SONARQUBE INTEGRATION WITH JENKINS

As soon as developer commits the code to GitHub, Jenkins will fetch/pull the code from repository and will perform static code analysis with help of Sonar Scanner and send analysis report to SonarQube Server then it will automatically build the project code.

*Pre-requisite*

Jenkins is setup with GitHub with some build trigger (in my case its Poll SCM)

Steps to setup **SonarQube**in windows machine.

**Step 1.** Download Community edition from ***h***[***ttps://www.sonarqube.org/downloads/***](https://www.sonarqube.org/downloads/)

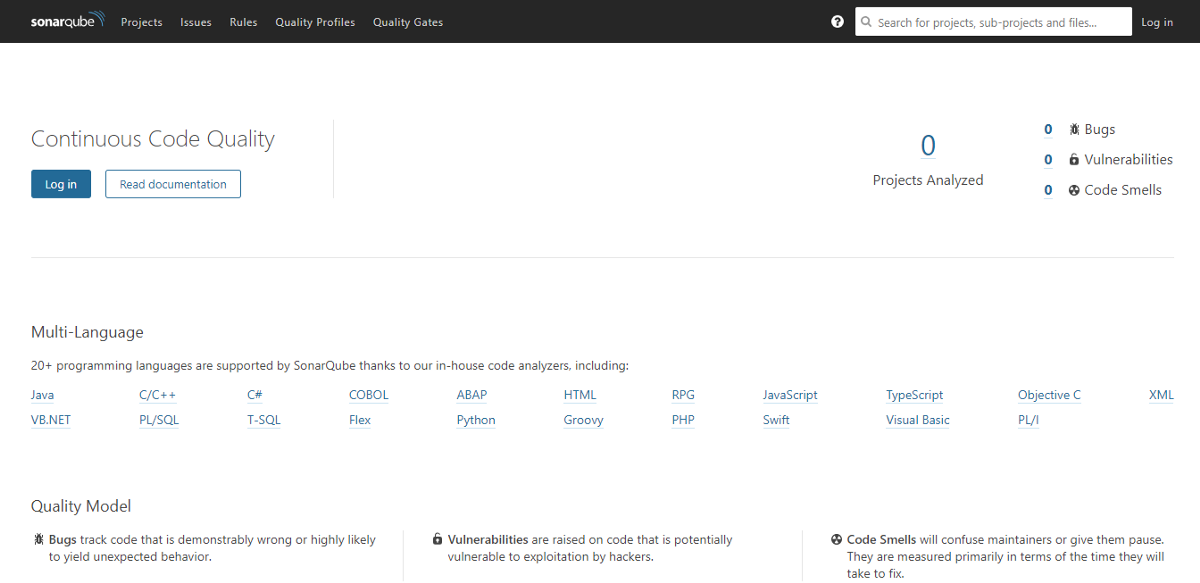
**Step 2**. Extract .zip file and navigate to *bin* folder

**Step 3.** Based on your machine configuration go to folder for *32-bit OS*move to ***windows-x86–32*** and for 64-bit OS move to ***windows-x86–64***

**Step 4.** Run ***StartSonar.bat*** and after few minute it will start your SonarQube server.

**Step 5.** Open browser and**http://localhost:9000/** (9000 is default) you will be navigated to below window

**Note:** For any configuration changes go to ***conf*** folder and ***sonar.properties***file. Here you can configure *database*, *ldap*, *webserver*, *sso authentication, logging, etc…*, e.g. for port — under webserver section I have added ***sonar.web.port=9005***



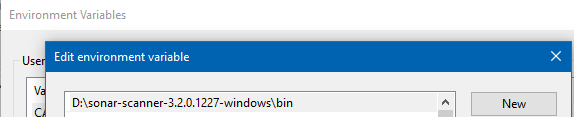
**Step 6.** You can login using default user and password (***admin : admin***)

***Sonar Scanner :*** If you want to scan any project standalone just follow the steps

**Step 1.** Open and download *Sonar Scanner* from below link [***https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner***](https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner)

**Step 2.** Unzip the file (in my case its in D drive so *D:\sonar-scanner-3.2.0.1227-windows*) and navigate to its ***bin*** folder and add ***Path*** to ***environment variables*** of system and **Save**

***Path — D:\sonar-scanner-3.2.0.1227-windows\bin***



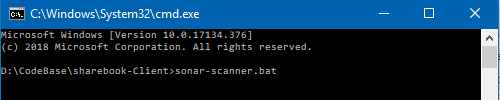
**Step 3 :** Navigate to ***config*** folder of sonar scanner (*D:\sonar-scanner-3.2.0.1227-windows\conf*) here you will get a ***sonar-scanner.properties*** file. To keep it simple I have used only three properties(as below), ***sonar.host.url***property tells the url of *Sonar server* to post analysis report after scanning , ***sonar.projectKey***property will create a project inside your *SonarQube server*with the same name if project don’t exist else it will append analysis to it, ***sonar.sources***property defines that which folder to scan. You can provide either relative path or actual path to the folder you want to scan.

