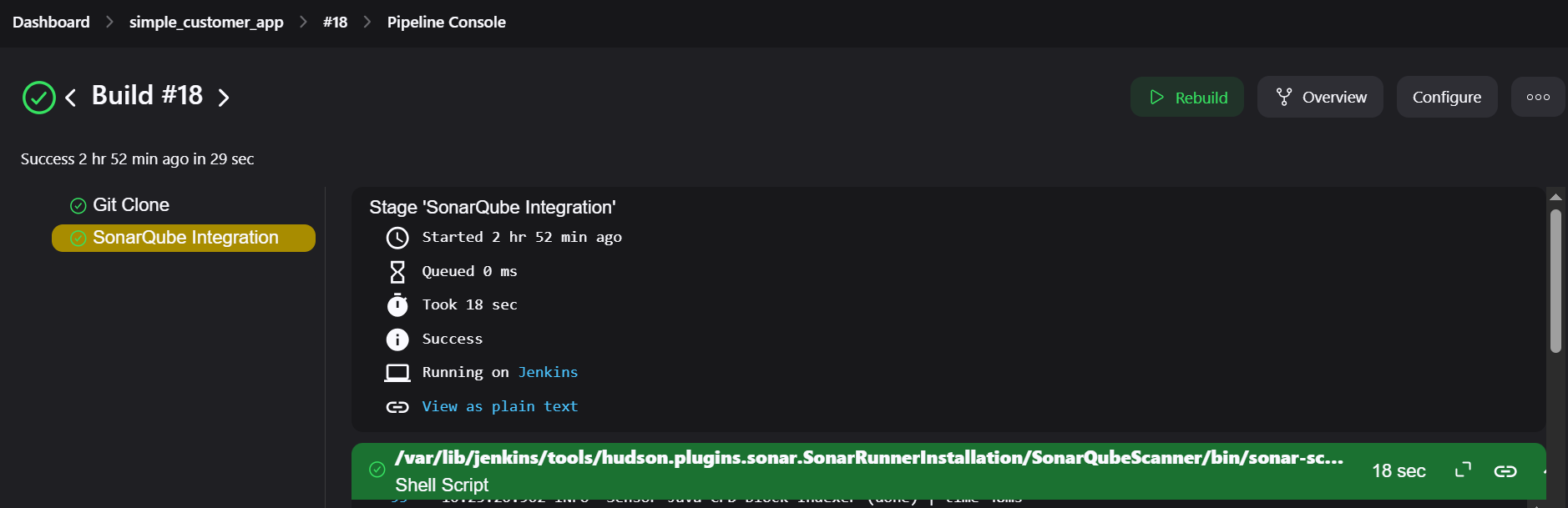


--- script for static code analysis

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

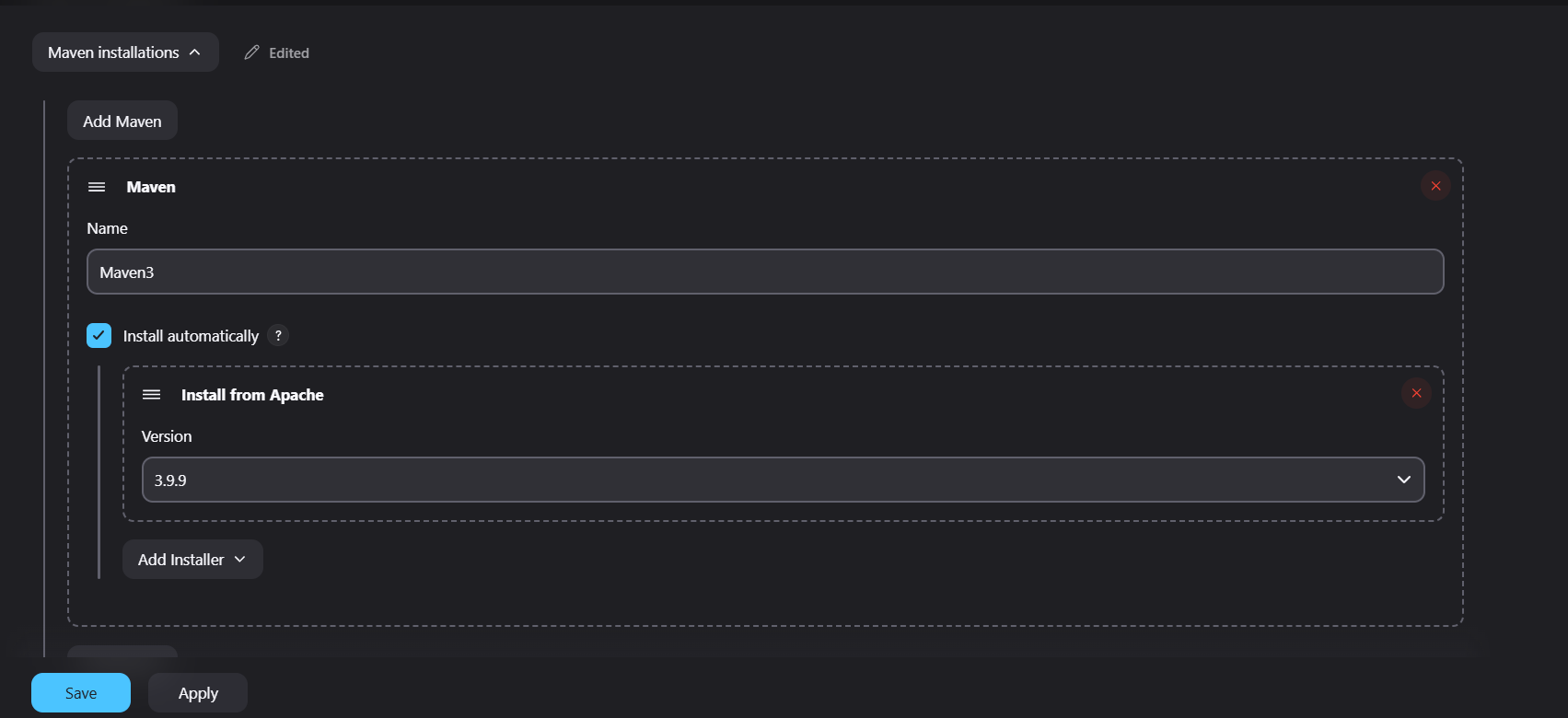


--- build job

