



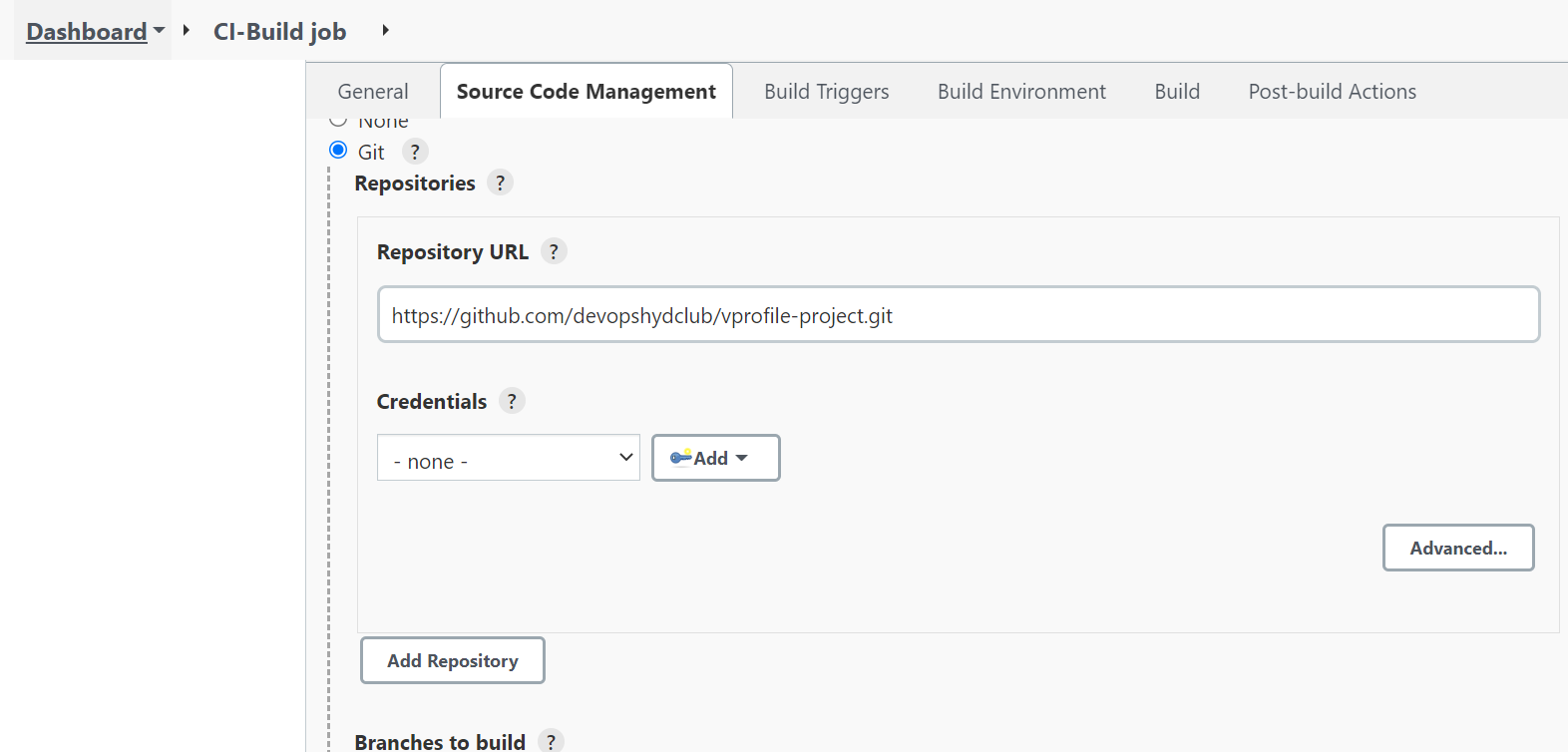
Servers involved:

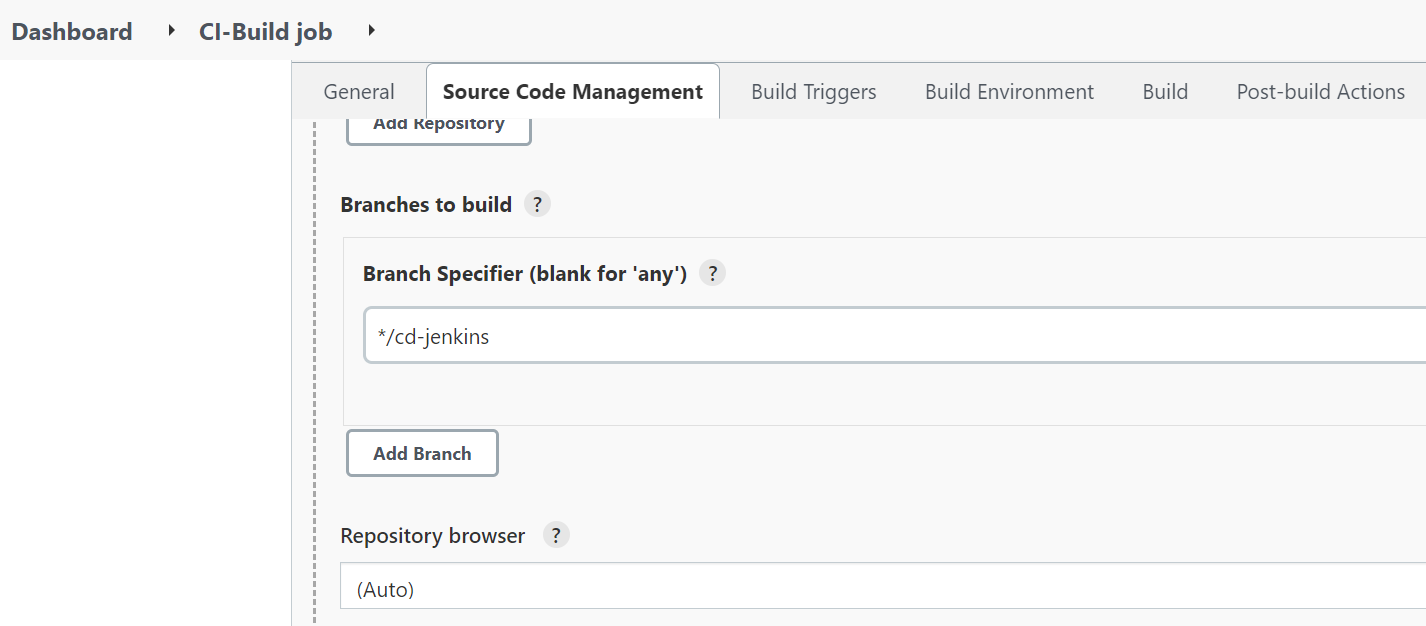
Graphical user interface, application

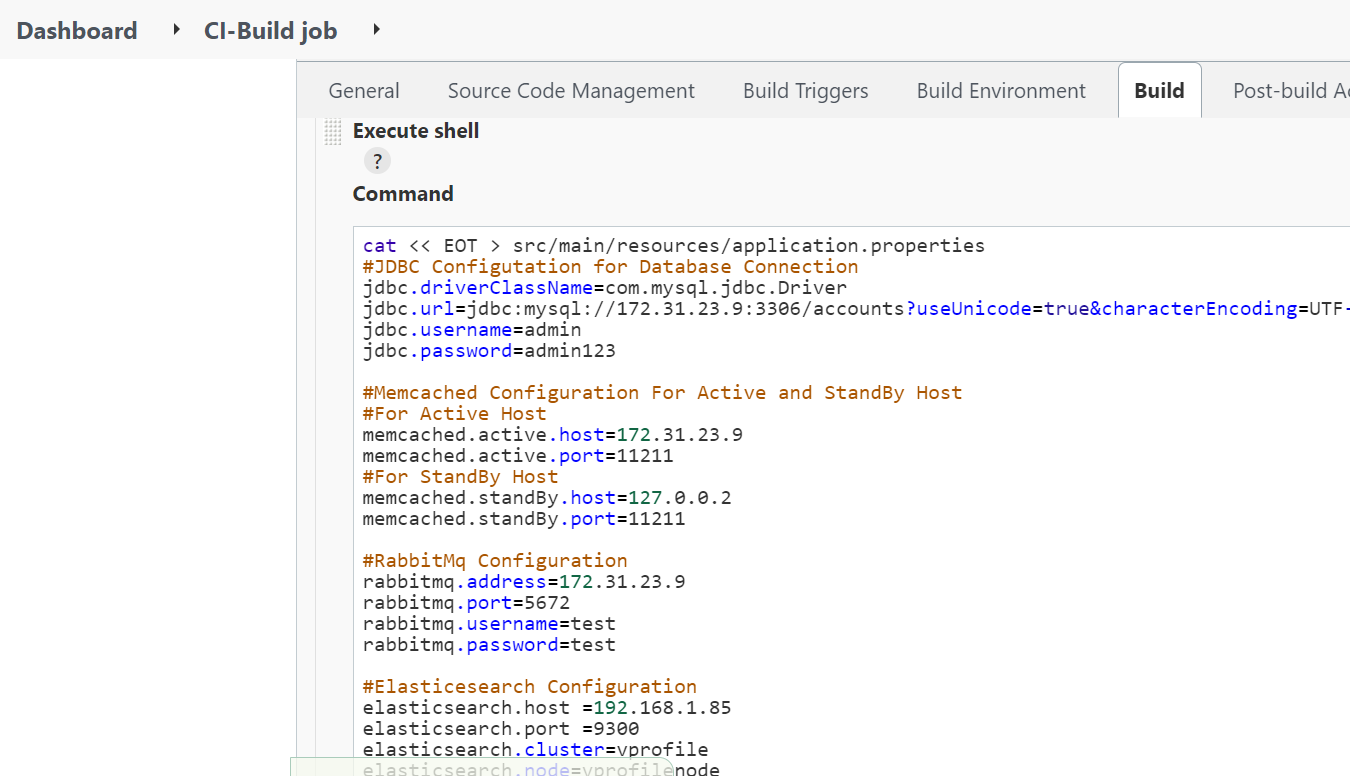
Description automatically generated

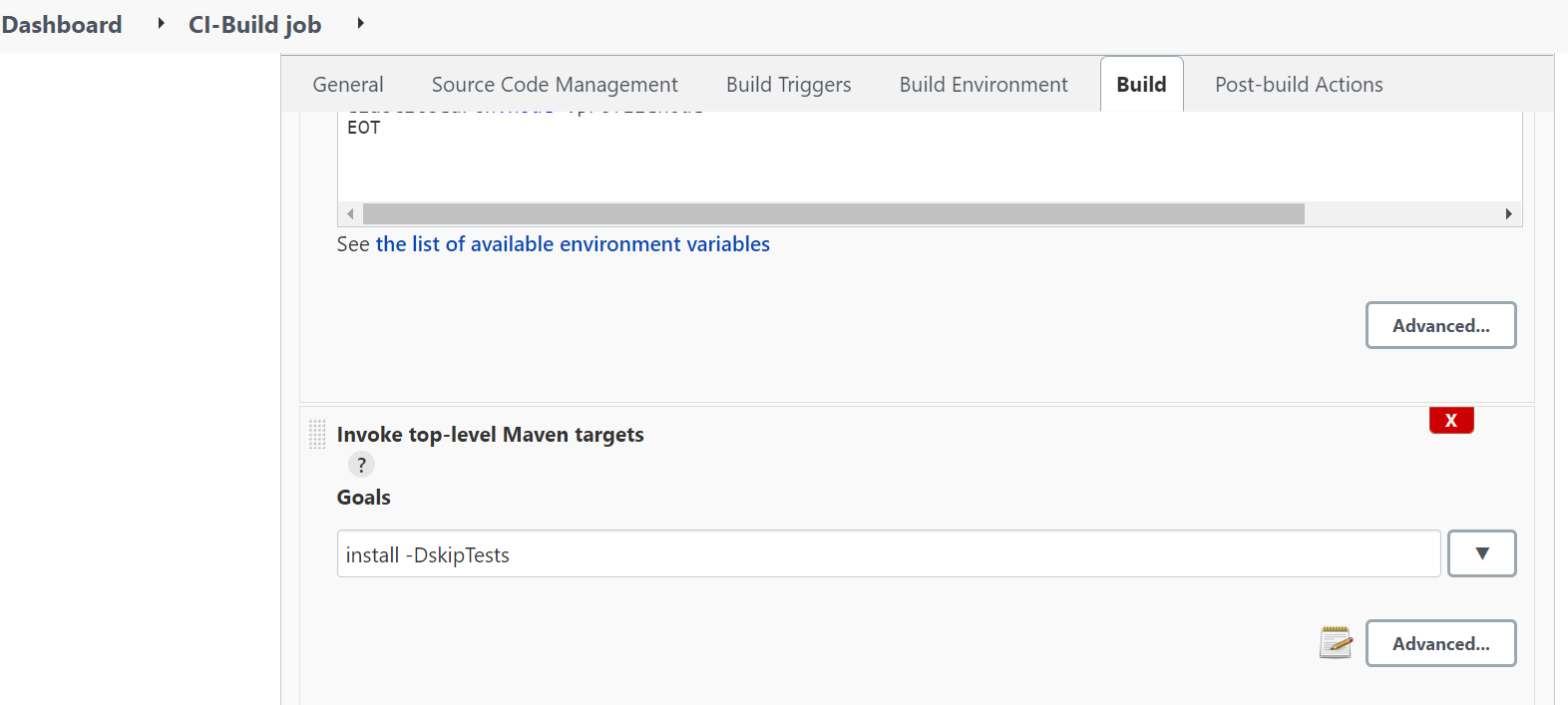
Step1: Build Job

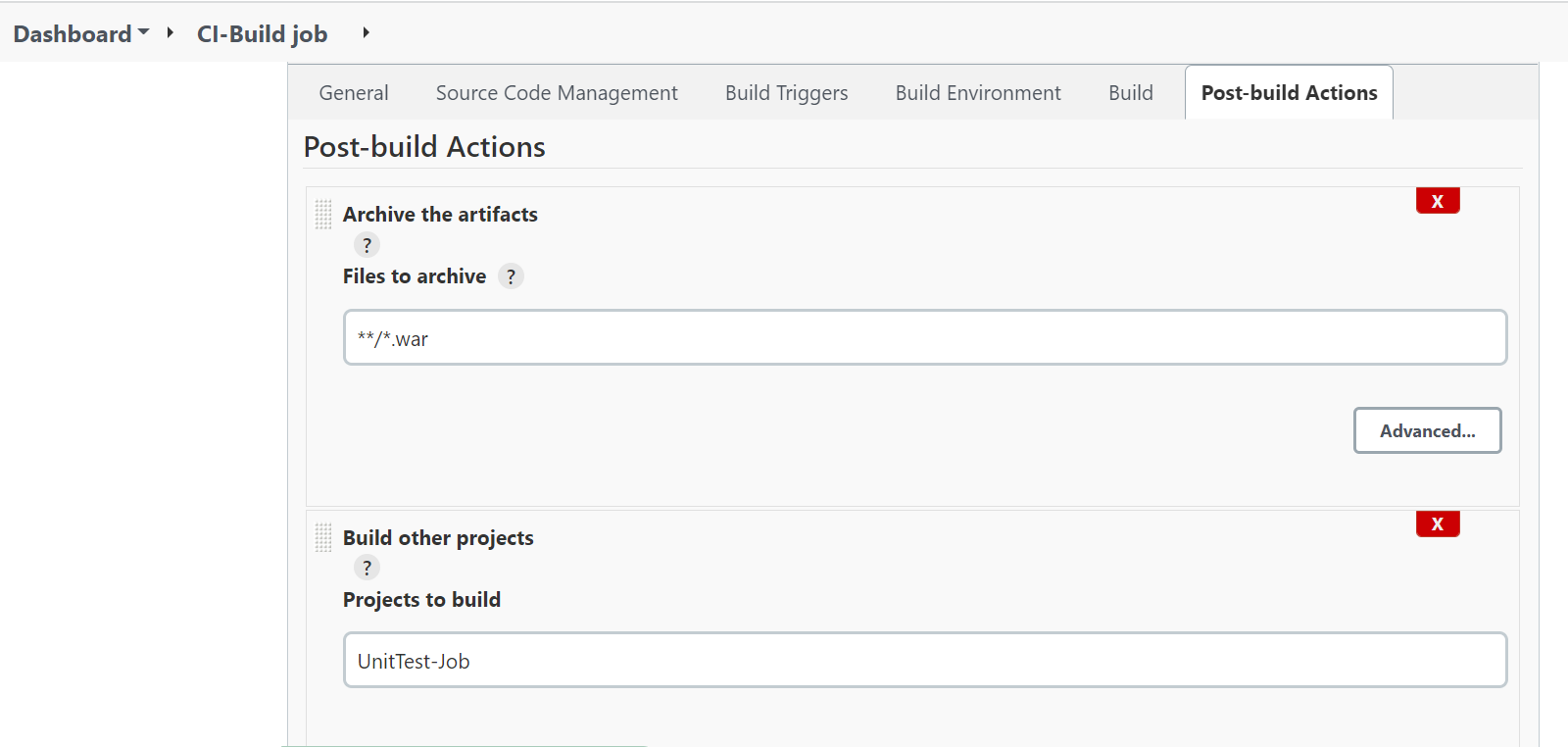
Fetch the code from git hub and generate artifact