Edit it and add below lines

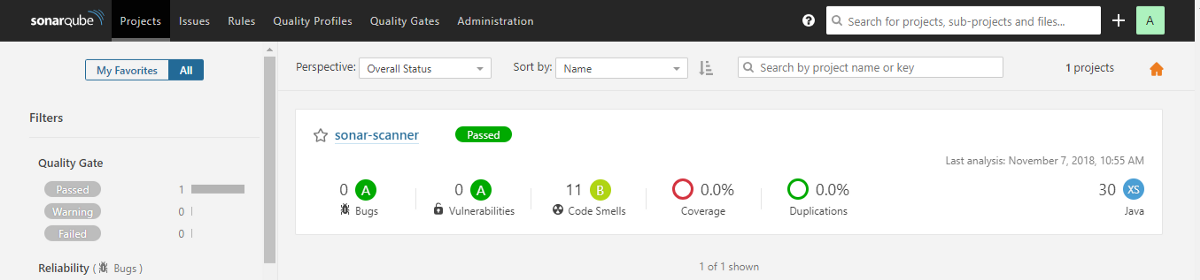
***sonar.host.url=http://localhost:9000  
sonar.projectKey=sonar-scanner  
sonar.sources=D:/CodeBase/sharebook-Client/src***

**Note :** This property file will be ***project specific***. It contains certain sonar properties like which folder to scan, which folder to exclude in scanning, what is the project key and many more you can see it from [***https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner***](https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner)

**Step 4.** Now navigate to the Project folder of you code you want to scan (**D:\CodeBase\sharebook-Client**) open command prompt on this location and run command ***sonar-scanner.bat*** as in below image.



**Step 5.** After successful analysis you will see it has posted the result to your *SonarQube server* [*http://locathost:9000/*](http://locathost:9000/)with new project inside it you have mentioned it in*sonar.projetKey*



**How it works :** When we run command *sonar-scanner.bat* from Project location it will find the *Path* from*system environment variable*and then sonar scanner will read its *properties file* from *conf* folder (*sonar-scanner.properties*). It scans the*source path*you have provide and finally it *posts* the analysis report to *SonarQube server*url with the project key mentioned.

**Step 1.** Open SonarQube server- Go to Administration > click on Security > Users > Click on Tokens (image 1)> Generate token with some name > Copy the token (image 2), it will be used in Jenkins for Sonar authentication.

