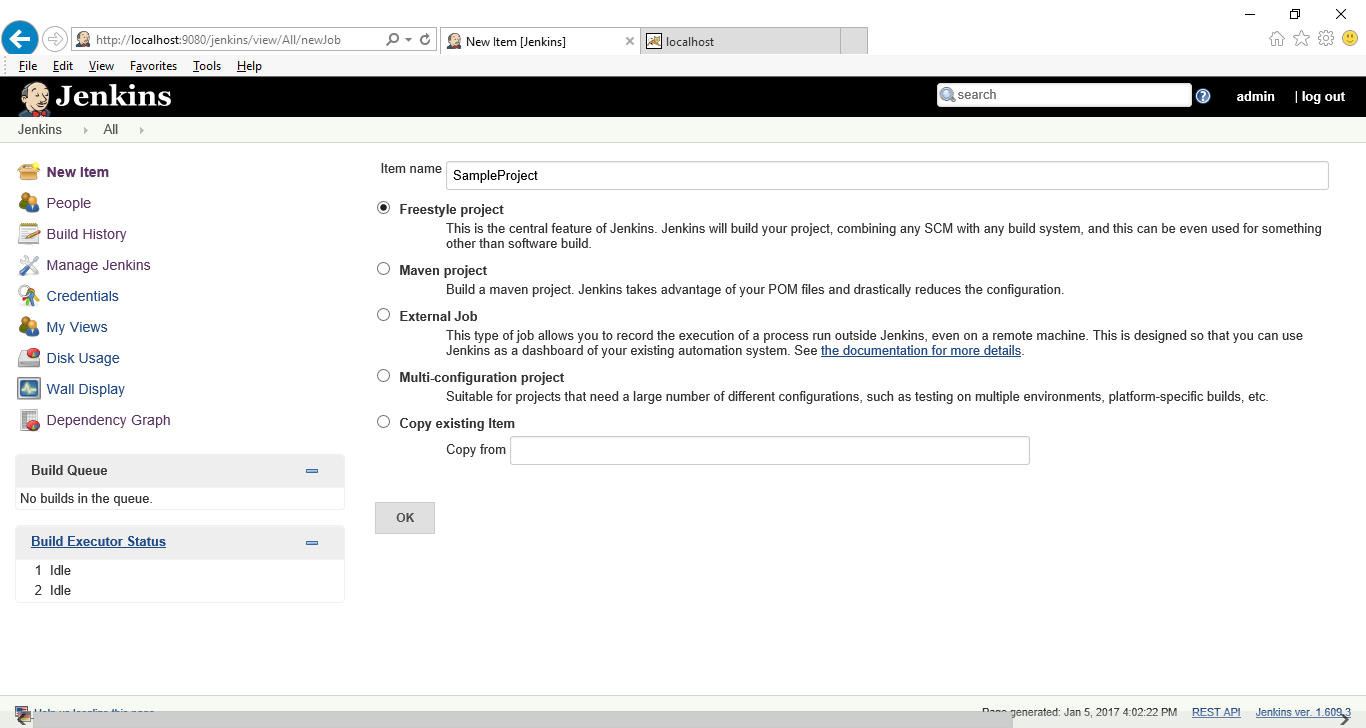
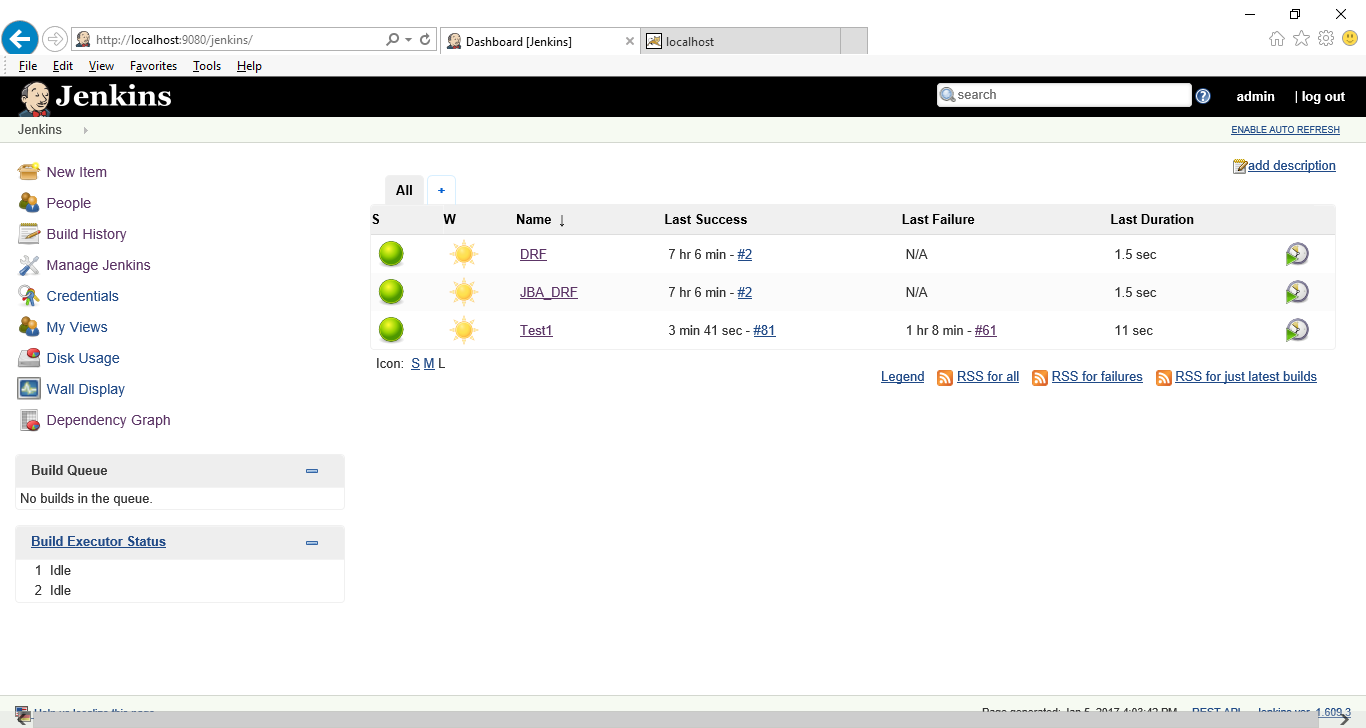
* After installing jba dw and jba ci, execute c:/jba/set-jba-dw-environment.bat and c:/jba\_ci/set-jba-ci-environment.bat.

**Jenkins:**

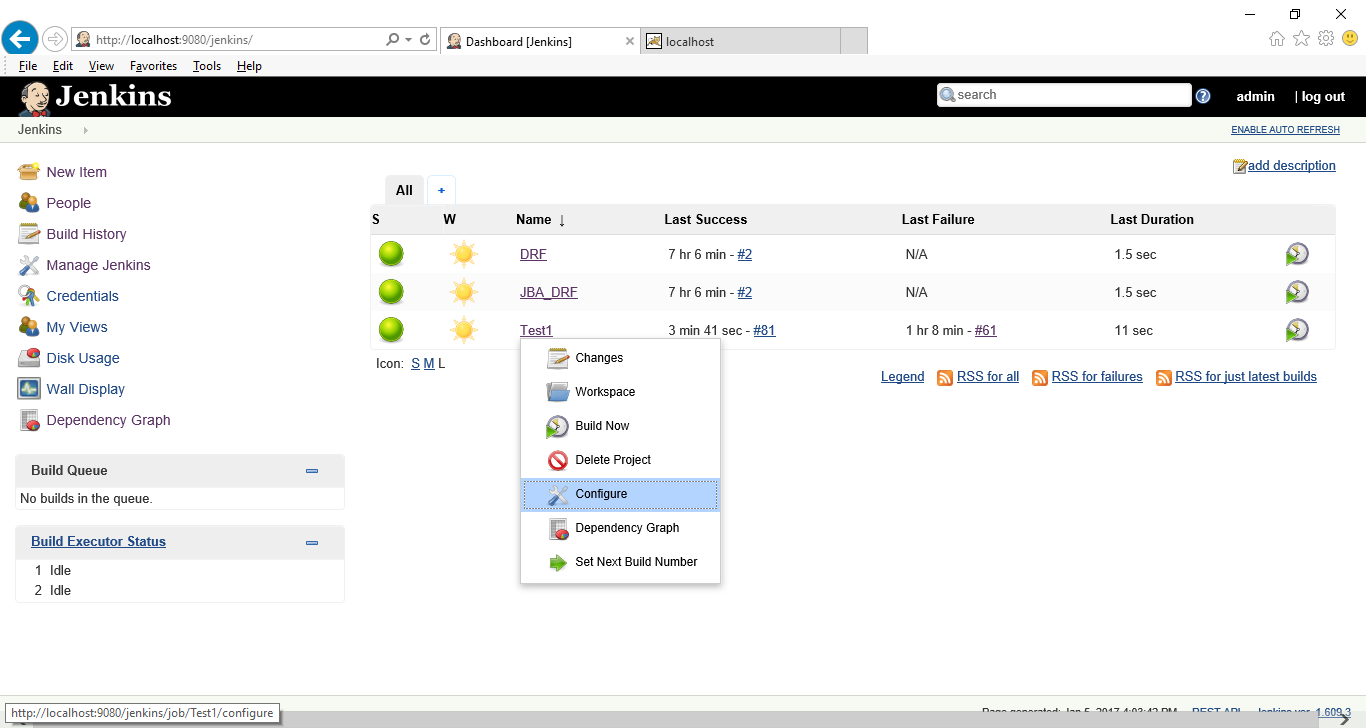
* Start Jenkins by executing below command