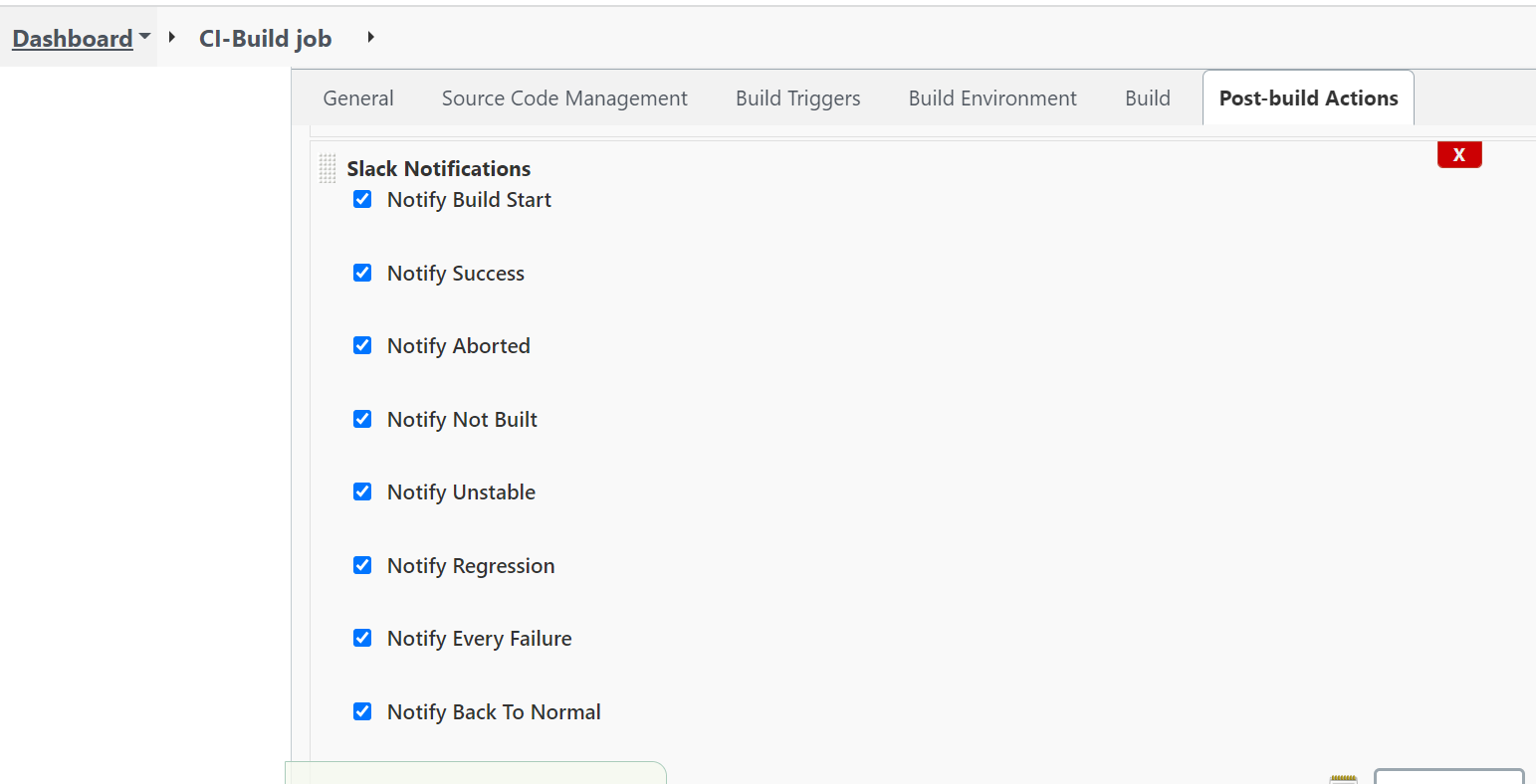








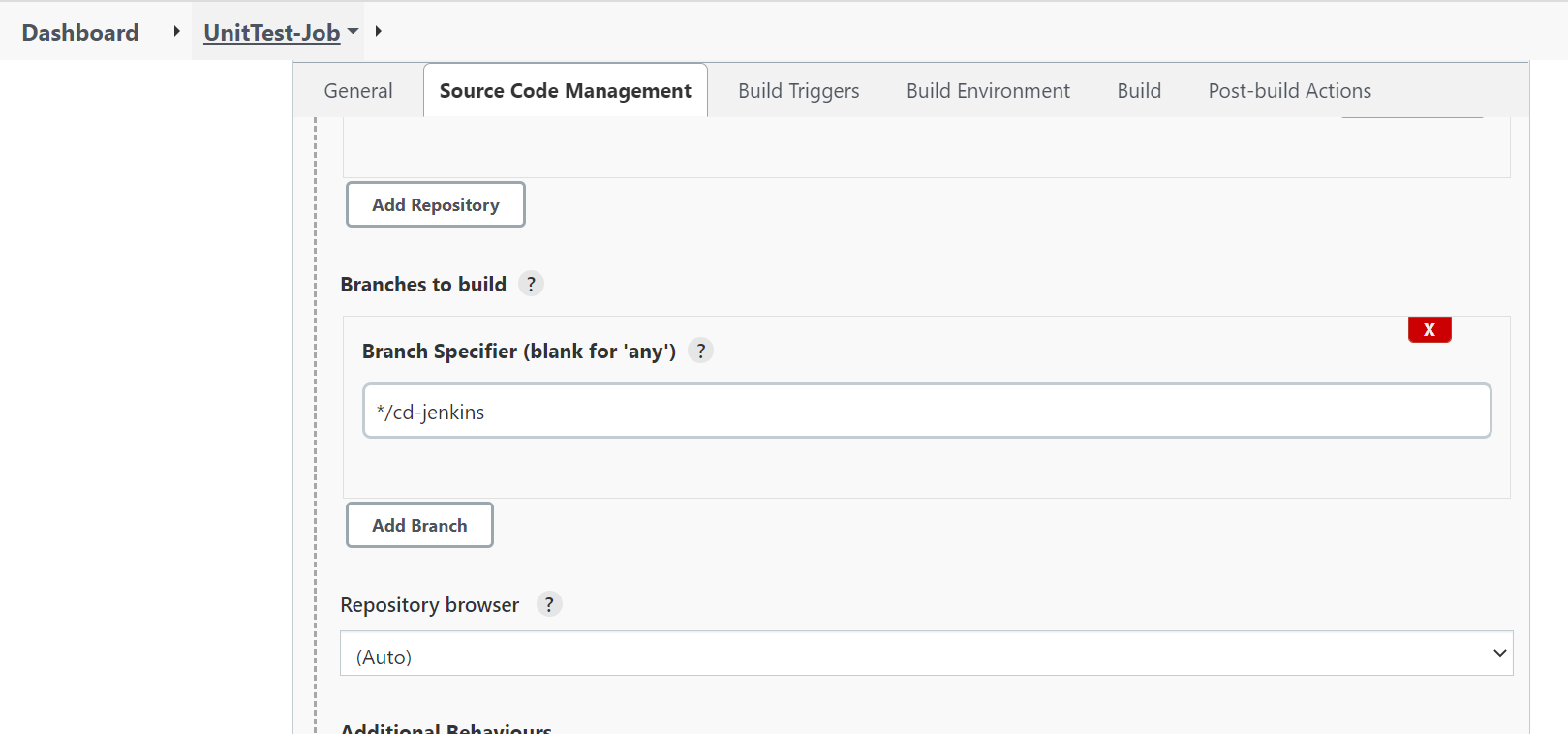


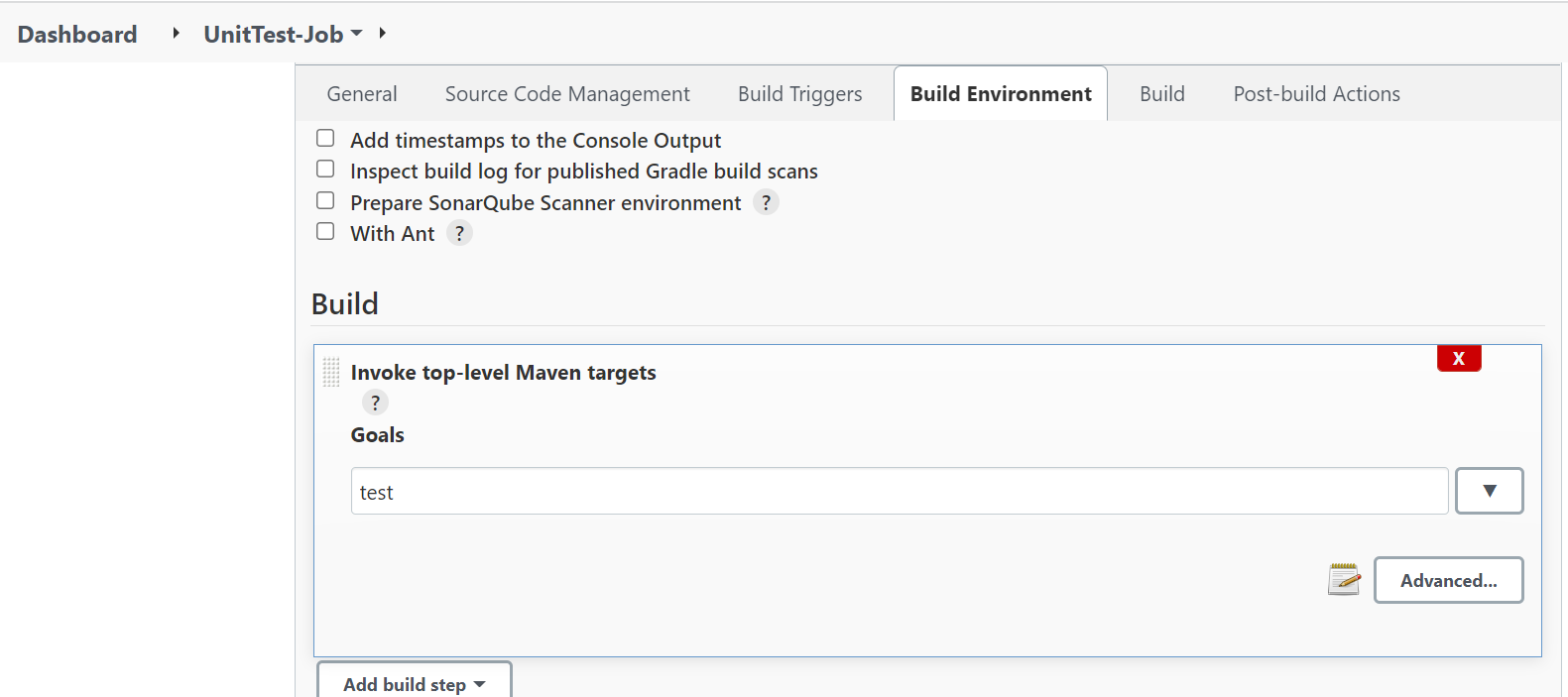


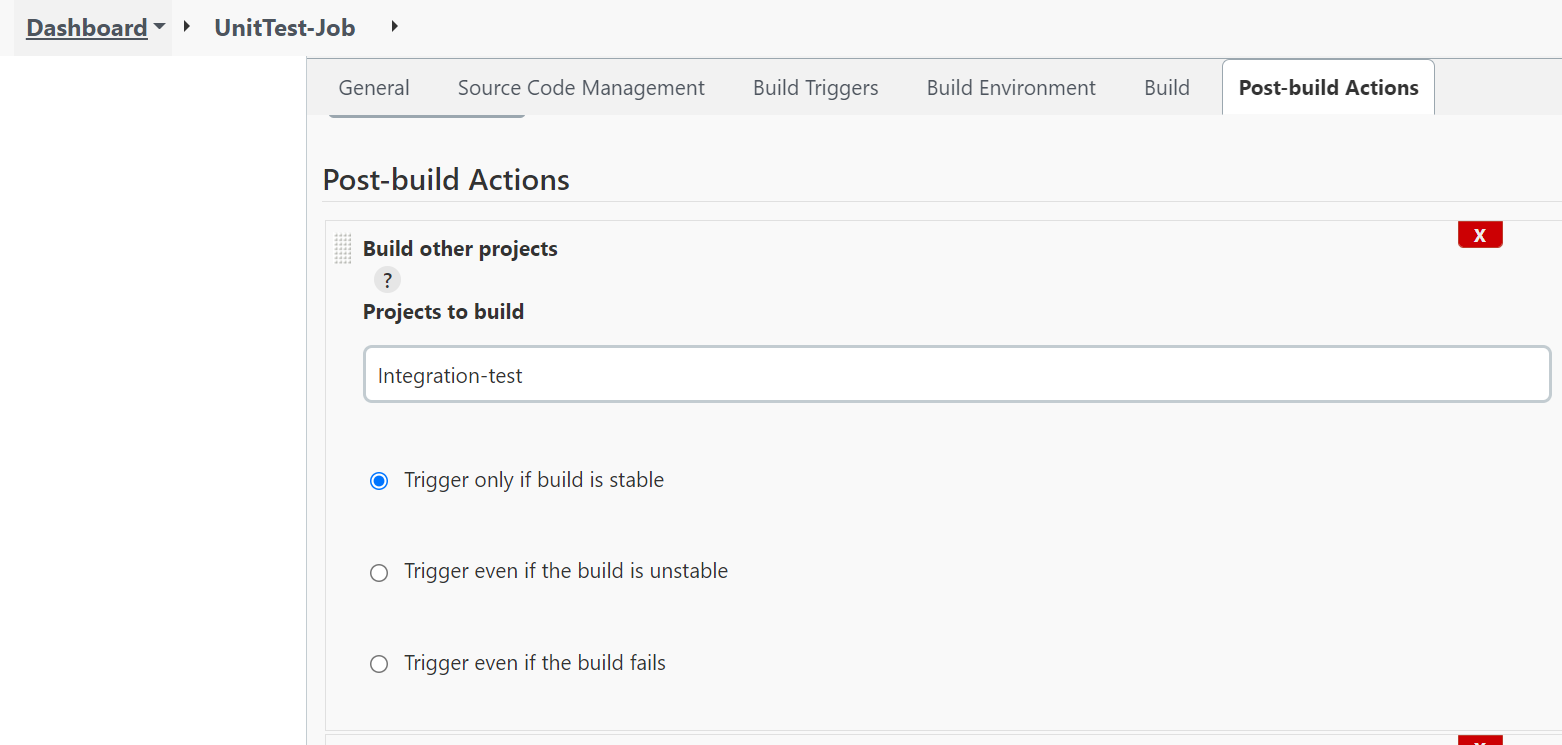
Specify workspace, channel, workspace(invite a bot to the created channel)