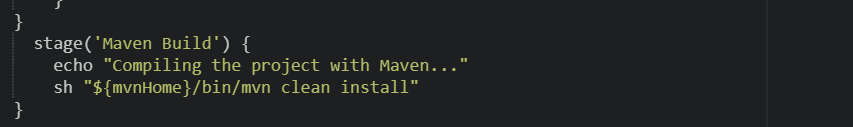


**STEP-7:** Use Maven to compile and package the Java application

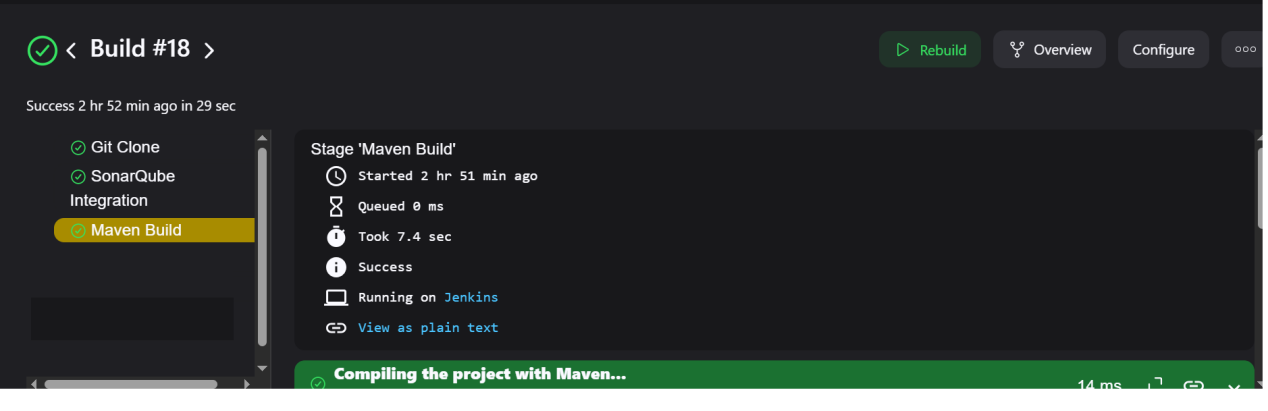
---Install maven-integration plugin and configure in global tool configuration



