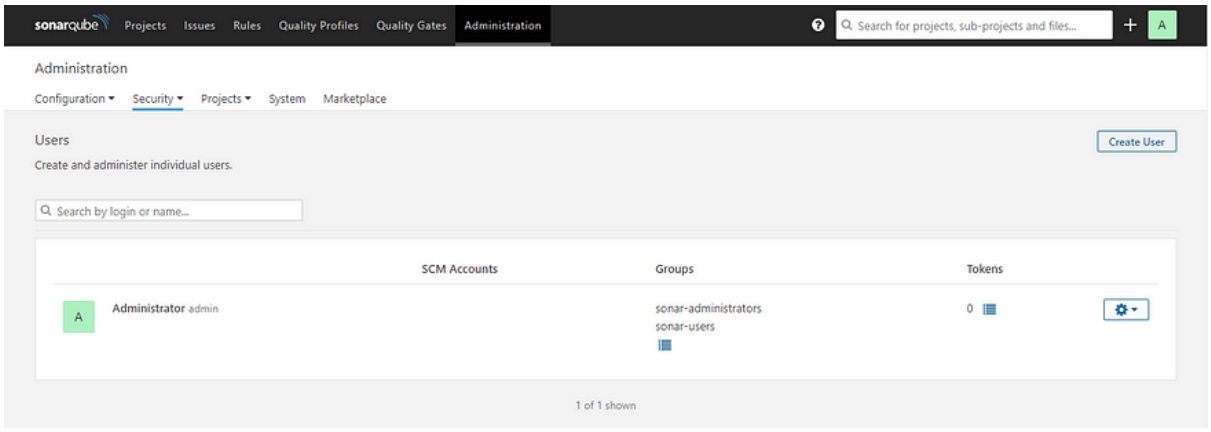
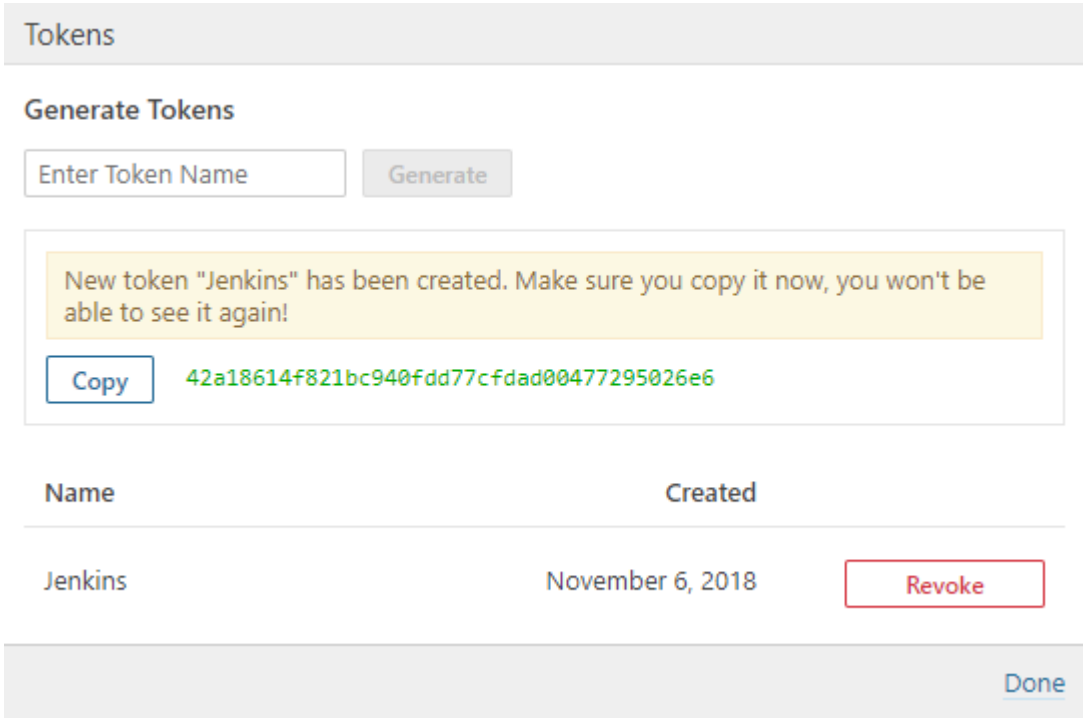
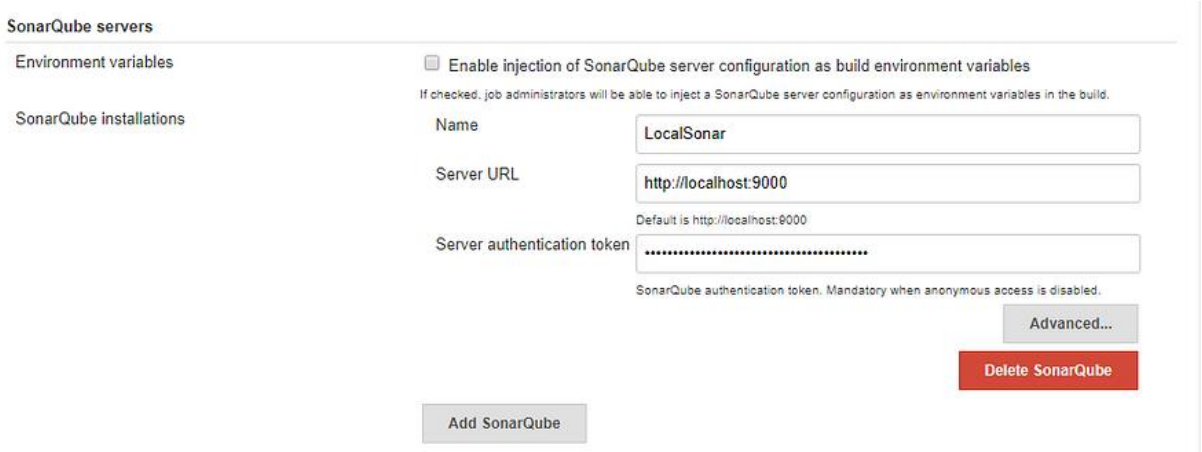
Integrate SonarQube with Jenkins