C:\jba\_ci>"C:\Program Files\Java\jre1.8.0\_111\bin\java" -Xmx512m -Duser.language=en -DhttpProxyHost=myproxy.accenture.com -DHttpProxyPort=8080 -jar C:\jba\_ci\jenkins-1.609.3\jenkins.war --httpPort=9080 --prefix=/jenkins --ajp13Port=-1

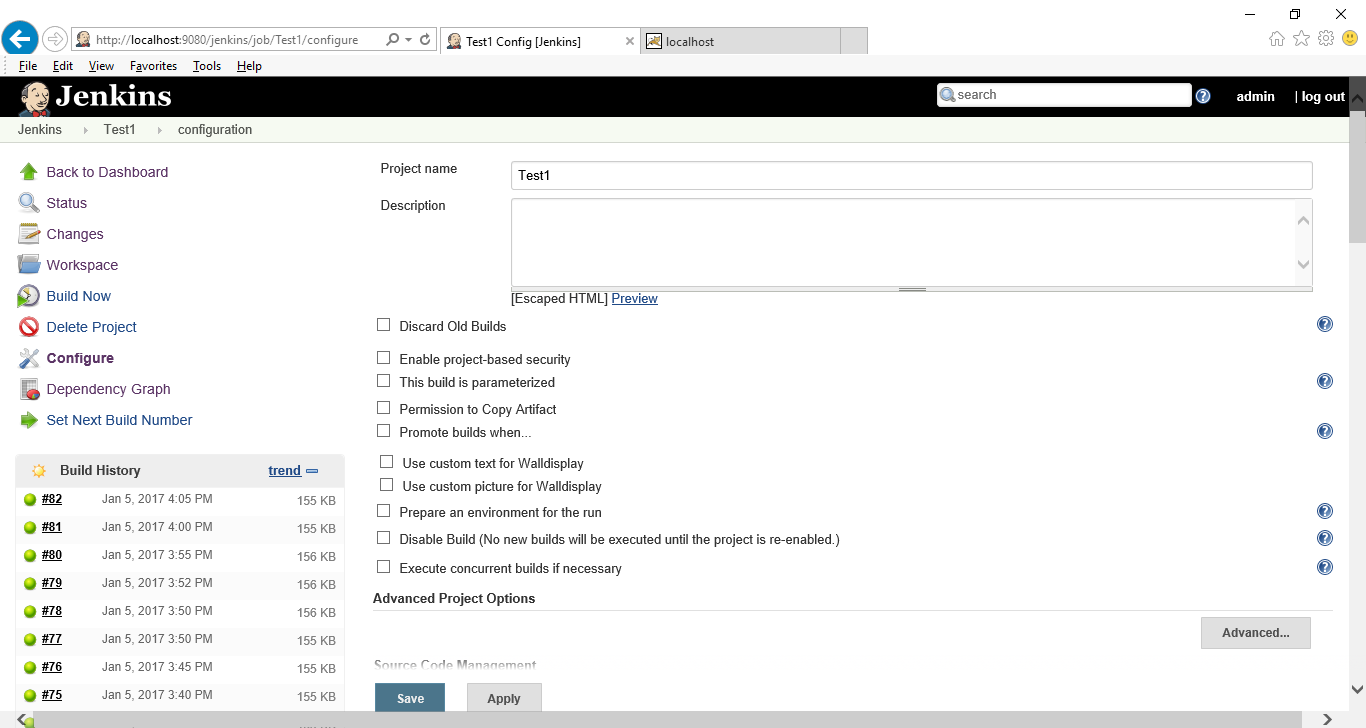
* Click on New Item 🡪Enter item name and select any one option from the listed projects -->Click on OK.
* 
* Created item/job(s) appears in the home page.



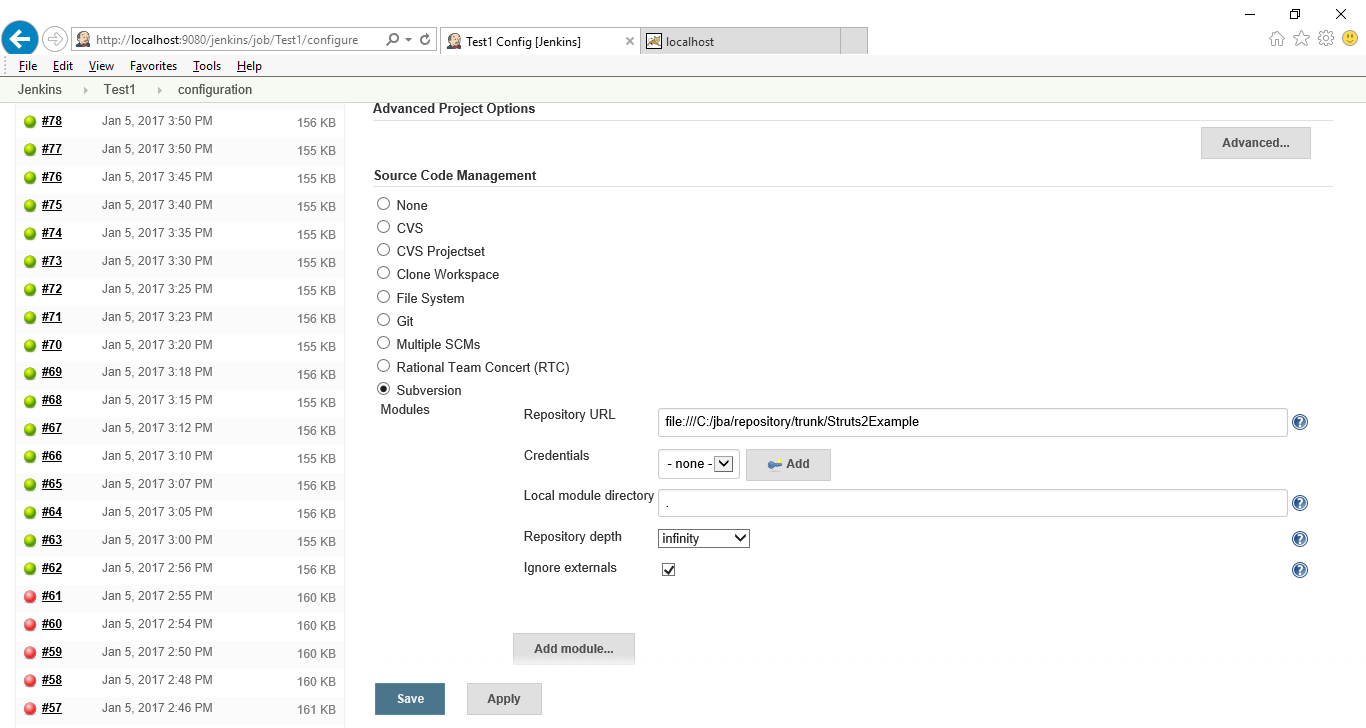
* Select ‘Configure’ option from the item menu to configure the properties.