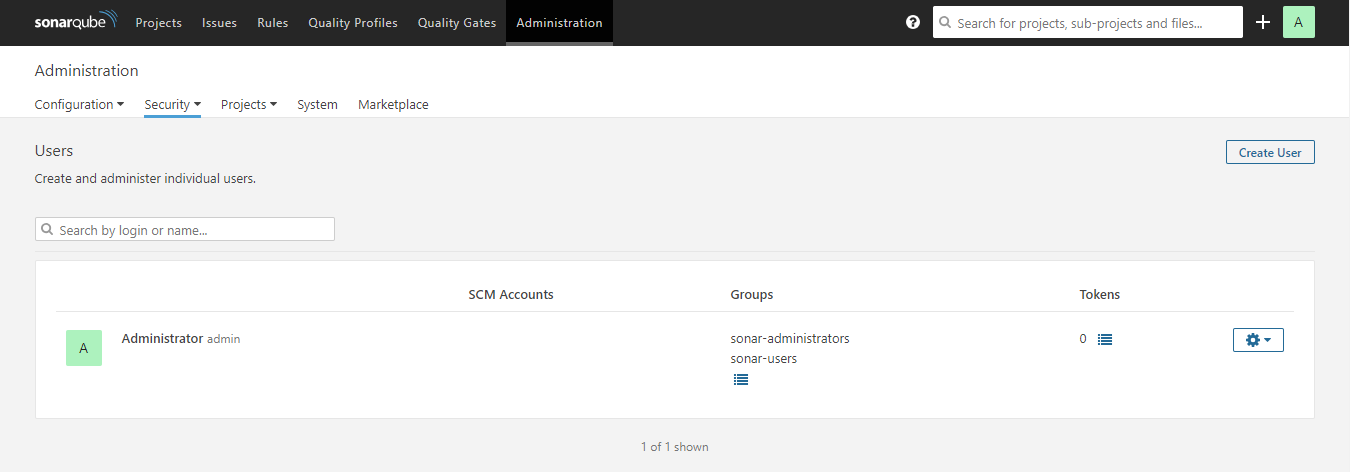


Image 1

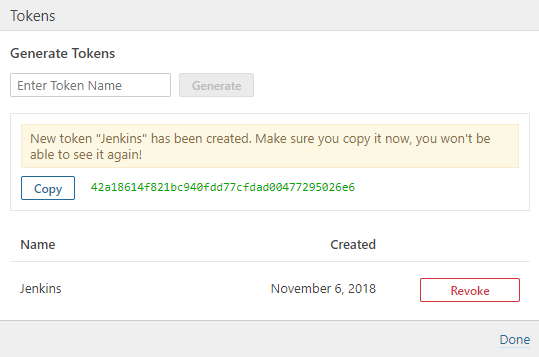
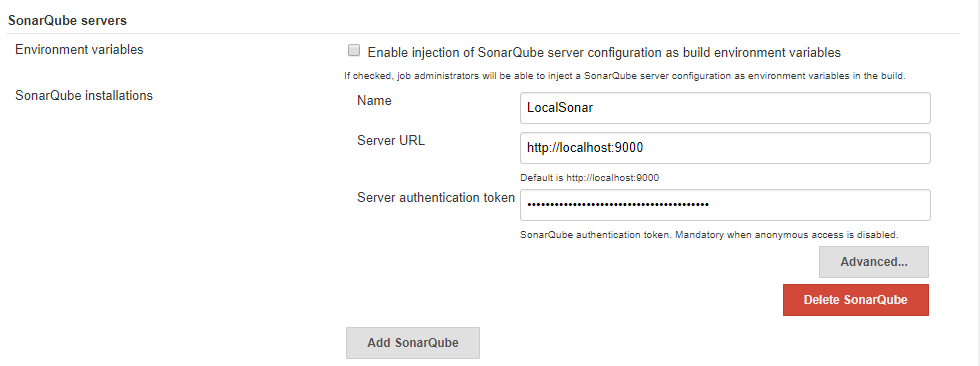


Image 2

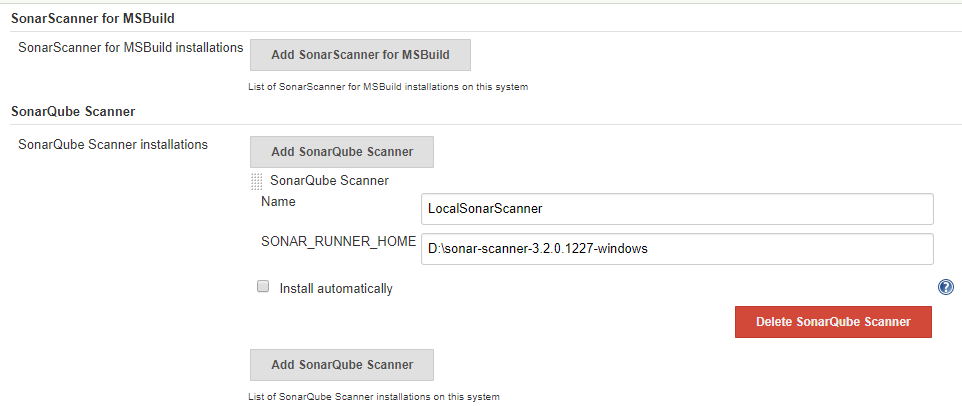
**Step 2.** Setup ***SonarQube*** with ***Jenkins***- Go to *Manage Jenkins* > *Configure system*> *SonarQube* server section > *Add SonarQube* > Name it, provide Server Url as ***http://<IP>:<port>*** > and authentication token copied from SonarQube Server > *Apply* and ***Save***