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.





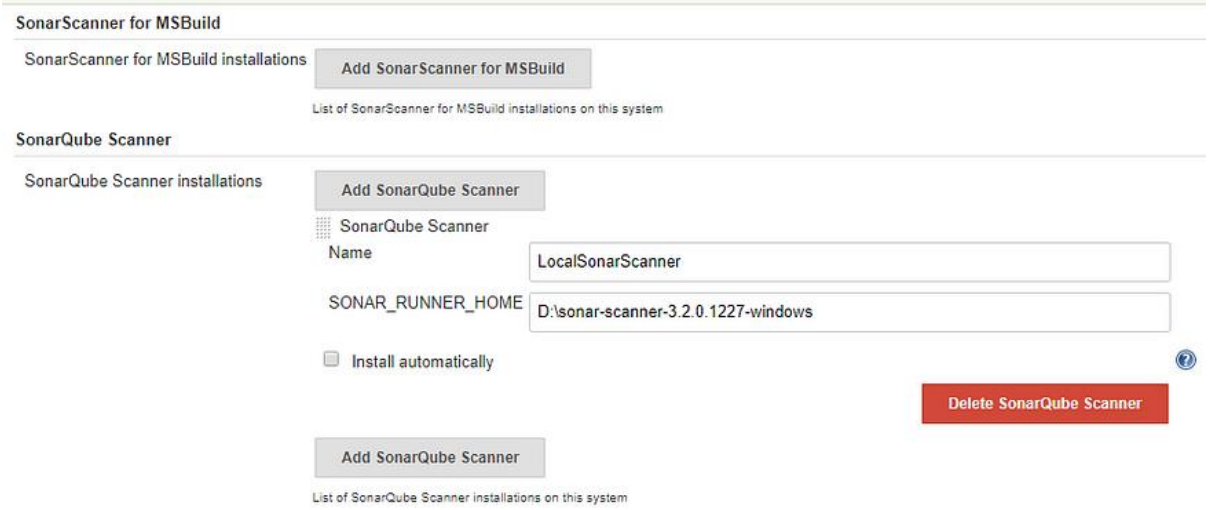
Step 2. Setup ***SonarQube*** with ***Jenkins***- Go to *Manage Jenkins* > *Configure system* > *SonarQube* server section > *Add SonarQube* > Name it, provide Server Url as ***http://<IP>:<port>*** > and authentication token copied from SonarQube Server > *Apply* and ***Save***