Step2: Unit test job

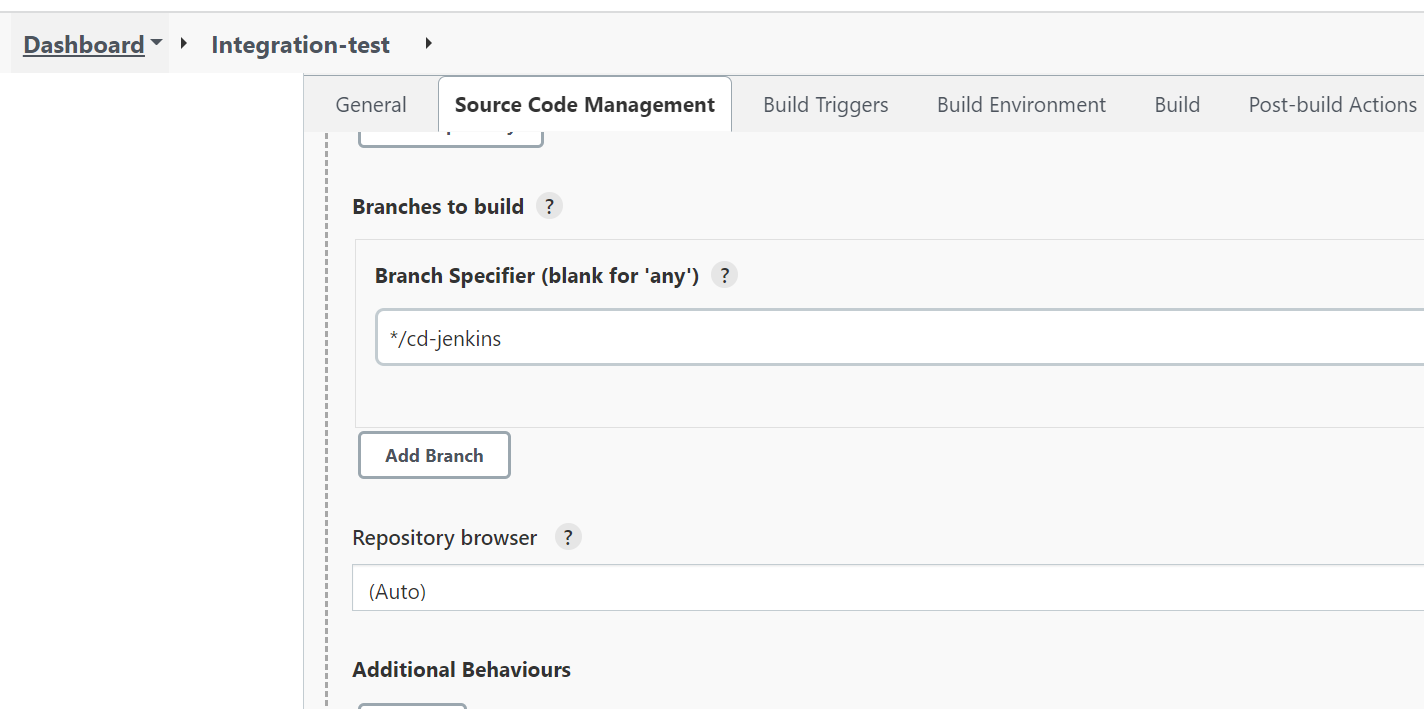
perform unit testing using maven

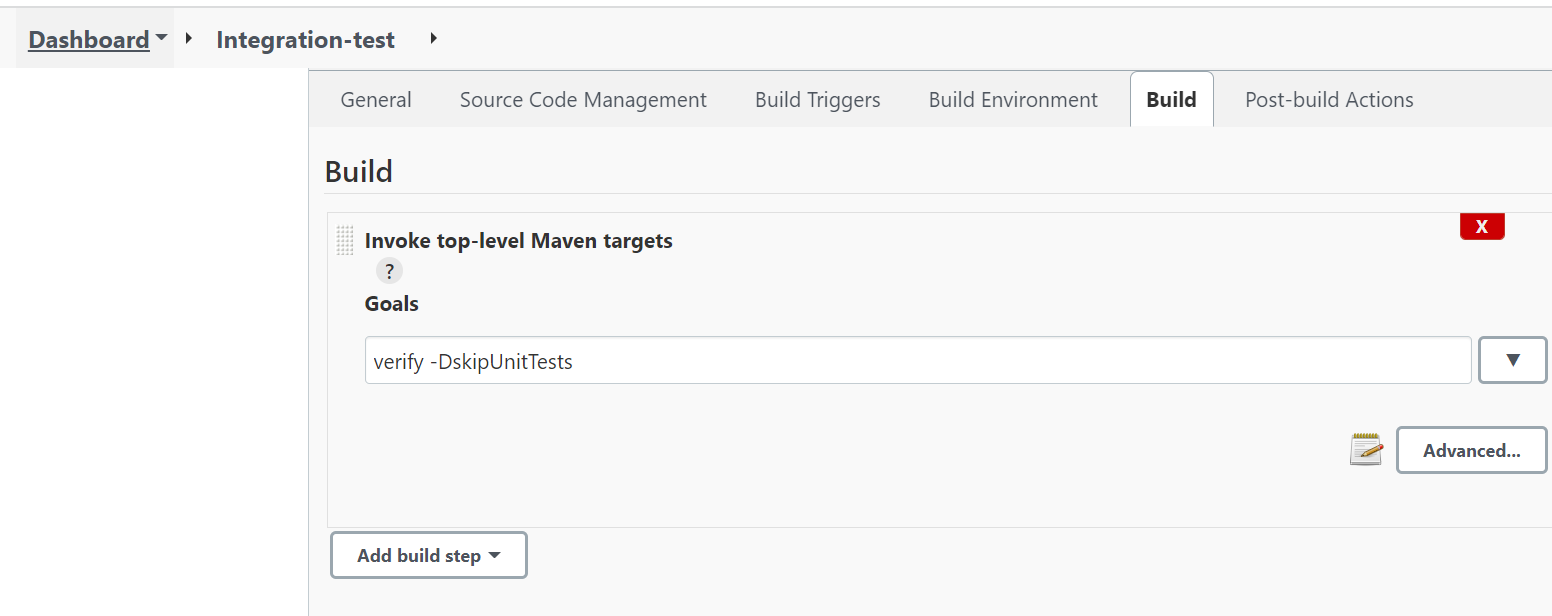


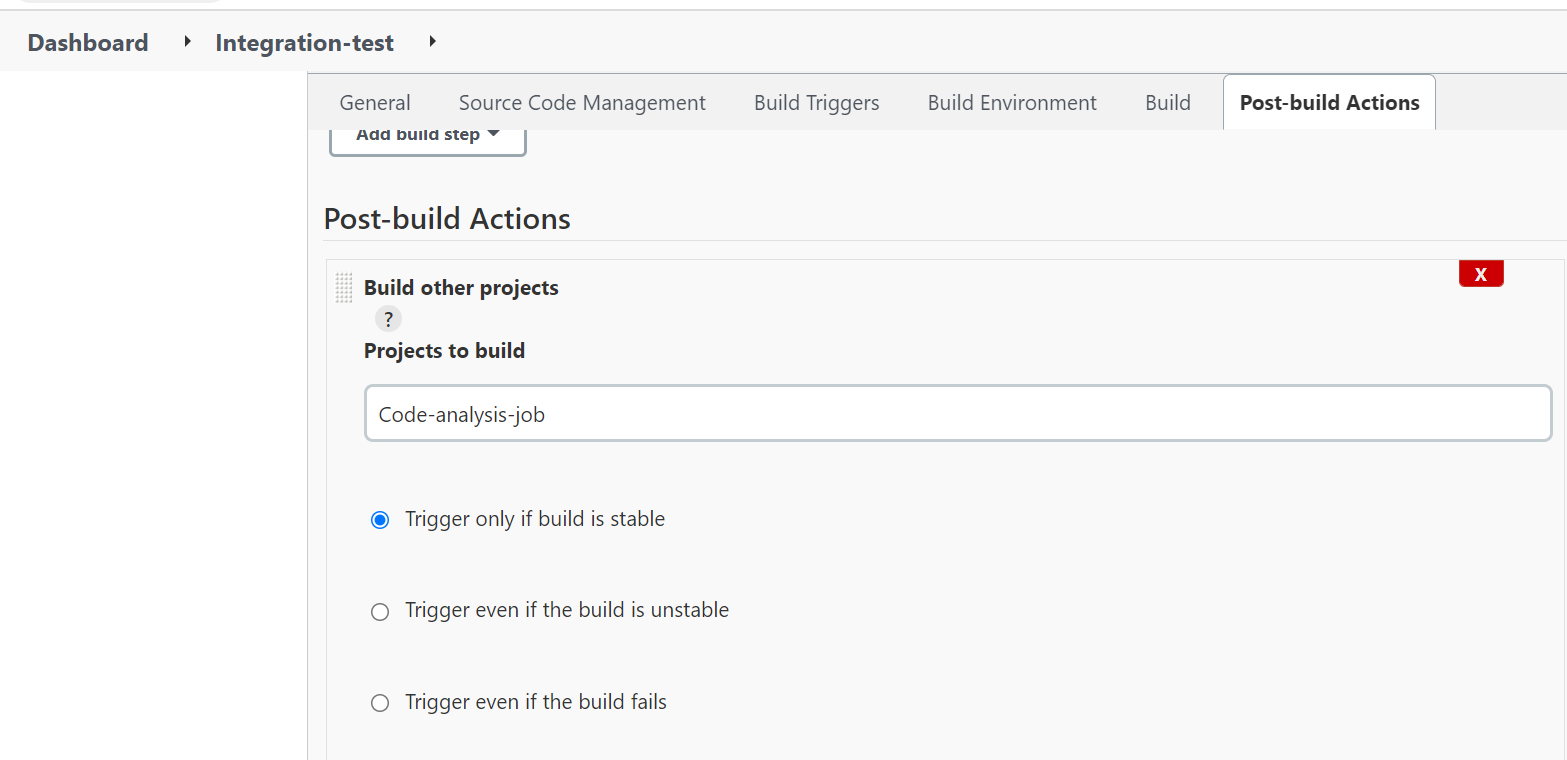




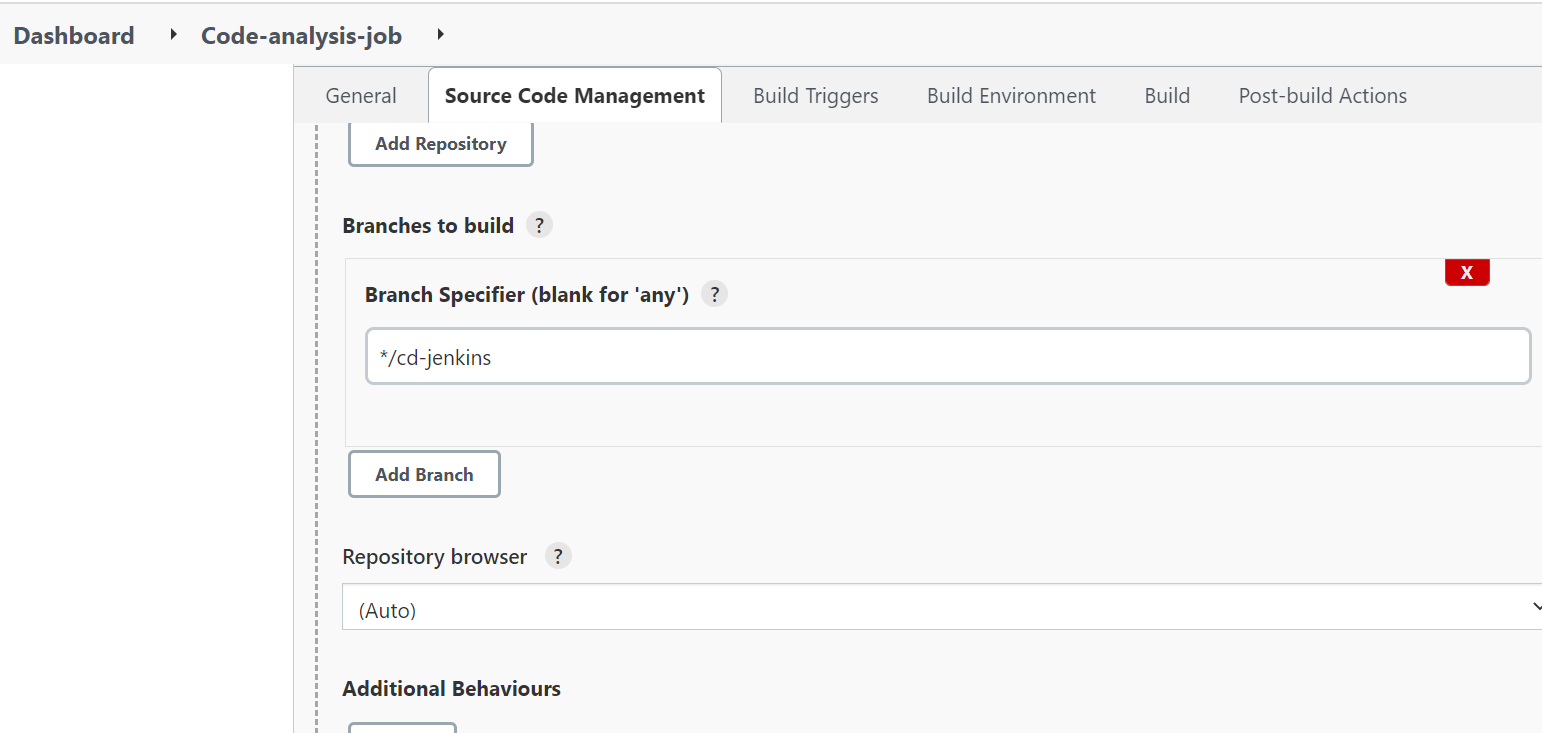
Step 3: Perform integrated testing with maven