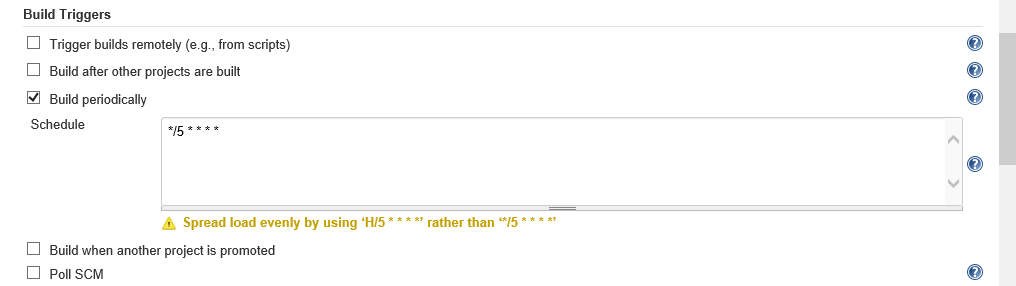
* Build related properties can be configured in configuration page.



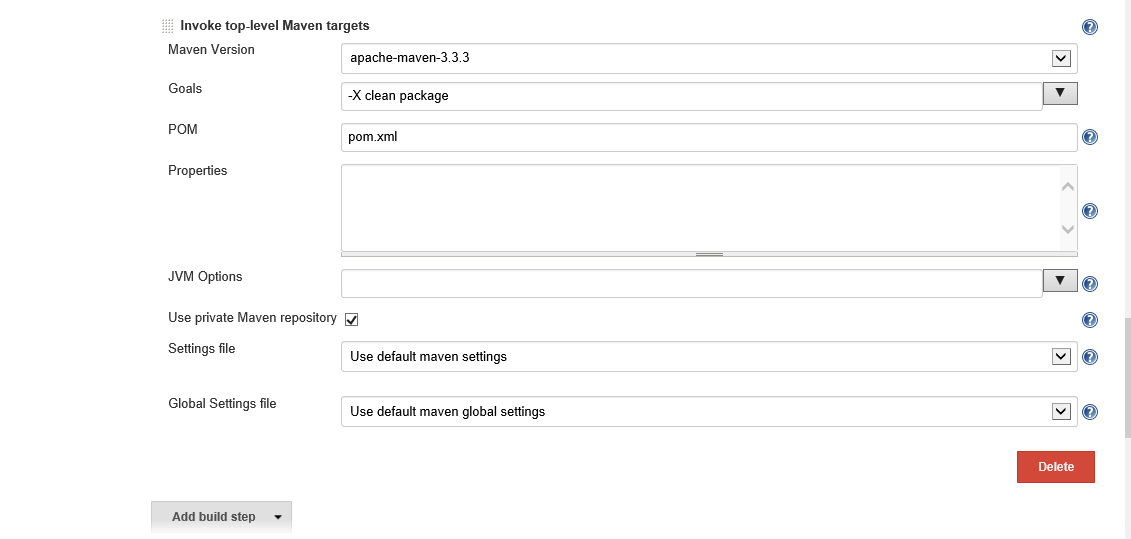
* Select Subversion to configure repository in ‘Source Code Management’ section.Enter details required to access repository as shown below



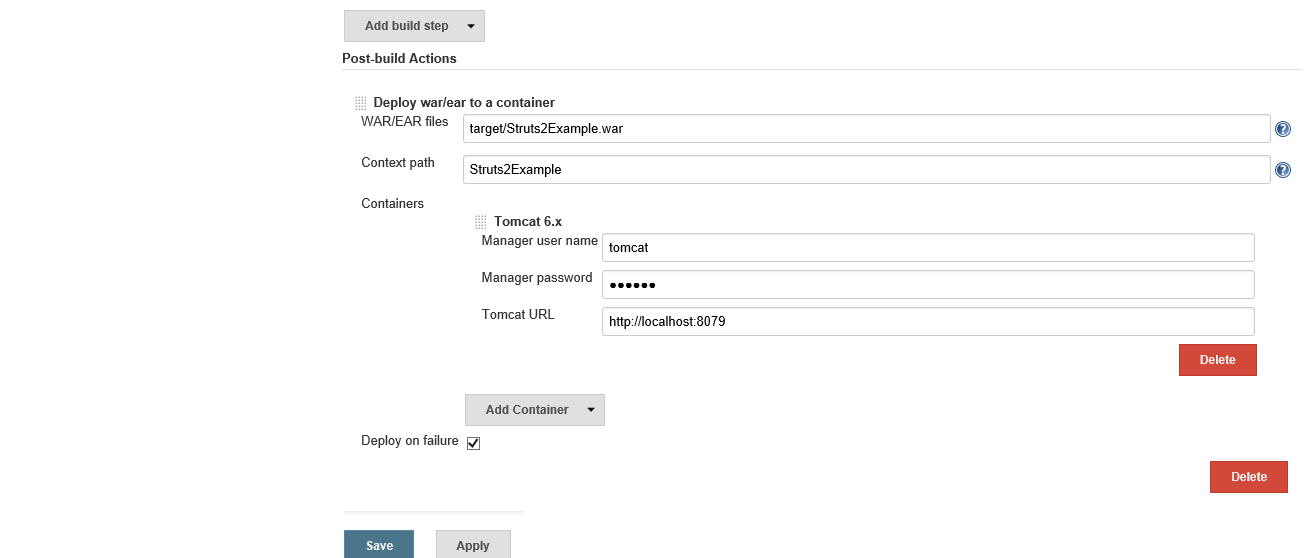
* Use ‘Build Triggers’ section to define time interval for builds. ‘\*/5 \* \* \* \*’ indicates



* Use ‘Build’ section to automate the build process by selecting build automation tools from ‘Add build step’ drop down.



* Use ‘Post Build Sections’ for post build processes like deployment to container. Enter server details as shown below.



**Note:** Manager username and Manager password are from tomcat\_users.xml with role ‘manager’.

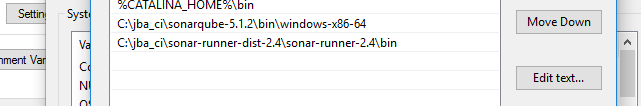
* Click on Save.
* We can manually build the job by selecting ‘Build now’ option or it will be automatically triggered as per the time interval given in ‘Build triggers’.

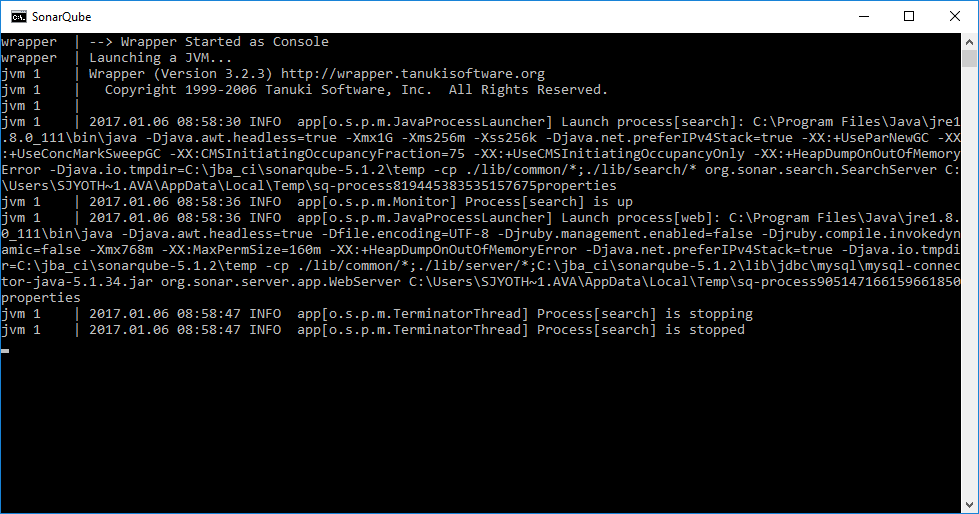
**SonarQube:**

* It is for continuous integration and improving code quality.
* It is an open source platform for Continuous Inspection of code quality. It is written in java and supported for 25+ languages such as Java, C/C++, C#, PHP, Flex, Groovy, JavaScript, Python, PL/SQL, COBOL, etc, it is also used for Android Development.
* It helps for various tasks and provide reports on duplicated code, coding standards, unit tests, code coverage, complex code, potential bugs, comments and design and architecture.

**Setup:**

* Copy and unzip sonar.zip and sonar-runner.zip
* Edit Environment variables and set it to ‘path’ variable for sonarqube abd sonar runner as shown below





* Edit C:\jba\_ci\sonarqube-5.1.2\conf\sonar.properties to configure db settings.