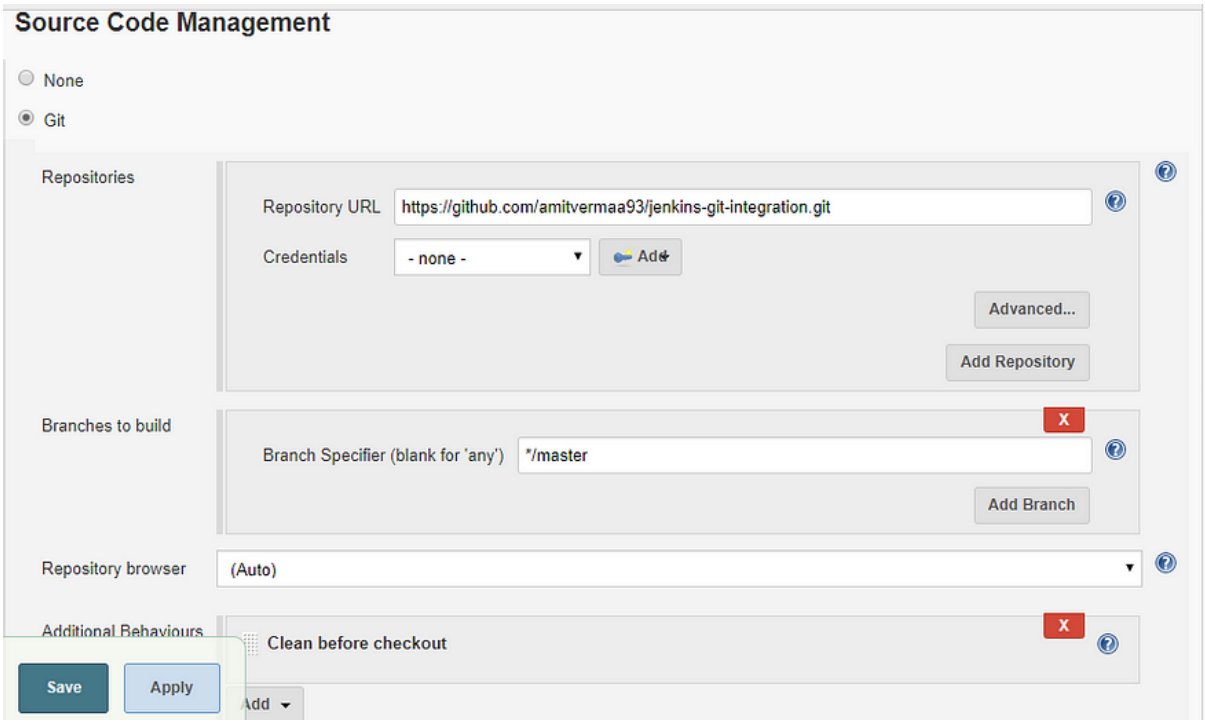
Step 3. Install ***SonarQube plugin*** to Jenkins. Go to *Manage Jenkins* > *Manage Plugins* > *Available* > Search for *SonarQube Scanner*> *Install.*