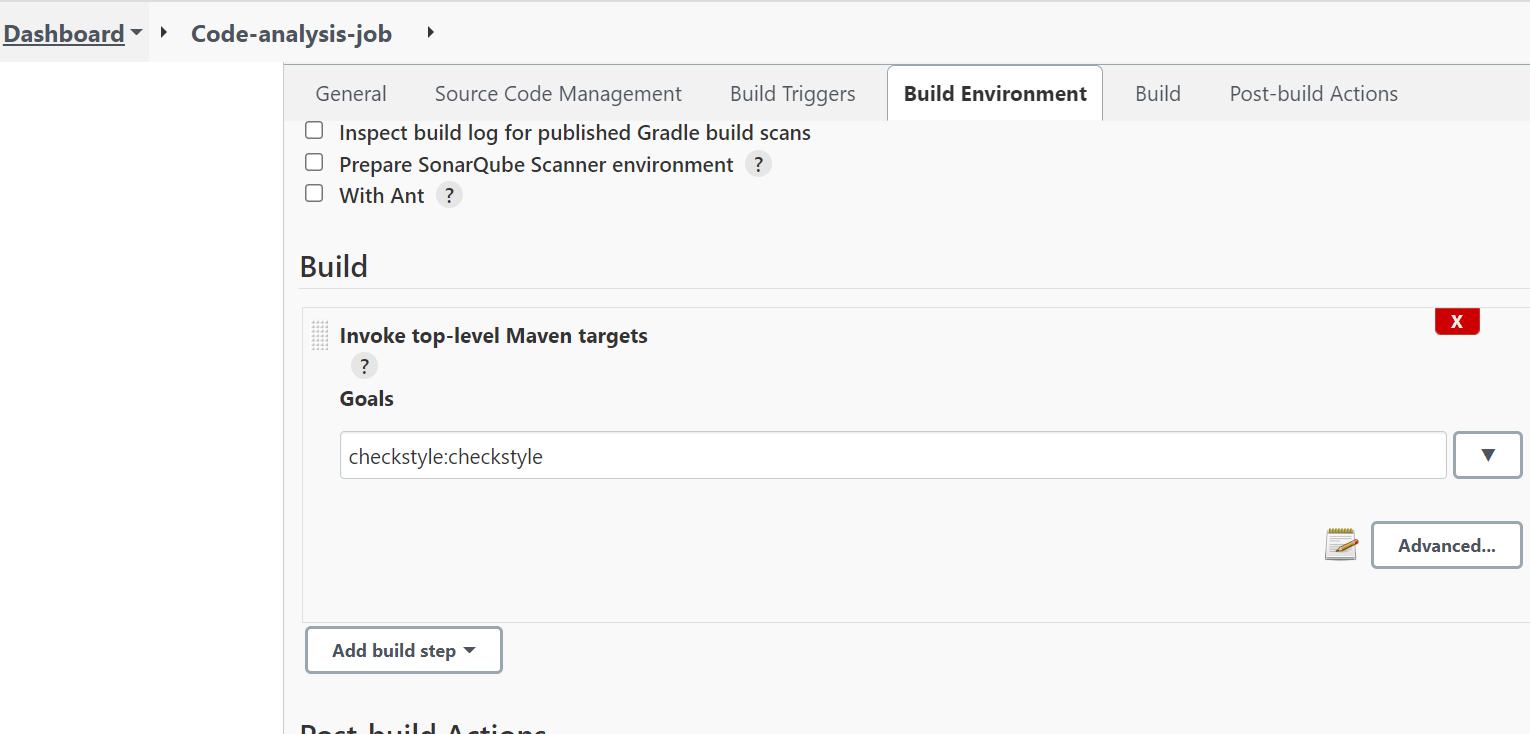


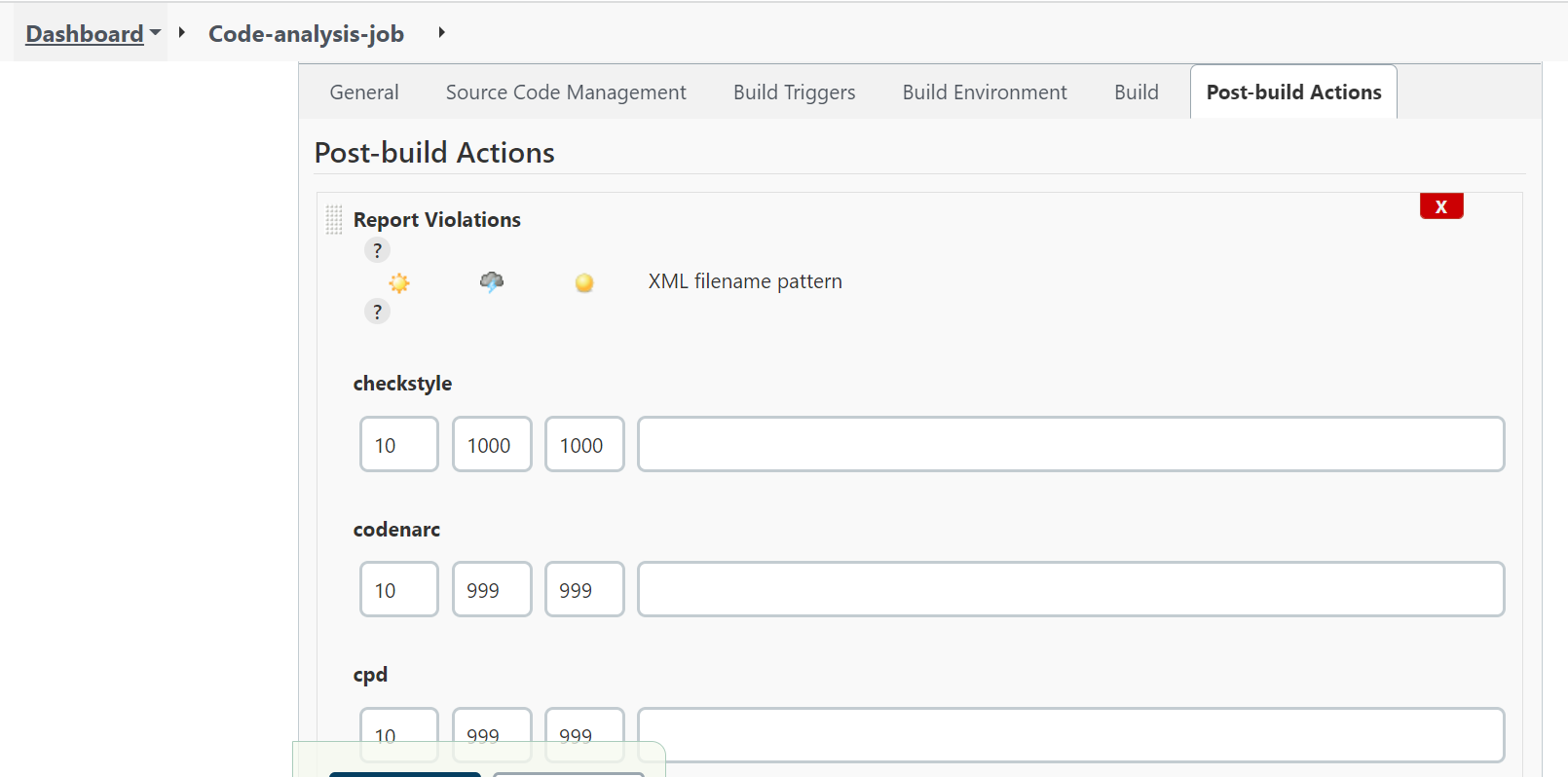


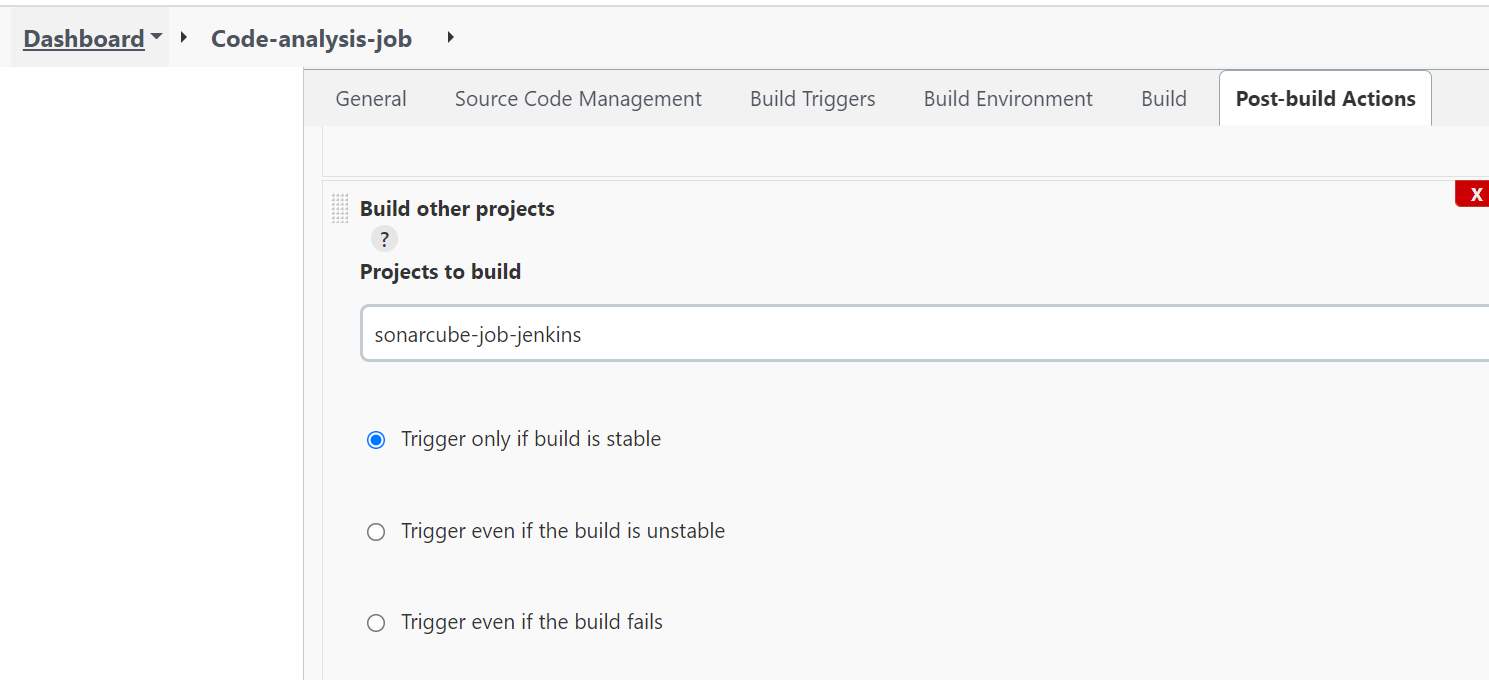


Step 4: Perform code analysis using checkstyle



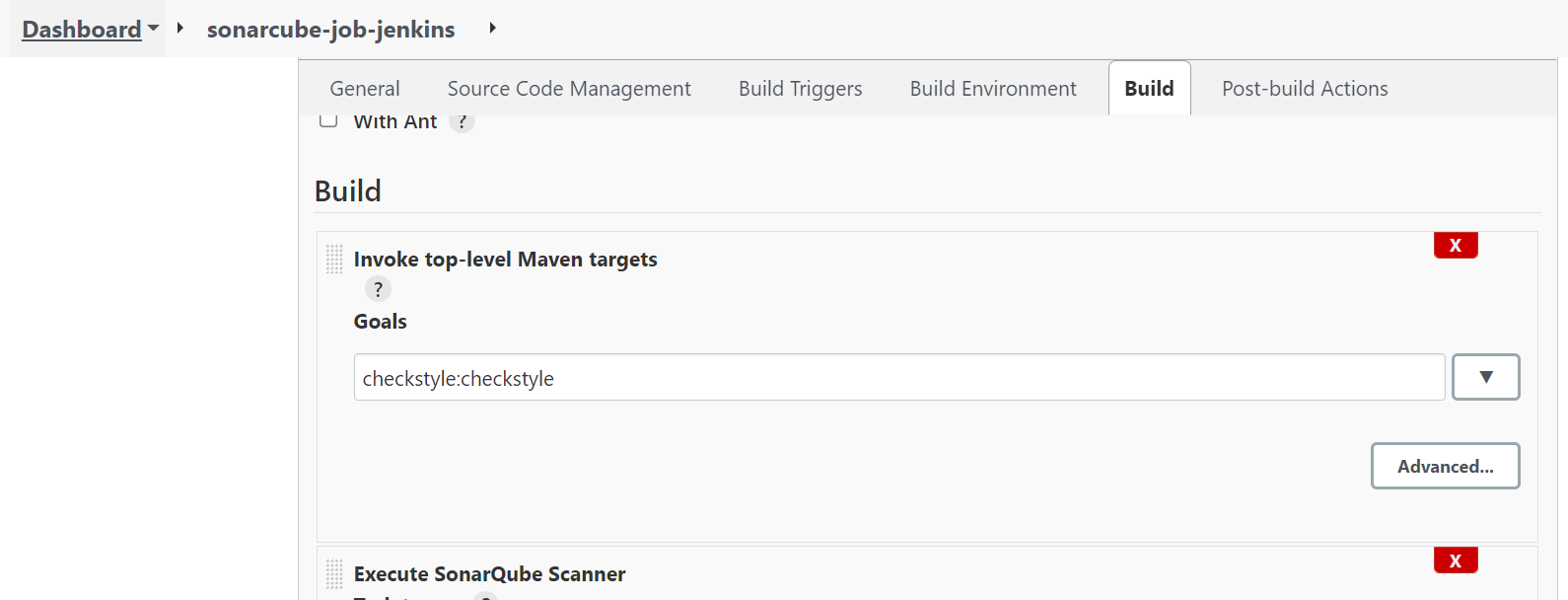