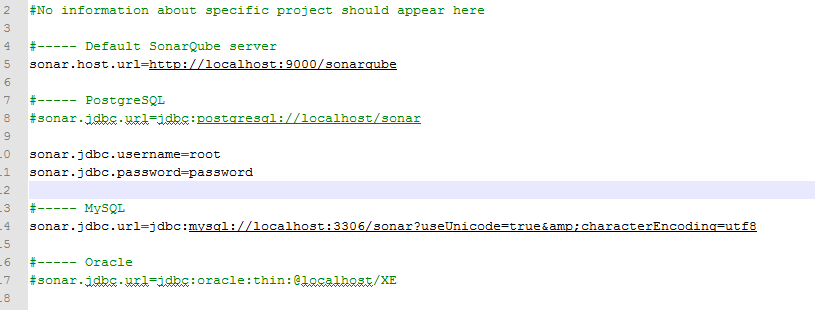
Ex: changed sonar.jdbc.username=sonar, sonar.jdbc.password=sonarsonar to sonar.jdbc.username=root

sonar.jdbc.password=password (which are m=MySql username,password).

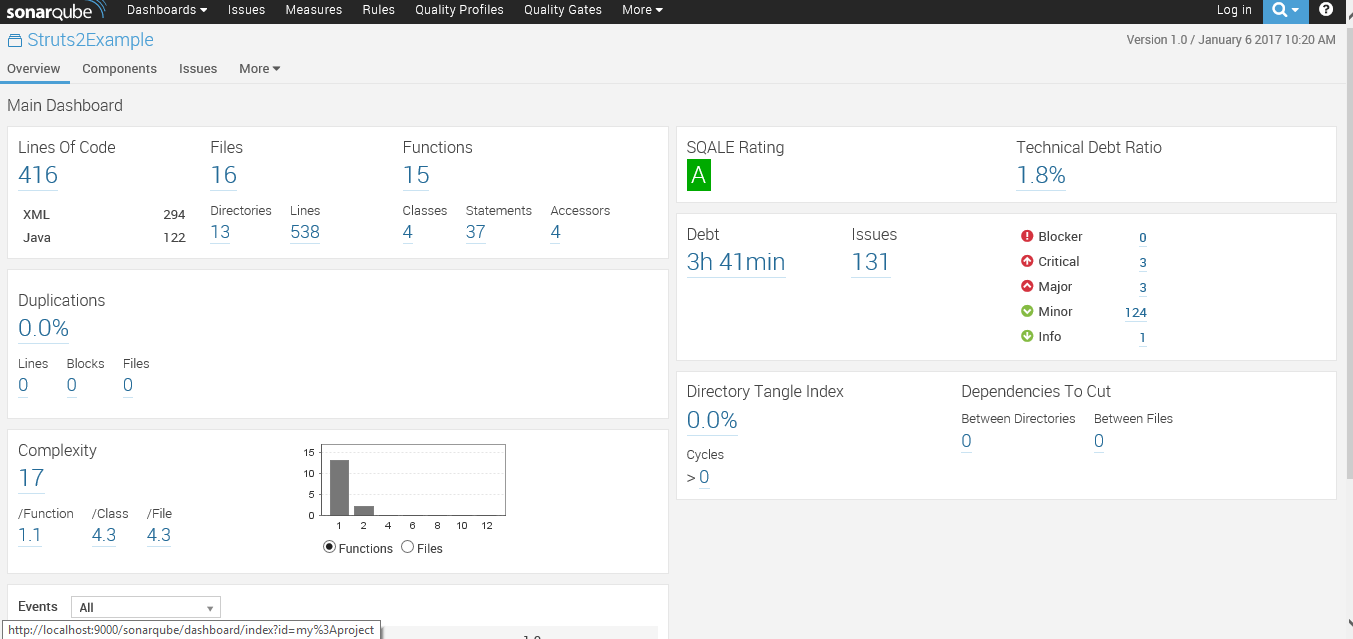
* Create schema with name ‘sonar’ in Mysql.
* Start the Sonar server using the script available in SONAR\_HOME\bin\windows-x86-32\StartSonar.bat (OS dependent)
* After starting the server you can browse to <http://localhost:9000/sonarqube>.

Sonar Runner:

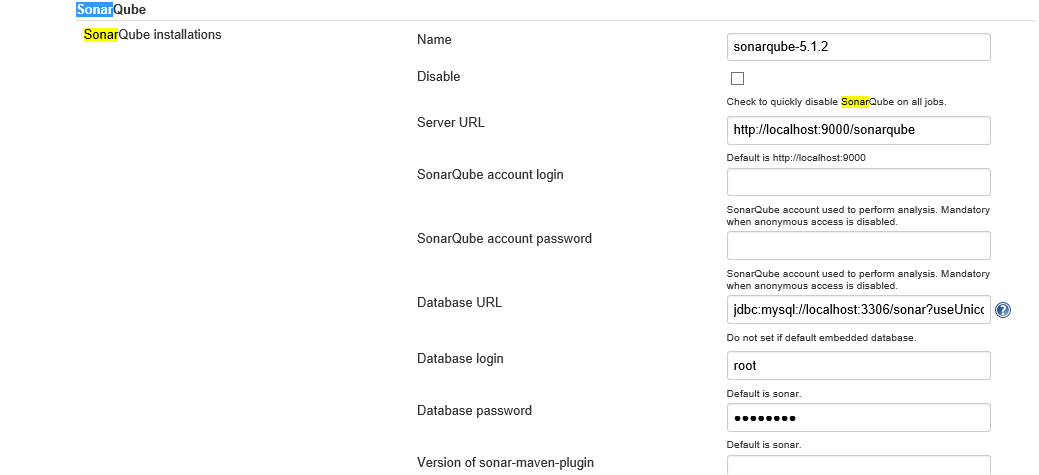
Also edit sonar-runner.properties to configure db settings.



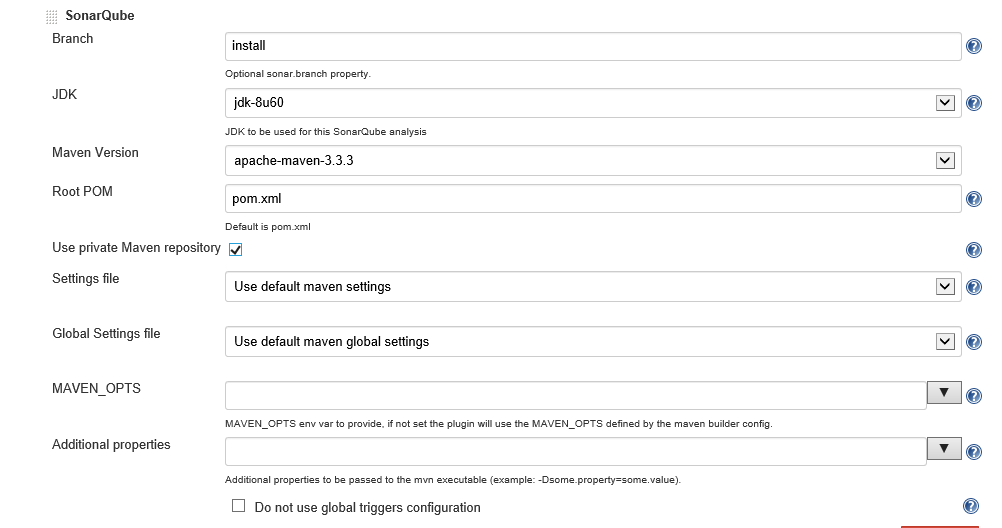
* Create a configuration file in the root directory of the project, namely, sonar-project.properties
* You may choose to copy and paste the code given on this page, [Analyzing with SonarQube Runner](http://docs.codehaus.org/display/SONAR/Analyzing+with+SonarQube+Runner), in sonar-project.properties file. For setup purpose, just change the value of sonar.projectName to the project name you created.
* Open a command prompt, and go to project root folder.
* Execute “sonar-runner” command to run the analysis. You would see the analysis run.
* When ‘sonar-runner’ is executed, it creates tables in the schema named(sonar-which needs to be created by user)
* Goto the browser and access the page, <http://localhost:9000/sonarqube>
* You would find your project listed under “PROJECTS”. Click on your project listing and you would land up on the project dashboard.