Download *SonarScanner* if you don't have ***https://docs.sonarqube.org/display/SCAN/Analyzing+with+SonarQube+Scanner***

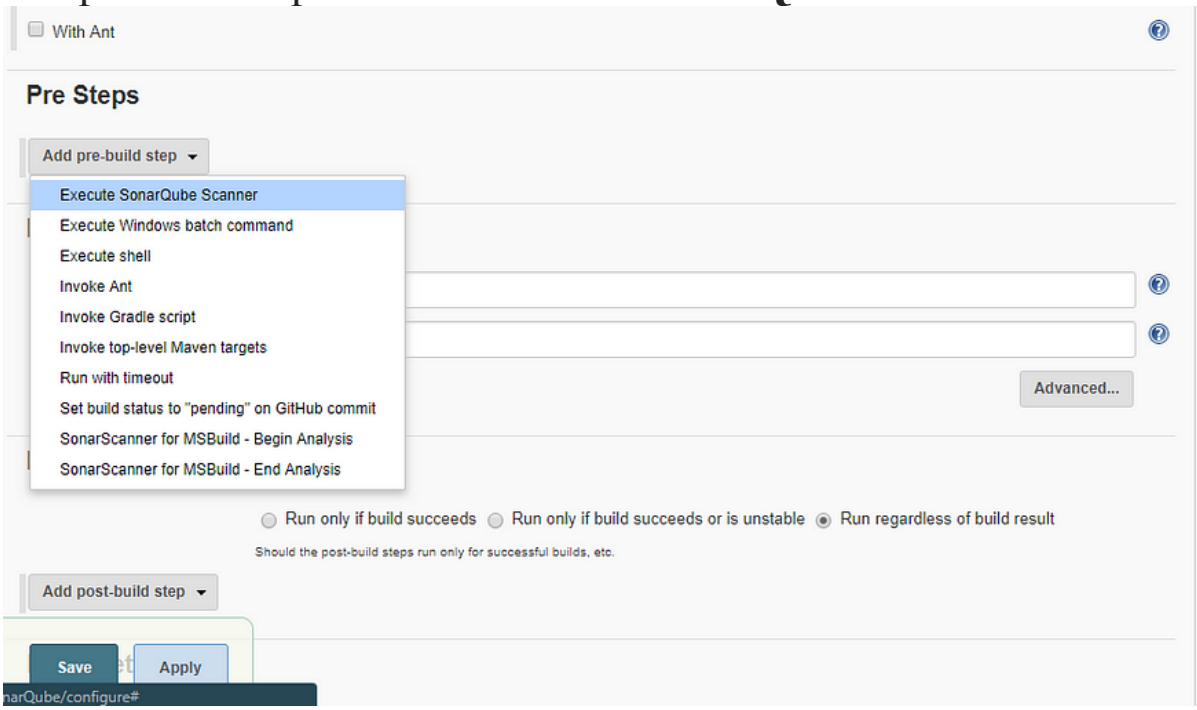
Configure ***Sonar Scanner*** in *Jenkins* : Go to *Mange Jenkins* > *Global Tool Configuration* > Scroll for *SonarQube Scanner* > *Add* sonar scanner > name it, uncheck if you already have sonar else it will automatically download for you and your sonar scanner setup will be done(in my case I already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