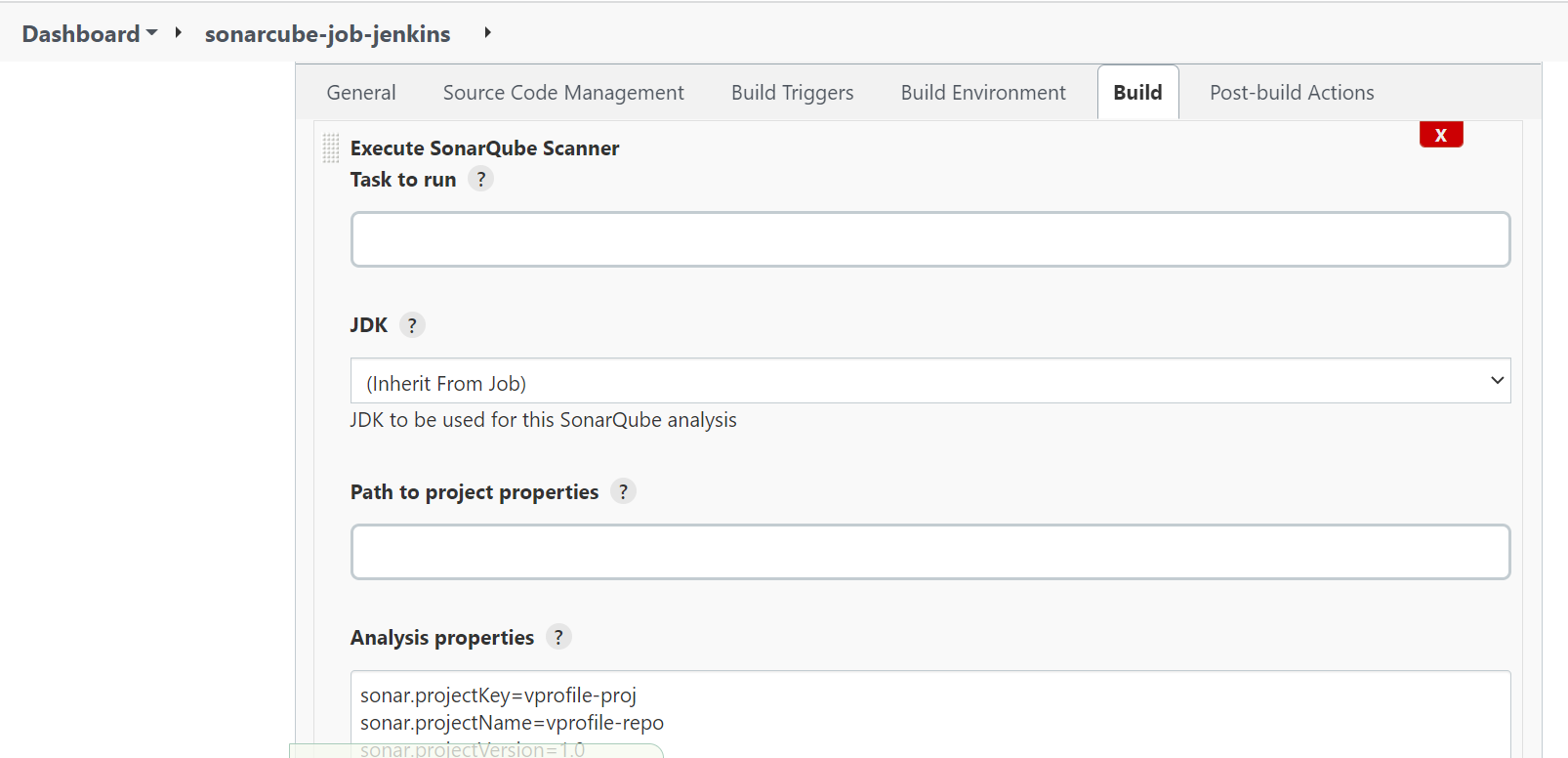




Step 5: Perform code analysis and quality gate checking with sonarqube







(In configure system -> install sonarcube installer,download quality gates and sonarcube plugins