* In Jenkins select ‘SonarQube’ form ‘Post build Actions’ drop down and fill the required details
* Modify sonar properties in Managejenkins🡪configure system🡪SonarQube section🡪modify db settings



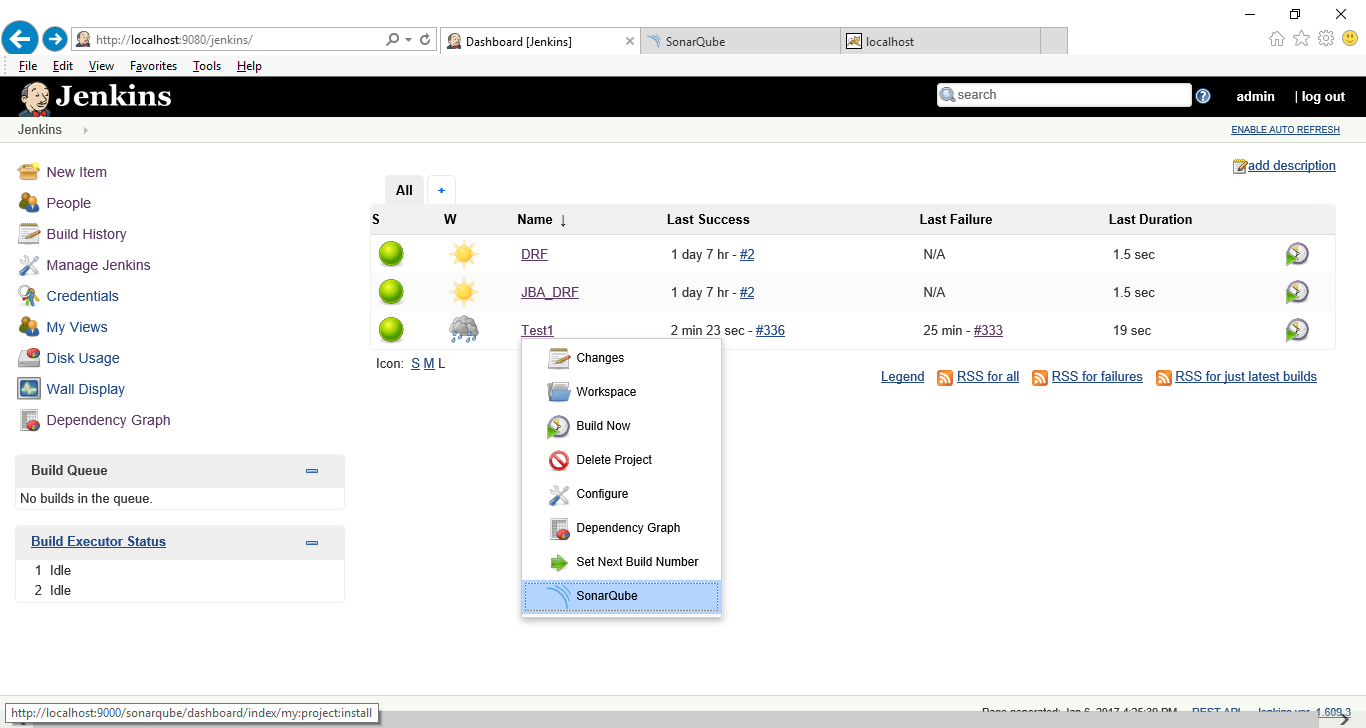
Sonarqube configuration for the item/job:



**Note:**

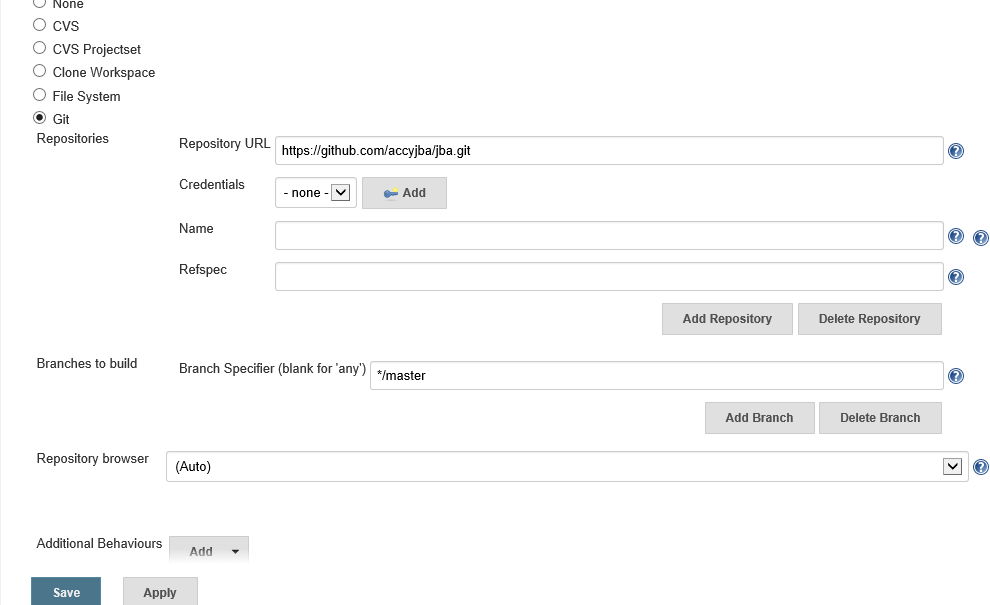
**Solarqube must be up and running**

* After successful build we can ‘Sonar Build’ option in the item menu.



**Git:**

**Git configuration in Jenkins Job/Item:**



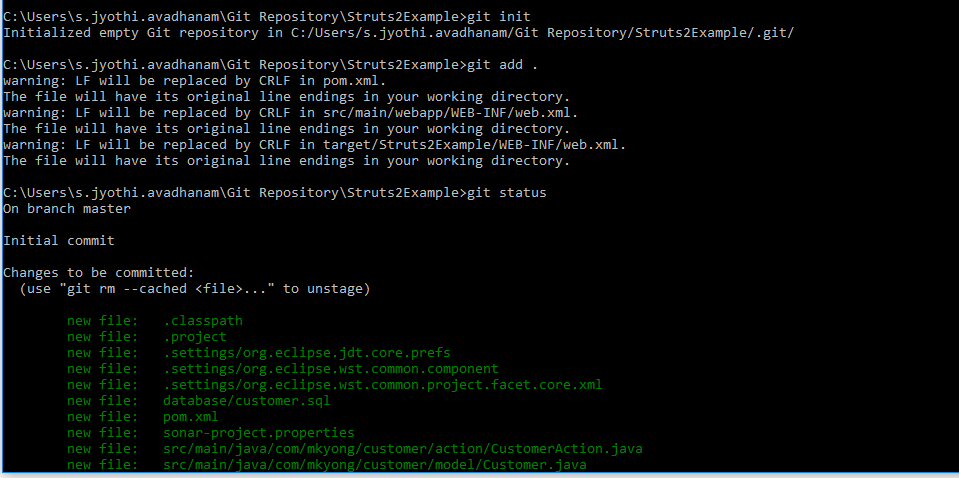
**Note:** set the git/bin path to environment variables.

Go to project location which needs to be added to repository.

$ git init🡪Executing git init creates a .git subdirectory in the project root, which contains all of the necessary metadata for the repo.

$ git add .--> use ‘add’ to add files to git repository. ‘.’ Indicates all files.

$ git status🡪 to see the current status of the project.



$git commit -m ‘First commit’🡪commits all the uncommitted changes to repository.

* Login to <Https://github.com> and create repository.
* Run the below commands in command prompt.