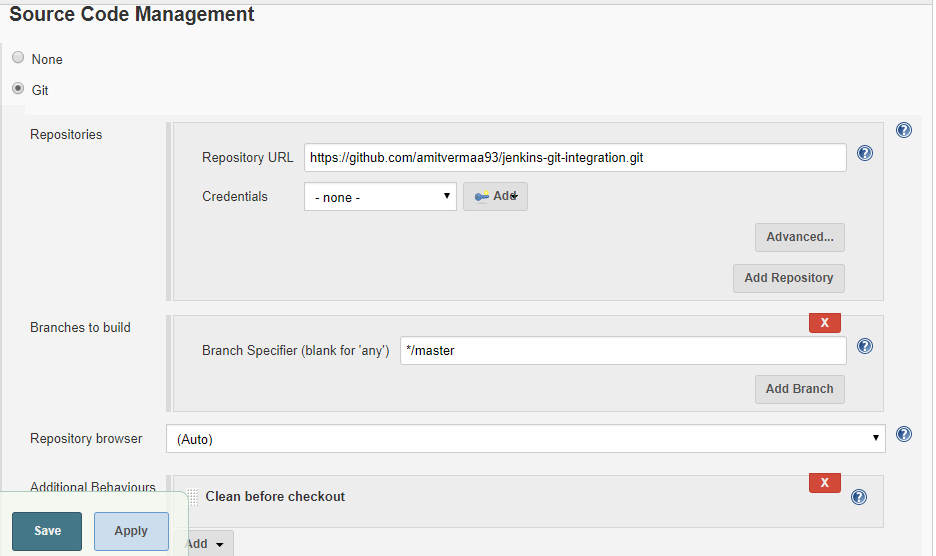
**Step 3**. Install ***SonarQube plugin*** to Jenkins. Go to *Manage Jenkins* > *Manage Plugins* > *Available*> Search for *SonarQube Scanner*> *Install.*



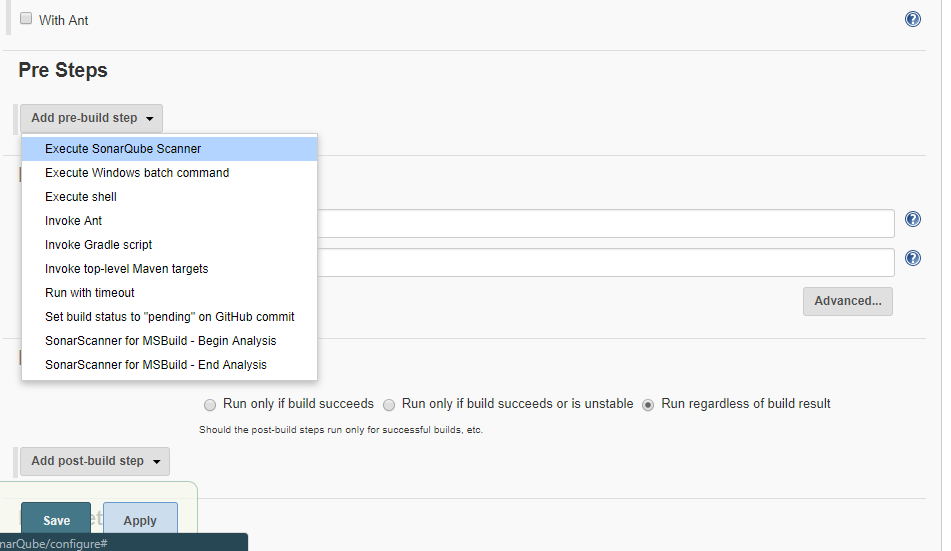
Configure ***Sonar Scanner*** in *Jenkins* : Go to *Mange Jenkins* > *Global Tool Configuration* > Scroll for *SonarQube Scanner* > *Add* sonar scanner > name it, uncheck if you already have sonar else it will automatically download for you and your sonar scanner setup will be done(in my case I already have) > provide path to***sonar runner home***as in below image