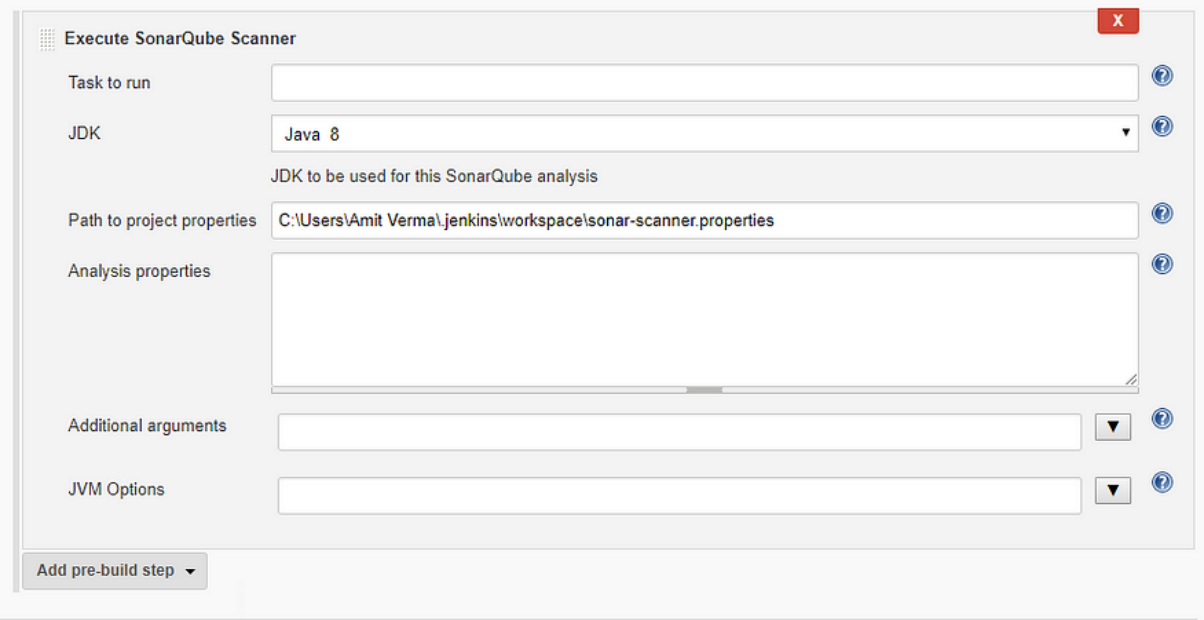
Step 4. Create a Job- *New Item* > Name and select a project type (in my case I am selecting *Maven* project you can opt for freestyle as well)



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***



Inside sonar-scanner.properties write below code —

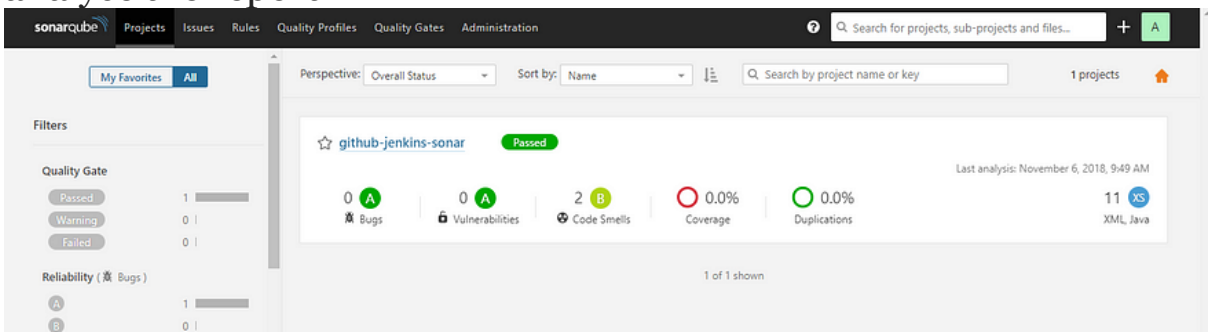
sonar.projectKey=github-jenkins-sonar
sonar.sources=./src

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)* that means 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* click 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



Example using declarative pipeline:

```
pipeline {
```

```
  agent none
  stages {
    stage("build & SonarQube analysis") {
      agent any
      steps {
        withSonarQubeEnv('My SonarQube Server') {
          sh 'mvn clean package sonar:sonar'
        }
      }
    }
    stage("Quality Gate") {
      steps {
        timeout(time: 1, unit: 'HOURS') {
          waitForQualityGate abortPipeline: true
        }
      }
    }
  }
}
```