**Git Commands:**

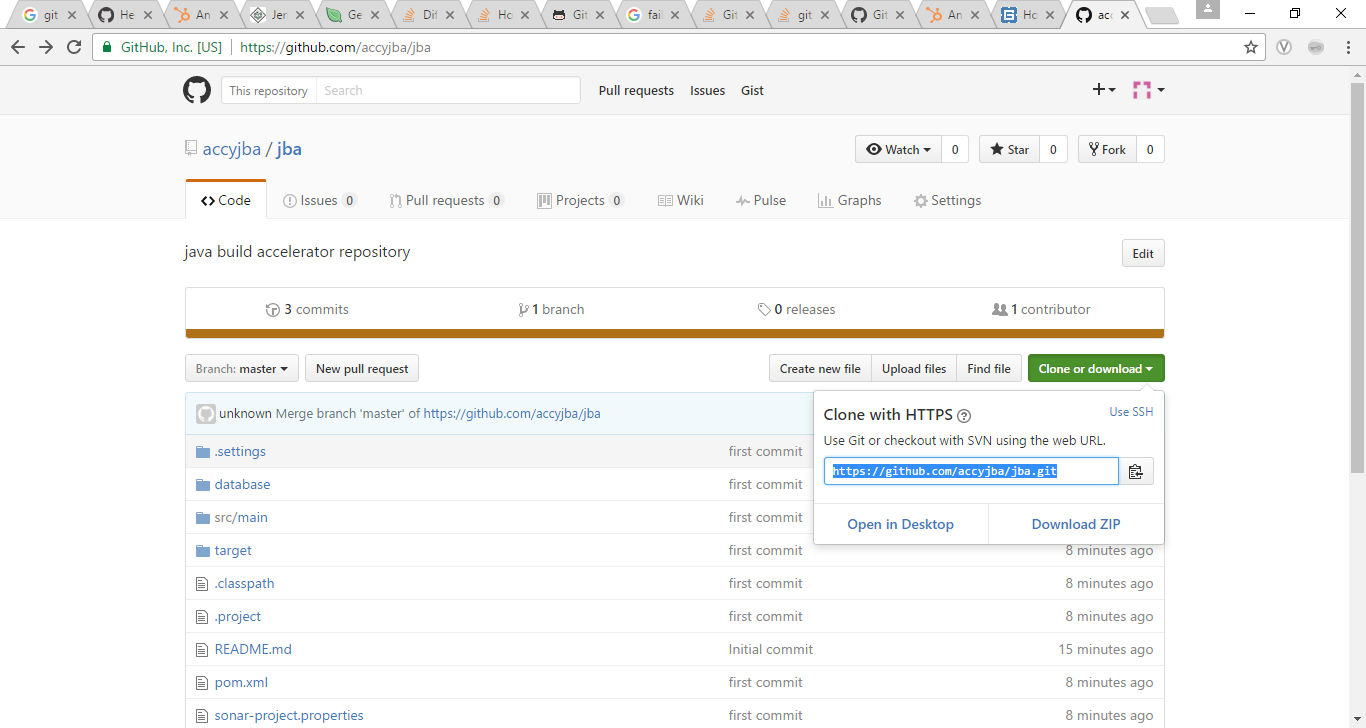
1. git init
2. git add ProjectFolderName
3. git commit -m "first commit"
4. git remote add origin https://github.com/accyjba/jba.git
5. git pull origin master (only for the first time files uploading to Git)
6. git push -u origin master
7. git config --global http.sslVerify false (to disable certificates which is not recommended)

**Note :For further commits:**

1. git add filename
2. git commit -m filename
3. git push origin master

**Git with Jenkins:**

Take repository URL from Git hub.



**Build Triggers:**

Check Poll SCM option to have the build triggered immediately once the code is committed

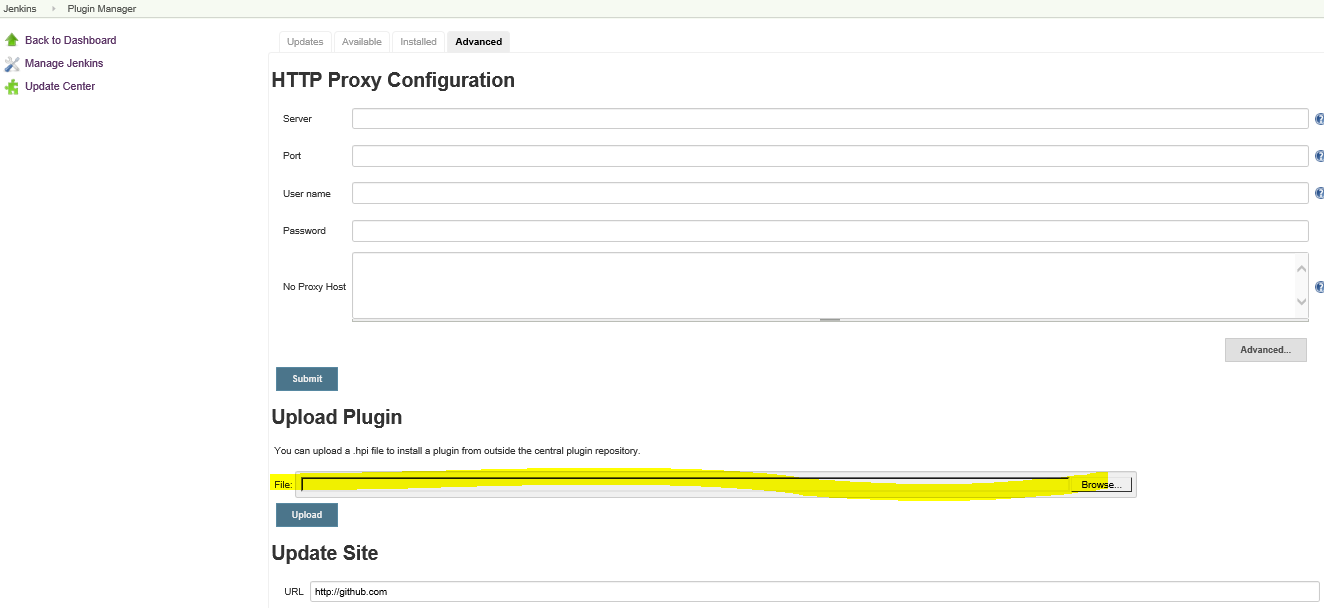
 For polling every 5 minutes, \*/5 \* \* \* \*

**Quality Gates:**

you define a set of Boolean conditions based on measure thresholds against which projects are measured. For example:

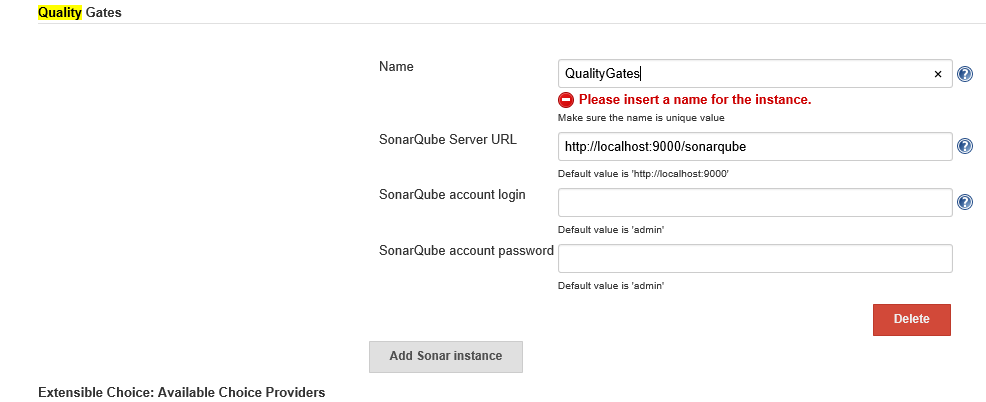
* No new blocker issues
* Code coverage on new code greater than 80%
* Etc.
* Use Jenkins [Quality Gates Plugin](https://wiki.jenkins-ci.org/display/JENKINS/Quality+Gates+Plugin), which fails the build if the predefined sonar quality gates are not green.

Note: If plugins are not found in ‘Manage Plugins’, download plugins (.hpi) from <https://wiki.jenkins-ci.org/display/JENKINS/> and upload it in Manage Jenkins🡪Manage Plugins🡪Advanced