sonar.projectKey=vprofile-proj

sonar.projectName=vprofile-repo

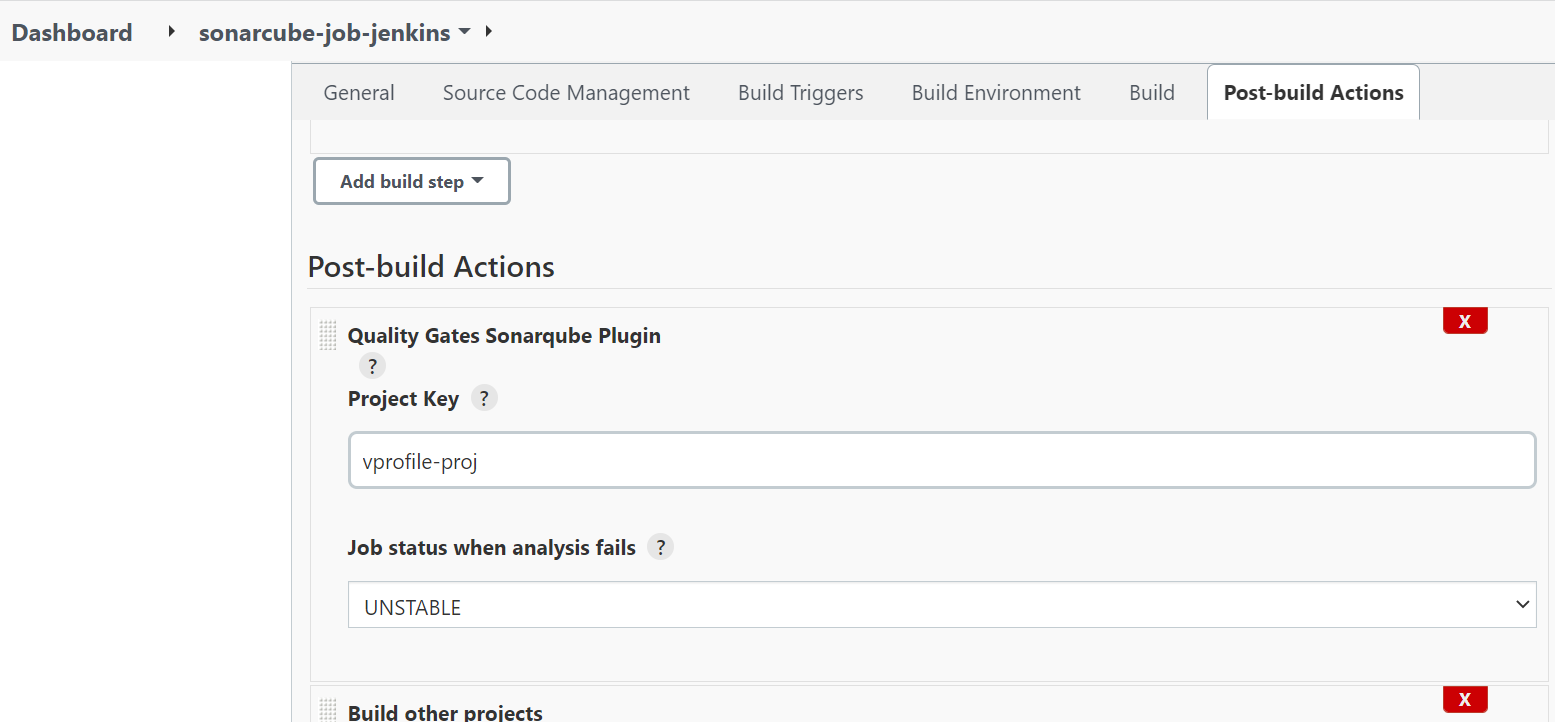
sonar.projectVersion=1.0

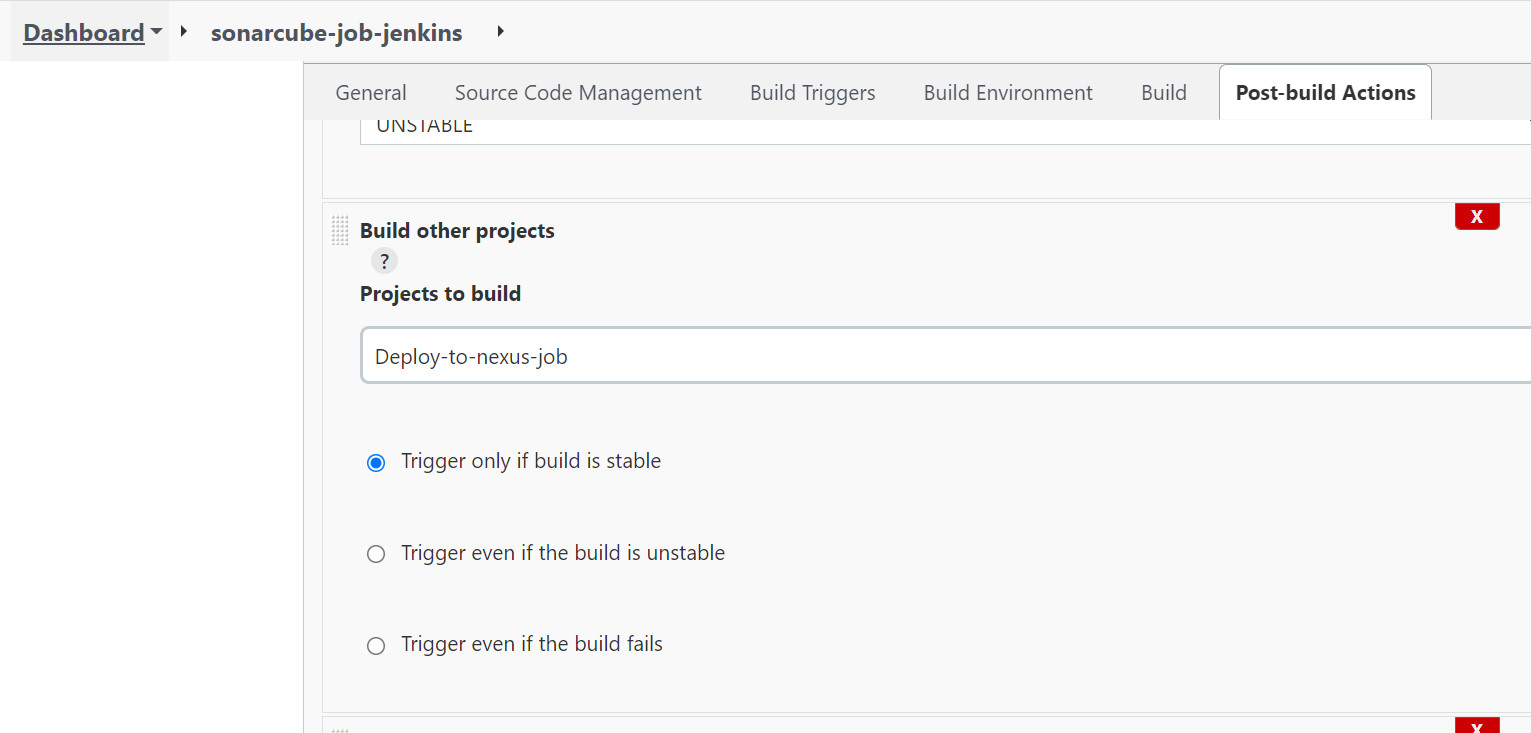
sonar.sources=src/

sonar.java.binaries=target/test-classes/com/visualpathit/account/controllerTest/

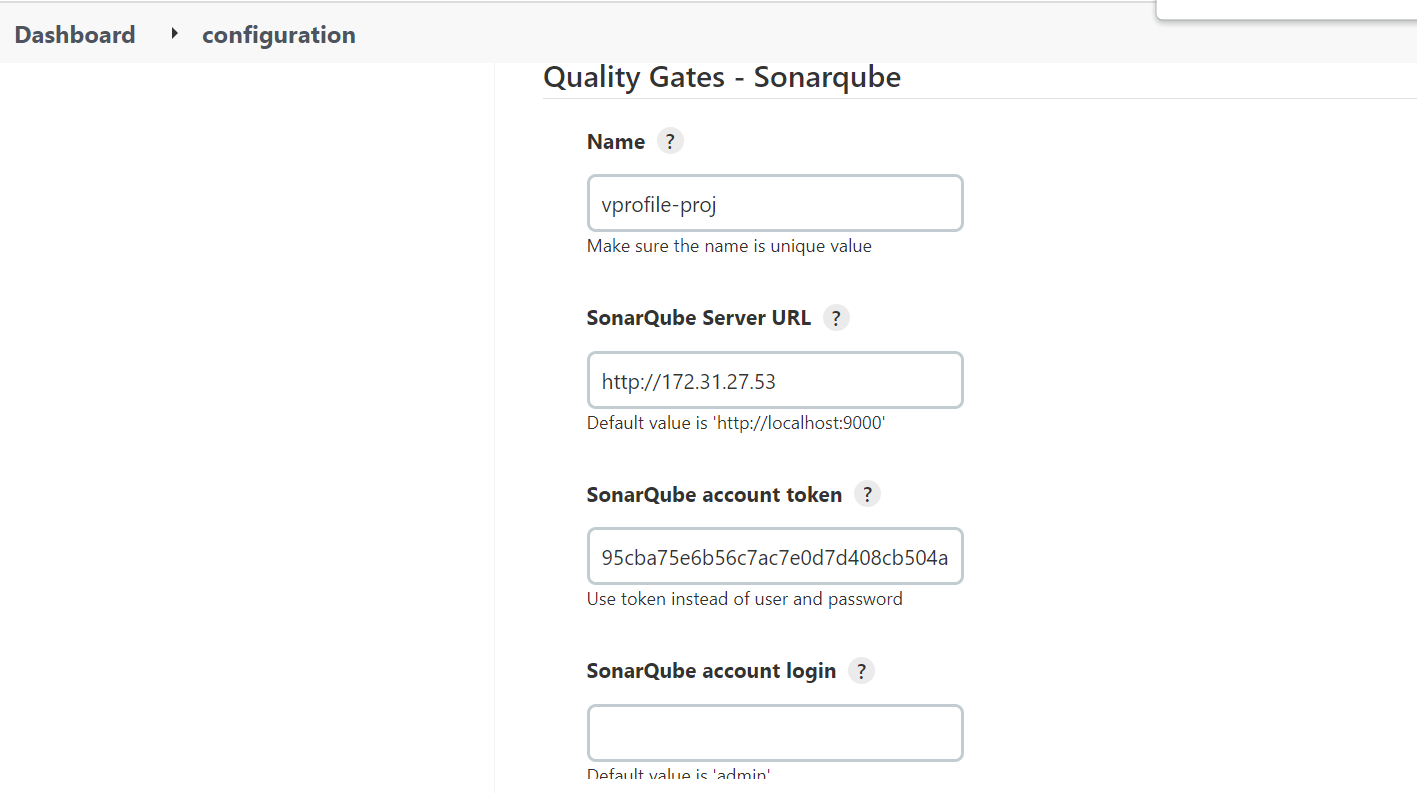
sonar.junit.reportsPath=target/surefire-reports/

