

DEVOPS PROFESSIONAL CERTIFICATION PROGRAM

Lab – 7: Code Quality Check - SonarQube

- Create Maven Project
- Configure SonarQube Scanner
 - Perform Static Code Analysis
 - Perform Code Coverage
- Orchestrate Code Review process using pom.xml / Jenkins Plugins



- Use any maven project created in previous Lab
- Create Jenkins Job – Maven Project

The screenshot shows the 'Enter an item name' dialog box in Jenkins. The text input field contains '6 Lab'. Below the input field, there is a small text indicating '» Required field'. Below the input field, there are two options: 'Freestyle project' and 'Maven project'. The 'Maven project' option is selected and highlighted with a blue border. The description for 'Maven project' reads: 'Build a maven project. Jenkins takes advantage of your POM files and drastically reduces the configuration.'

- Configure SCM

The screenshot shows the 'Source Code Management' configuration page in Jenkins. The 'Git' radio button is selected. Under 'Repositories', the 'Repository URL' is set to '/home/osgdev/ap/GettingStarted/mavenProcess/Demo1'. The 'Credentials' dropdown is set to '- none -'. There is an 'Add Repository' button. Under 'Branches to build', the 'Branch Specifier (blank for 'any')' is set to '*/master'. There is an 'Add Branch' button. The 'Repository browser' is set to '(Auto)'. At the bottom, there are 'Save', 'Apply', and 'Add' buttons.

- Configure SonarQube Scanner within pre-Steps section

The screenshot shows the 'Pre Steps' configuration page in Jenkins. The 'Add pre-build step' dropdown menu is open, showing a list of options. The option 'Build a Visual Studio project or solution using MSBuild' is highlighted. Other options include 'Execute Python script', 'Execute SonarQube Scanner', 'Execute Windows batch command', 'Execute shell', 'Invoke Ansible Ad-Hoc Command', 'Invoke Ansible Playbook', 'Invoke Ansible Vault', 'Invoke Ant', 'Invoke Gradle script', 'Invoke top-level Maven targets', 'Provide Configuration files', 'SonarQube Scanner for MSBuild - Begin Analysis', and 'SonarQube Scanner for MSBuild - End Analysis'. The 'Run regardless of build result' radio button is selected.

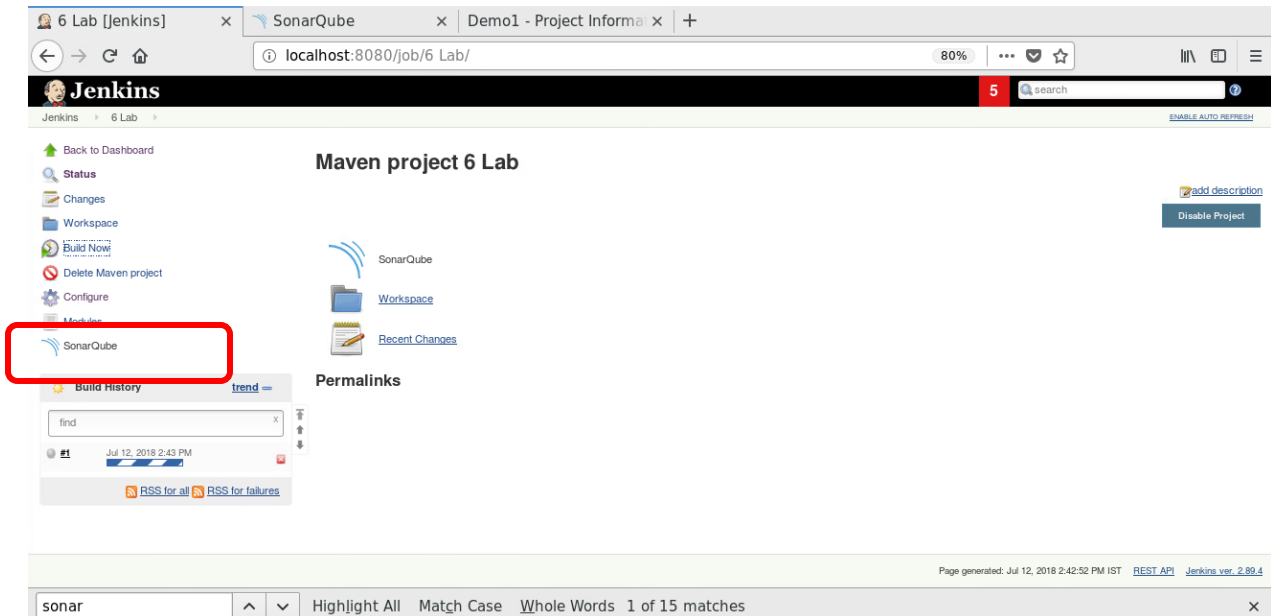
- Define sonar properties

The screenshot shows the Jenkins configuration page for a job, specifically the 'Build Environment' tab. Under the 'Pre Steps' section, there is a configuration for 'Execute SonarQube Scanner'. The 'Task to run' is set to 'scan'. The 'JDK' is set to '(Inherit From Job)'. The 'Path to project properties' is empty. The 'Analysis properties' field contains the following text: `sonar.projectKey=my:Aviproject`, `sonar.projectName=MyQualityReview$BUILD_NUMBER`, `sonar.projectVersion=1.0`, `sonar.sources=src/main/java`, and `sonar.tests=src/test/java`. At the bottom left, there are 'Save' and 'Apply' buttons.

- Configure Build with maven goal (Example: install)

The screenshot shows the Jenkins configuration page for a job, specifically the 'Build' tab. Under the 'Build' section, the 'Root POM' is set to 'pom.xml' and the 'Goals and options' are set to 'clean install'. There is an 'Advanced...' button to the right. Under the 'Post Steps' section, there are three radio buttons: 'Run only if build succeeds', 'Run only if build succeeds or is unstable', and 'Run regardless of build result'. The third option is selected. Below the radio buttons, there is a text box that says 'Add post-build step'. Under the 'Build Settings' section, there is a checkbox for 'E-mail Notification'. At the bottom left, there are 'Save' and 'Apply' buttons.

- Initiate Build process by clicking “Build Now” and observe Jenkins console responses



- Refresh SonarQube and walkthrough project Analyzed for more details