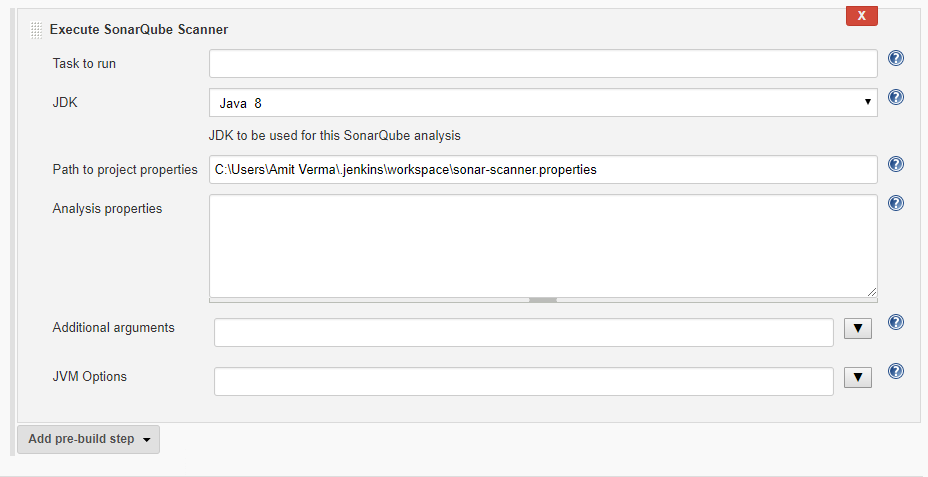
**Step 4.** Create a Job- *New Item* > Name and select a project type (in my case I am selecting *Maven* project you can opt for freestyle as well)



**Step 5.** Set *Git* under *SCM* section and use **\* \* \* \* \*** for *Poll SCM* under B*uild Trigger*section. Under *Build Environment* section add pre-buid step > select ***Execute SonarQube Scanner***



**Step 6.** Create a **.properties** file at any location and provide path on the task as below(I have created it in Jenkins workspace folder). This property file will be ***project specific***. It contains certain sonar properties like which folder to scan, which folder to exclude in scanning, what is the project key and many more you can see it from [***https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner***](https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner)



Inside sonar-scanner.properties write below code —

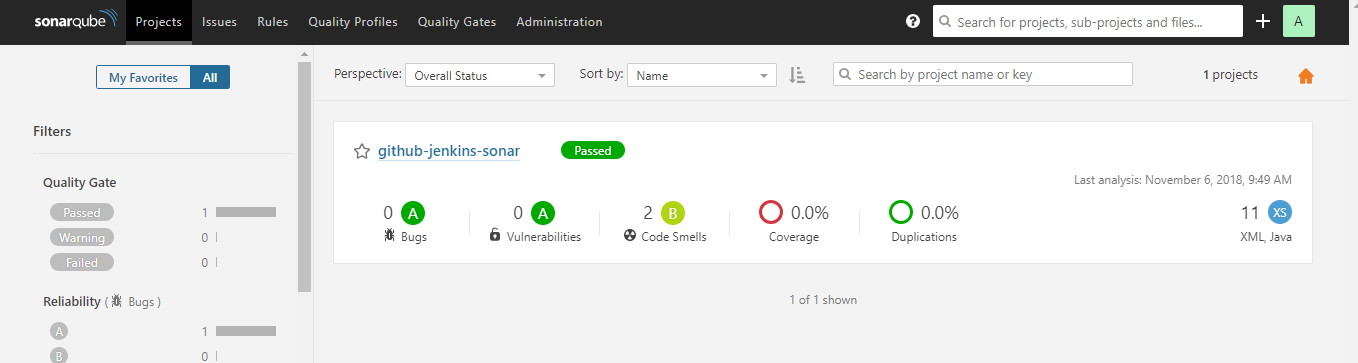
***sonar.projectKey=github-jenkins-sonar  
sonar.sources=./src***

To keep it simple I have used only two properties(as above), ***sonar.projectKey***property will create a project inside your *SonarQube server* with the same name if project don’t exist else it will append analysis to it, ***sonar.sources***defines that which folder to scan. You can provide either relative path from your Jenkins Job workspace or actual path to the folder you want to scan.

Since I have used ***./src****(use / for windows path )*thatmeans that I am currently on my Job workspace i.e. on ***C:\Users\Amit Verma\.jenkins\workspace\Jenkins-GitHub-SonarQube*** location and from here I am providing the path to the folder(src) I want to scan.

**Step 7.** **Build** the job. After successful build if you can see build logs it will show you the files and folder it has scanned and after scanning it has posted the analysis report to *SonarQube Server* you have integrated.

**Step 8.** From job dashboard, click on sonar icon or navigate to *Sonar server c*lick on P**rojects** (on header)you will see a new project with same project key you have given in ***sonar-scanner.properties*** file. Now you can go inside your project and analyse the report



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*