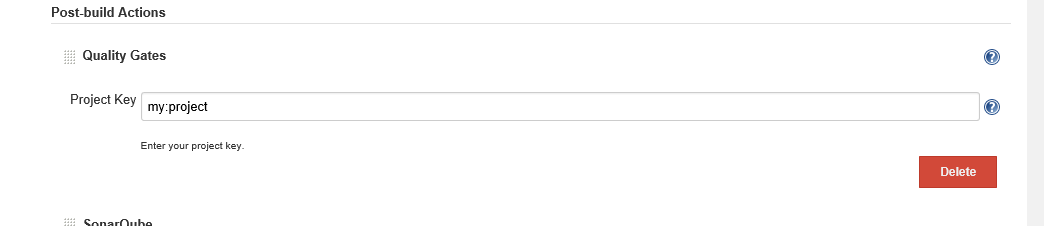


This will install Quality Gates plugin into Jenkins

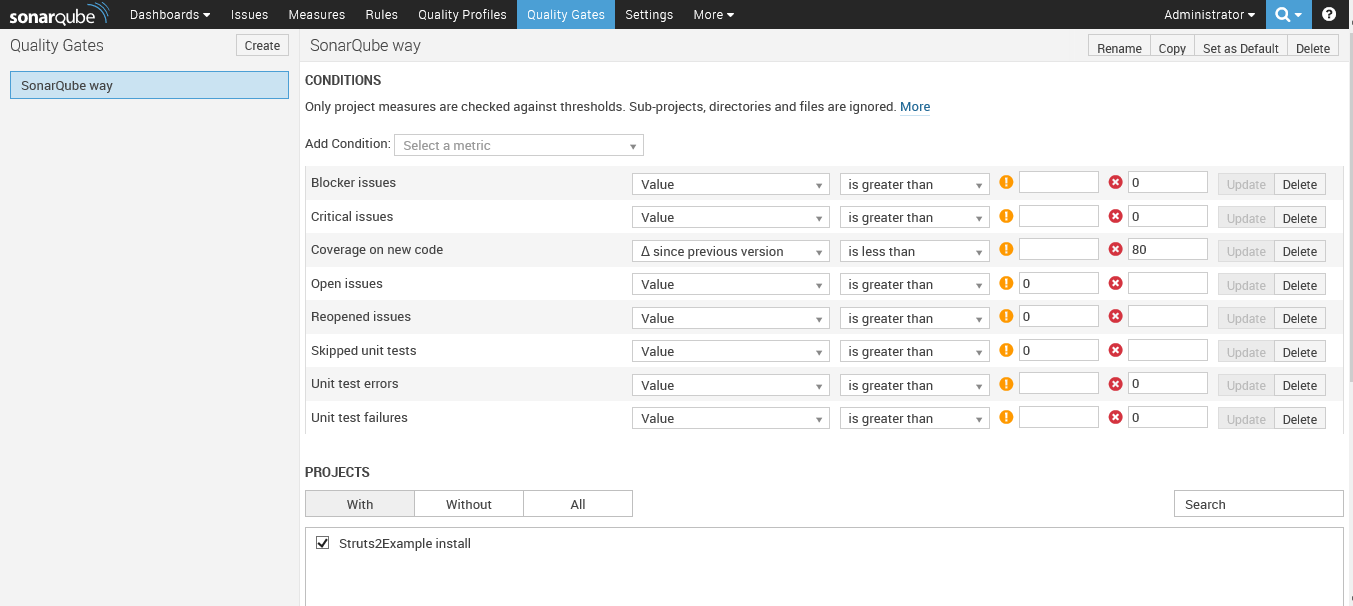
* Go to Manage Jenkins🡪Configure System🡪Quality Gates



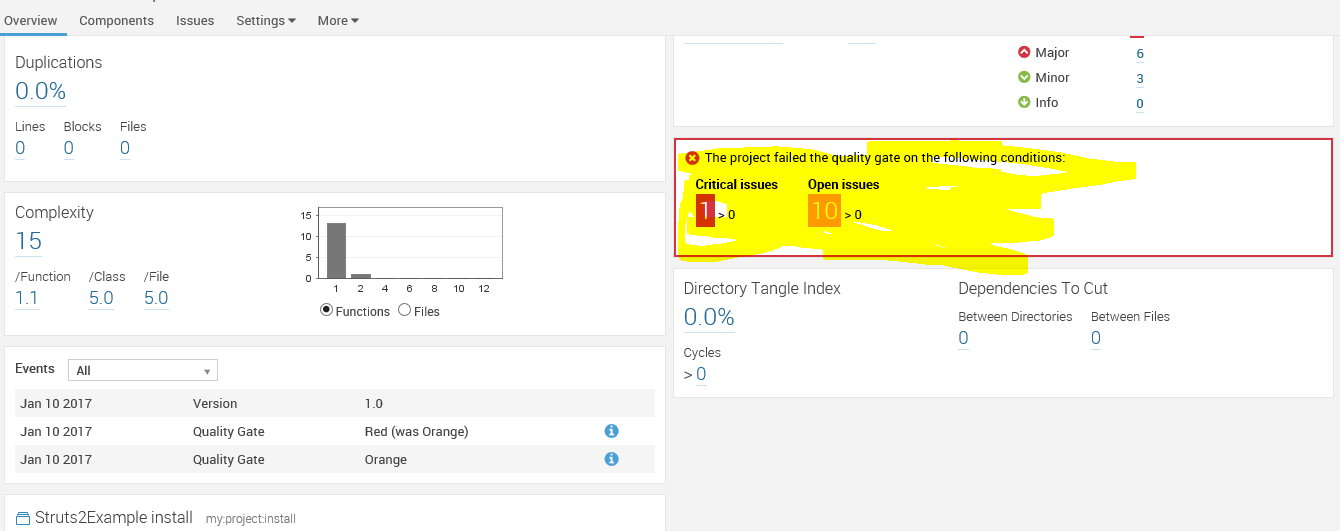
🡪Select Item/Job🡪Configure🡪select Quality Gates from ‘Add post-build’ Actions dropdown🡪Enter project key which was configured in sonar-project.properties

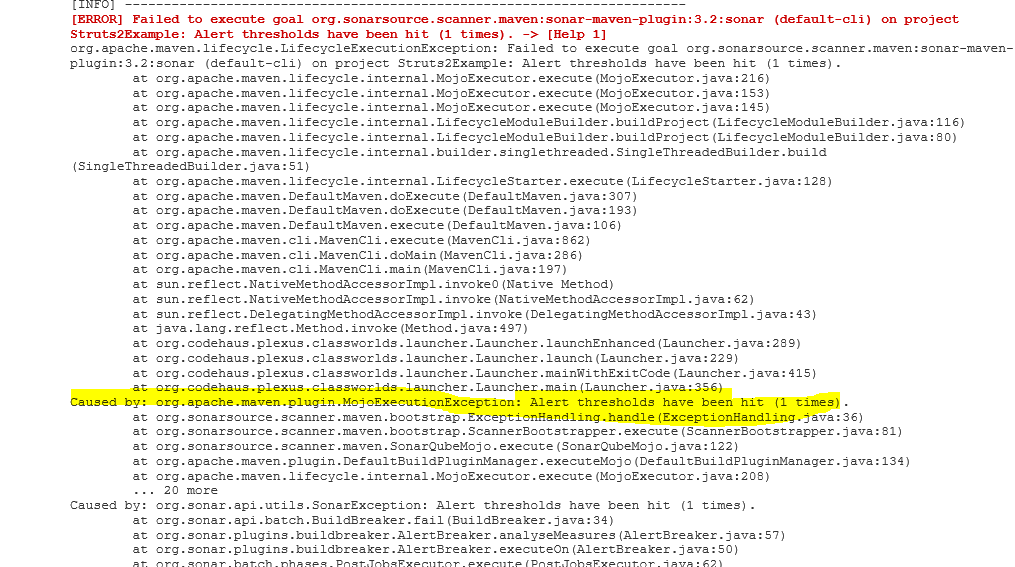


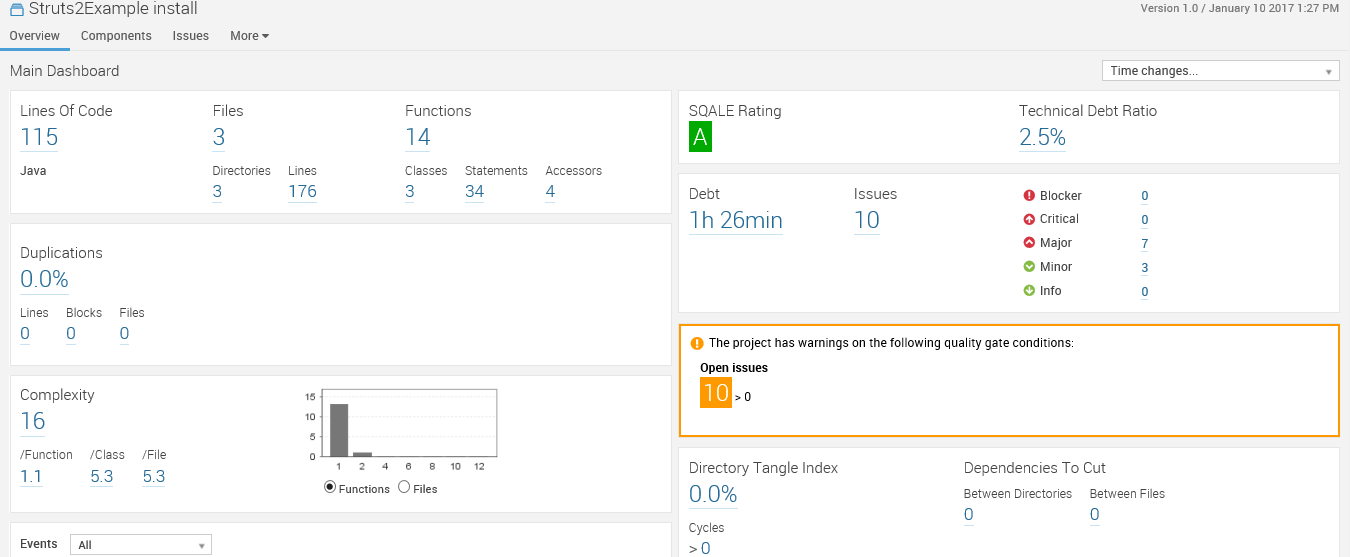
* Login as Administrator into SonarQube and select QualityGates.
* A quality gate is a set of conditions and a set of projects to be checked against these conditions.
* Conditions can be set on measures (i.e. No blocker issues) or on deltas (i.e. No new blocker issues since previous version). Two thresholds can be set for each condition: warning and error.
* In the below screen Critical Issues is been modified with ‘Value’,’is greater than’ , blank for warnings and 0 for errors and click on update.



