How to integrate Sonarqube with xml Job

1. First go to your project and find the pom.xml file.
2. Edit pom.xml file under build section

Add below code under <build><plugins> section

<!-- Jacoco plugin for SonarQube Code Coverage -->  
<plugin>  
 <groupId>org.jacoco</groupId>  
 <artifactId>jacoco-maven-plugin</artifactId>  
 <version>0.8.2</version>  
 <executions>  
 <execution>  
 <goals>  
 <goal>prepare-agent</goal>  
 </goals>  
 </execution>  
 <!-- attached to Maven test phase -->  
 <execution>  
 <id>report</id>  
 <phase>test</phase>  
 <goals>  
 <goal>report</goal>  
 </goals>  
 </execution>  
 </executions>  
</plugin>

3.then login to slave and go to your project folder then go to your pom.xml

file location and from there execute below command

cmd: **mvn clean install**

4. Commit and Push your changes into the respective repository

5.Now Go to the Jenkins groovy file where pipeline stages are defined for your job an add below

Stages after checkout.

1. (Optional Step) Add a new boolean parameter that will allow user to include or exclude the SonarQube Scan step  
   Add below into "parameters" section

booleanParam(name:'SONARQUBE\_SCAN', defaultValue:'true', description:'Whether to scan source code with SonarQube')

1. Define a variable "project" with the desired name.   
   ex:"FMC\_RCMPlugin"

String project = 'FMC\_projctname'

1. The newly added stage will look like this:

stage('SonarQube Scan', {  
 steps {  
 script {  
 dir("build") {\\  
 if (params.SONARQUBE\_SCAN.toBoolean()) {  
 print("Starting code scan using SonarQube. sonar.projectName=${project}")  
 withSonarQubeEnv(' SonarQubeITPROD') {  
 sh "mvn sonar::sonar -Dsonar.projectName=${project}"  
 }  
 } else {  
 print("Skipping code scan using SonarQube for sonar.projectName=${project} as parameter params.SONARQUBE\_SCAN has value [${params.SONARQUBE\_SCAN}]")  
 }  
 }  
 }  
 }  
})

1. Add a new stage named SonarQube Quality Gate" after the "SonarQube Scan" stage

The newly added stage will look like this:

stage('SonarQube Quality Gate') {  
 when {  
 environment name: 'params.SONARQUBE\_SCAN', value: 'true'  
 }  
 steps {  
 timeout(time: 10, unit: 'MINUTES') {  
 // Parameter indicates whether to set pipeline to UNSTABLE if Quality Gate fails  
 // true = set pipeline to UNSTABLE, false = don't  
 waitForQualityGate abortPipeline: true  
 }  
 }  
}

6. Run the CI Build Plan. It should show the newly added two stages

7.If the build is successful - please check the code coverage report on <https://sonar.nice.com/projects>