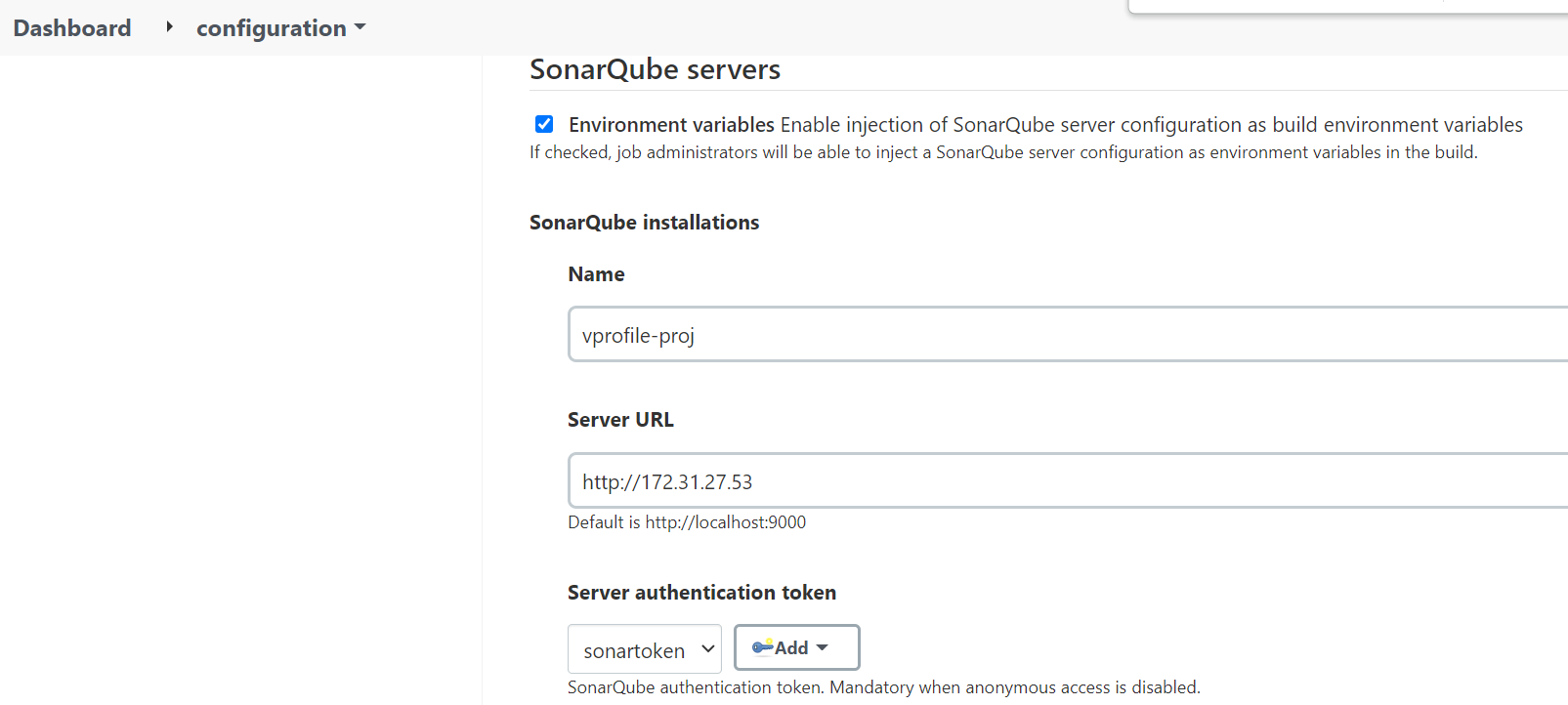
sonar.jacoco.reportsPath=target/jacoco.exec

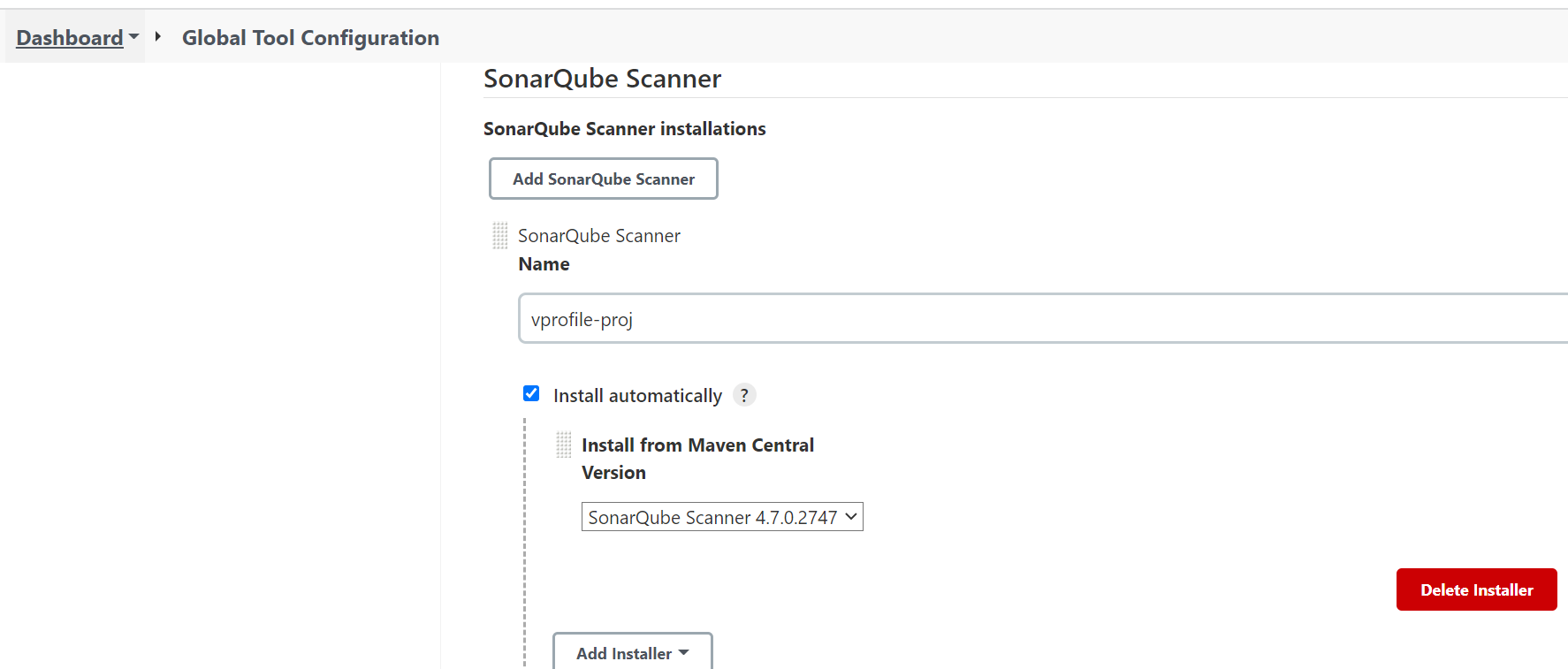
sonar.java.checkstyle.reportPaths=target/checkstyle-result.xml



Sonarqube global jenkins Configurations:







Graphical user interface, text, application, chat or text message

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Step 6: Upload artifact to nexus repository

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface

Description automatically generated

Graphical user interface, text

Description automatically generated

Step 7: Deploying artifact to staging environment(tomcat)

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

A picture containing table

Description automatically generated

Table

Description automatically generated

Step 8: Test URL job – to check if the URL is up and running

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Step 9: Configure a windows slave node and run selenium script to check if web application can be logged in

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

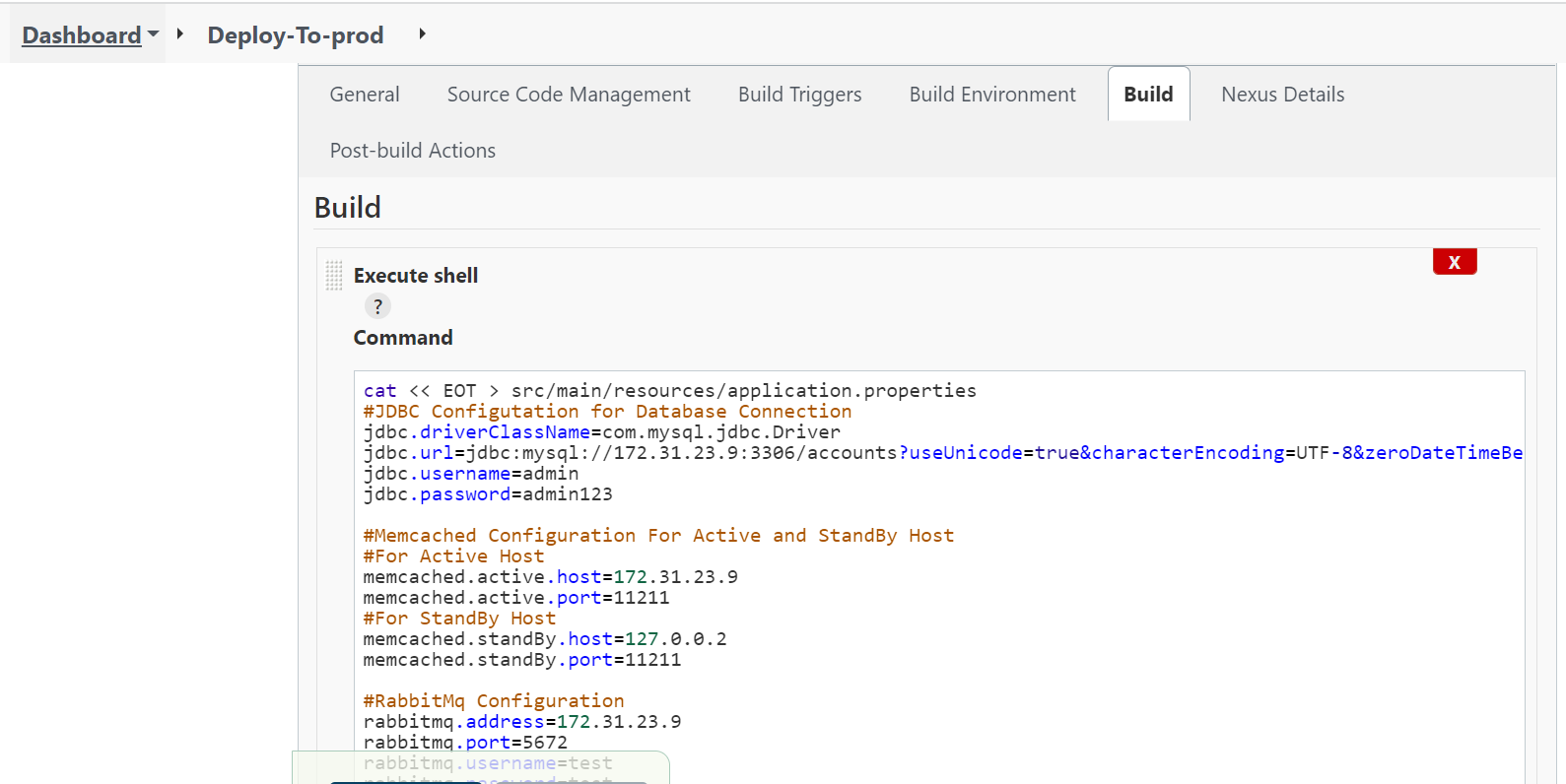
Description automatically generated

Graphical user interface, text, application, table

Description automatically generated



Step 9: Finally when all steps are successful, deploy the artifact to the production environment



Graphical user interface, application, Teams

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Overall integration of all jobs with slack

