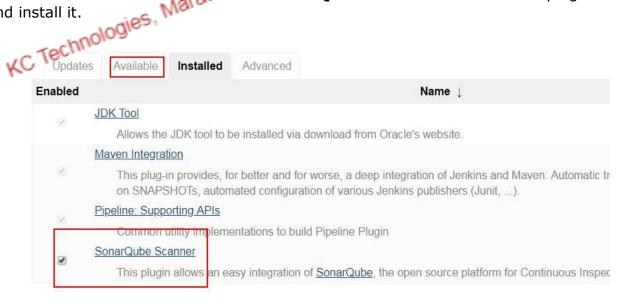
Integrating SonarQube with Jenkins for Maven projects using Pipeline Script 1013558586

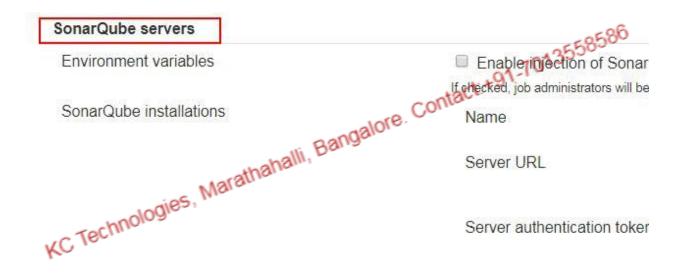
To integrate SonarQube with Jenkins, first we need to have sonar scanner plugin in Jenkins. Let's install it if we don't have it.

First login to Jenkins and go to Manage Jenkins, then go to Manage Plugins.

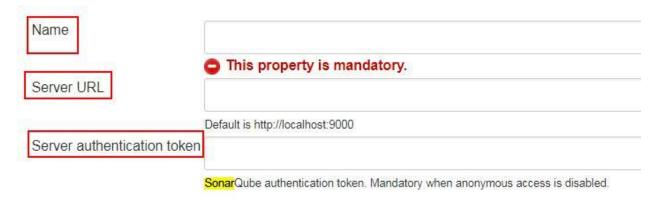
From the available section, search for **SonarQube Scanner** and select the plugin and install it.



Then we need to configure SonarQube instance. For this go to **Manage Jenkins**, then go to **Configure System**, then go to SonarQube servers' section.



We can see this option in Jenkins only after adding SonarQube Scanner plugin. Click on **Add SonarQube**. Then you need to provide few details.



Here, **Name** is the name we want to provide to SonarQube. We ca give any name as it is for our identification.

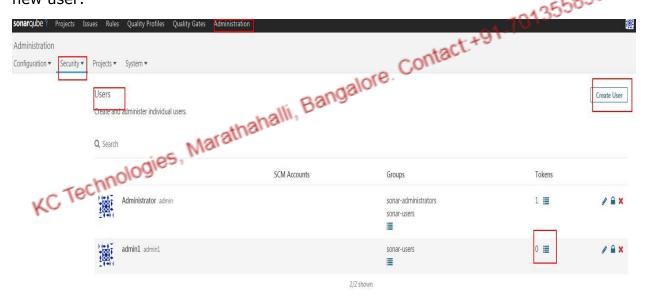
Server URL is the SonarQube server url on which we want to save the Sonar scanner report after generating.

Server authentication token: This is the taken we have to get from SonarQube server and is used by Jenkins to authenticate SonarQube while saving the report.

How to get SonarQube authentication token:

To get this token, first we have to start SonarQube server using **sudo**./**sonar.sh start** command on Linux and **StartSonar.bat start** on windows and then login.

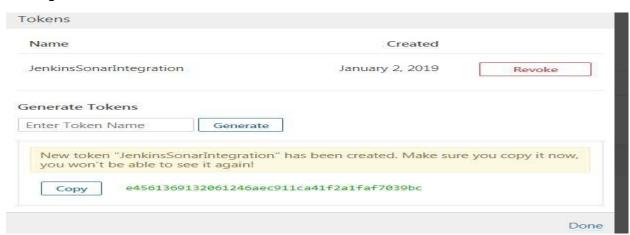
Then go to Administration, Security, Users, then click on **Create User** and create a new user.



Then click on Tokens, then you will get a popup.



Then give some token name there and click on Generate.



Copy the token and keep it sat	fe as you canno	ot get it back.
So, we created a Sonar Server		ract +91-7010
Now, go back to Jenkins configurations like below. SonarQube servers	gurations. Paste	e the token. Then you will have
SonarQube installations	Enable injection of SonarC	Qube server configuration as build environment variables able to inject a SonarQube server configuration as environment variables in the build.
SonarQube installations	Name	first-sonarscanning
VC. Technic	Server URL	http://localhost:9000
Ko		Default is http://localhost:9000
	Server authentication token	
		SonarQube authentication token. Mandatory when anonymous access is disabled.

Save configurations and go to Jenkins Pipeline job.