* Build the project in Jenkins. If critical issues are greater than 0, build fails.
* In Sonar Qube home page we can see the quality gate status as shown below.

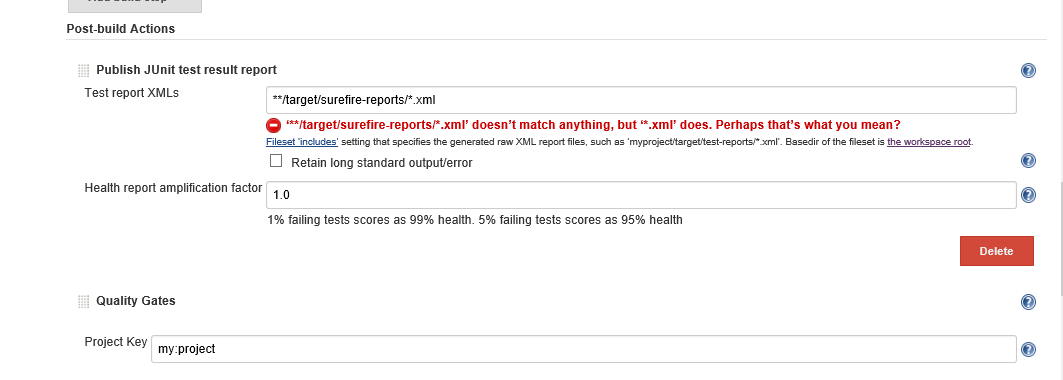


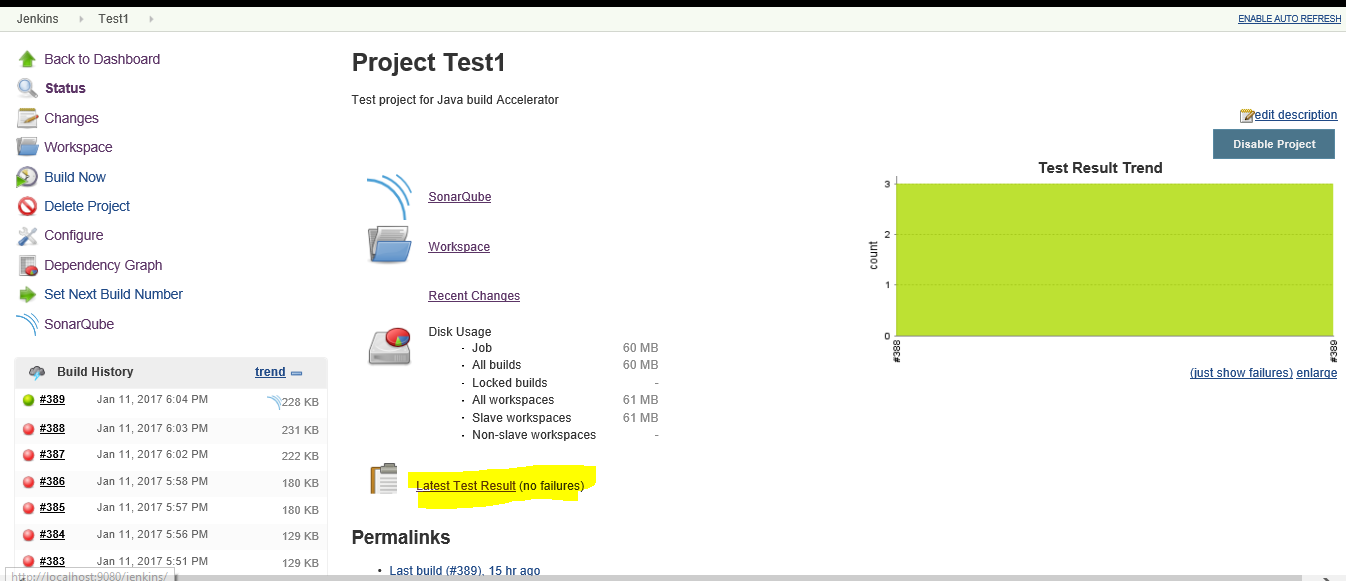
* In job console output we can see the error which causing build fail.
* 
* After fixing the critical issue, build will be successful.

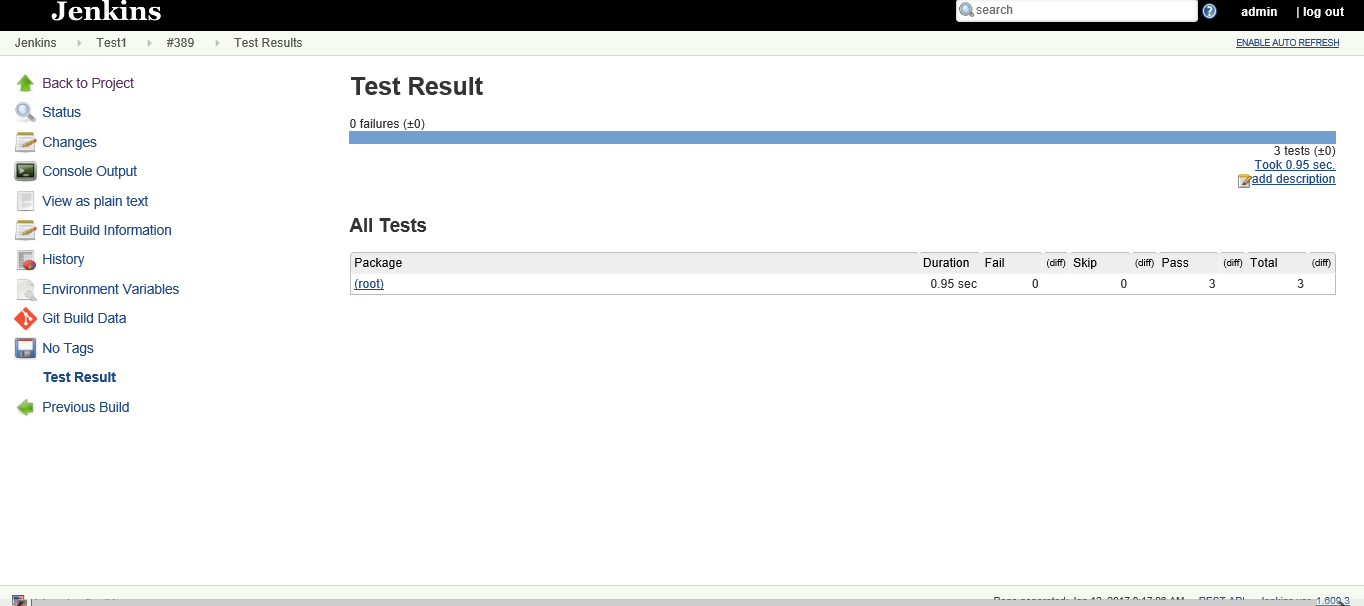


**Junit:**

* Create test cases in src/test/java file.
* Include junit plugin in pom.xml (4.1 is preferable version).
* Junit in Jenkins configuration as follows:
  + Select ‘Publish Junit test result report’ from Add post-build actions.
  + For Test report XMLs give ‘\*\*/target/surefire-reports/\*.xml’



* Build the job. We can see the test result by clicking on the job as shown below.
* 
* By clicking on Latest Test Result, test results will be shown as below



**Selenium:**

* Libraries required to run selenium web driver test scripts are available in JBA CI tools
* Also download IEDriverServer.exe 32bit and add it to project.
* Add IEDriverServer.exe to the system property as shown below



* Create java class file and write the script to execute the test and run the test.