---Snippet for maven configuration



---Build



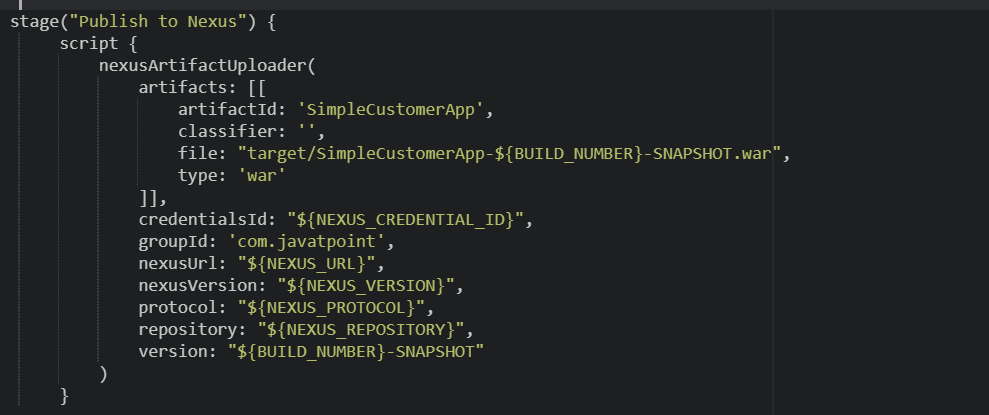
**STEP-8:** Upload the built .war file to Nexus for storage.

--- install nexus-artifact uploader plugin

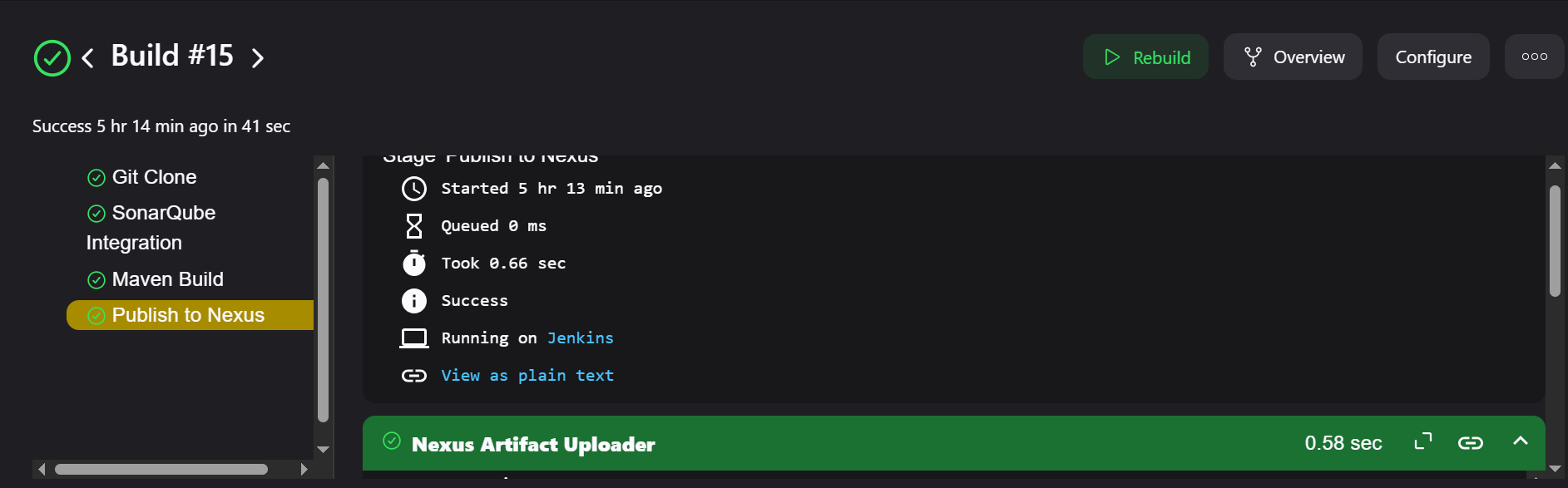
--- create a secret(credential) with username and password

--- create a Repository in nexus

--- Script forNexus configuration



---Build job



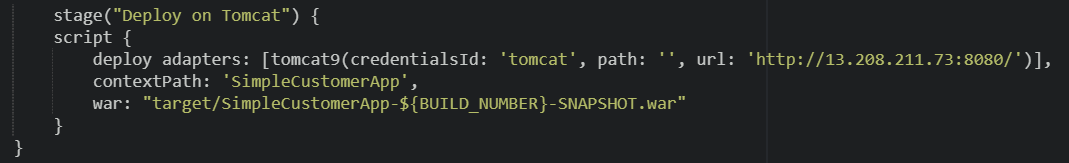
**STEP-9:** Deploy the .war file to the Tomcat application server.

--- Install deploy to container plugin

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

--- Create a secret(Credential) with tomcat URL

---Script for tomcat deployment



---Build job