Then enter below script. Or else read from our DockerHubJenkinsIntegration project from our git location.

```
node{
 stage('SCM Checkout'){
   git 'https://github.com/KCTechnologiesDevOps/DockerHubJenkinsIntegration'
  }
 stage('Maven Packaging'){
   def mavenHome = tool name: 'maven-3', type: 'maven'
   sh "${mavenHome}/bin/mvn package"
 }
 stage('Static code analysis using SonarQube'){
   def mavenHome = tool name: 'maven-3', type: 'maven'
        withSonarQubeEnv('first-sonarscanning'){
   sh "${mavenHome}/bin/mvn sonar:sonar"
        }
  }
  stage('Sending Email Notification'){
   mail bcc: ", body: "'Hi Welcome to KC Technologies to learn Pipeline script to send Email
Notifications
   Thanks
```

```
KC Technologies"', cc: ", from: 'kctechnologiesdevops@gmail.com', replyTo: ", subject: 'Jenkins Email notification using pipeline script', to: 'kctechnologiesdevops@gmail.com'
}
stage('Sending Slack Notification'){
    slackSend baseUrl: 'https://hooks.slack.com/services/', channel: '#demo',
    color: 'good',
    message: 'Sending slack Notification',
    teamDomain' 'devops-realtime',
    tokenCredentialId: 'integration-token'
}
```

Failing Jenkins build when SonarQube quality check fails:

We will use the same example, what we discussed above.

Here, our email is to stop the Jenkins build execution when your project has not met the SonarQube quality profiles.

For this, let us take help of Jenkins **Pipeline Script Generat**or example. Go to Pipeline Script Generator, select the step **waitForQualityGate**:

Then click on the question mark symbol and take its help to create the script. There you can see some code snippet with **stage('Quality Gate')**.

It says we need to wait for certain to generate the SonarQube report.

So, we will wait for some time to get the report, then check the status is OK or not. If not ok, then mark the build as failed and send a slack notification.

For this take the pipeline script of previous example we discussed above and do some modifications to it by creating new stage like below.

```
stage("Quality Gate Status checking){
  timeout(time: 1, unit: 'HOURS') {
    def qg = waitForQualityGate()
    if (qg.status != 'OK') {
        slackSend baseUrl: 'https://hooks.slack.com/services/',
        channel: '#demo',
```

```
color: 'danger',
message: 'SonarQube code analysis has failed',
teamDomain: 'devops-realtime',
tokenCredentialId: 'integration-token'
error "Pipeline ajon has been failed do to quality gate failure: ${qg.status}"
}

timeout function in the above stage indicates, the script waits for maximum 1 hour
```

timeout function in the above stage indicates, the script waits for maximum 1 hour to get the response from SonarQube. It means the script will be paused for 1 hour to get the response from SonarQube.

waitForQualityGate() is the predefined Groovy function to get Sonar scanner report.

After that, we are checking the status is OK or not. If it is not OK, then we are marking the build as failure and sending slack notification stating the build failure message.

Finally, the complete script looks like below:

```
node{
 stage('SCM Checkout'){
   git 'https://github.com/KCTechnologiesDevOps/DockerHubJenkinsIntegration'
  }
  stage('Maven Packaging'){
   def mavenHome = tool name: 'maven-3', type: 'maven'
   sh "${mavenHome}/bin/mvn package"
  }
 stage('Static code analysis using SonarQube'){
   def mavenHome = tool name: 'maven-3', type: 'maven'
        withSonarQubeEnv('first-sonarscanning'){
   sh "${mavenHome}/bin/mvn sonar:sonar"
        }
  }
   stage("Quality Gate Status checking){
      timeout(time: 1, unit: 'HOURS') {
```

```
contact of the contac
                                  def qg = waitForQualityGate()
                                  if (qg.status != 'OK') {
                                                                 "Pipeline ajon has been failed do to quality gate failure: ${qg.status}"
       stage('Sending Email Notification'){
            mail bcc: ", body: "'Hi Welcome to KC Technologies to learn Pipeline script to send Email
Notifications
              Thanks
              KC Technologies''', cc: '', from: 'kctechnologiesdevops@gmail.com', replyTo: '', subject: 'Jenkins
Email notification using pipeline script', to: 'kctechnologiesdevops@gmail.com'
       stage('Sending Slack Notification'){
                 slackSend baseUrl: 'https://hooks.slack.com/services/',
                channel: '#demo',
                color: 'good',
                message: 'Sending slack Notification',
                teamDomain: 'devops-realtime',
                tokenCredentialId: 'integration-token'
       }
}
```

You can use this script directly in the Jenkins pipeline job or can keep in SCM and use it.

Then try to build the job and observer the console output. You may observer the build keep on running without any status as we have not configured SonarQube webhooks.

webhooks.

To get SonarQube webhooks, go to SonarQube, Administration, then click on webhooks.



Then give some name and Jenkins url followed by sonarqube-webhooks/

Ex: http://localhost:8080/sonarqube-webhooks/

/ at the end is mandatory. Then save the webhooks.

Webhooks		
Webhooks are used to notify external services when a project analysis is done. An HTTP POST request	Name	URL
including a JSON payload is sent to each of the first ten	jenkins_Integration	http://localhost:8080/sonarqube
provided URLs. Learn more in the Webhooks documentation.	jenkins_integration	The state of the s
Key; sonar.webhooks.global		

So, we created SonarQube webhook successfully.

After creating SonarQube webhook, run the Jenkins job again and observe the console output. There you can see the response step by step.

KC Technologies, Marathahalli, Bangalore. Contact: +91-7013558586