**SONARQUBE**

* It is a open source platform to maintain source code quality. It is written in java. But, can analyze 20 different programming languages.
* Default port = **9000**.
* It analyzes architecture & design, unit tests, potential bugs, coding rules etc.
* You can access sonarqube from web browser with your server ip and sonarqube port.
* You should always use sonar as normal user. Root user can’t sonar.
* One of the main issues with sonar is. Your server should have atleast **4** gm ram. If you have less than 4 gb ram, sonar stops automatically.

**INSTALL SONAR**

* Before we install sonarqube, we have to install java and set home path for java.
* Install java = **yum install java –y**.
* Go to techadmin site, download java jdk file.

**wget --no-cookies --no-check-certificate --header "Cookie: gpw\_e24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie"** [**http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz**](http://download.oracle.com/otn-pub/java/jdk/8u161-b12/2f38c3b165be4555a1fa6e98c45e0808/jdk-8u161-linux-x64.tar.gz)

* Extractthe file **= tar –xvzf jdk-8u161-linux-x64.tar.gz**
* Go to **/opt/jdk1.8.0\_161/**
* Install java8 with alternatives by typing these commands.

**alternatives --install /usr/bin/java java /opt/jdk1.8.0\_161/bin/java 2**

**alternatives --config java**

* select java version that you have installed and type these commands.

**alternatives --install /usr/bin/jar jar /opt/jdk1.8.0\_161/bin/jar 2**

**alternatives --install /usr/bin/javac javac /opt/jdk1.8.0\_161/bin/javac 2**

**alternatives --set jar /opt/jdk1.8.0\_161/bin/jar**

**alternatives --set javac /opt/jdk1.8.0\_161/bin/javac**

* after executing these commands, check the java -version whether we selected correct version (or) not.
* Export the path of java to use java in your system anywhere.
* **export JAVA\_HOME=/opt/jdk1.8.0\_161**
* **export JRE\_HOME=/opt/jdk1.8.0\_161/jre**
* **export PATH=$PATH:/opt/jdk1.8.0\_161/bin:/opt/jdk1.8.0\_161/jre/bin**
* After exporting the java paths, check with **echo $JAVA\_HOME** command whether it is successfully applied (or) not.
* To export the path permanently, copy the exported lines to **/etc/environment** and update the file with source command.

**source /etc/environment**

* You can also export by creating java.sh file in **/etc/profile.d/** dir, copy the exported lines and update the file with source command.

**source /etc/profile.d/java.sh**

* Nowinstall a database (mysql in my case). Because**,** Sonarqube needs a database to store the metrics of analysis...It supports all types of DB available in the market.
* To install mysql, you need to download mysql rpm package and install the rpm package.

**wget** [**https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm**](https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm)

**rpm -ivh mysql57-community-release-el7-9.noarch.rpm =** install the rpm package**.**

**yum install mysql-server** = install mysql server.

**sudo systemctl start mysqld** = start mysql.

* During the installation process, a temporary password is generated for the MySQL root user. Locate it in the **mysqld.log** with this command. Make a note of that password with this command.

**grep 'temporary password' /var/log/mysqld.log**

* After this, do mysql secure installation.

**mysql\_secure\_installation**

**Type your old password.**

**Enter new password and confirm it.**

**Do you want to change root password = NO**

**Remove anonymous users = Yes**

**Disallow root login remotely = No**

**Remove test db and access to it = yes**

**reload privilege tables = yes**

* Now, we have installed mysql successfully and secured it.
* Login into mysql with user root. If you have successfully logged into db it means that you have configured mysql correctly.
* Upto now, we have installed java and mysql, which uses by sonarqube.
* Now, Download sonarqube zip file from the official site and unzip the file.
* Login into your database and create a database and user for sonarqube as below.

**CREATE DATABASE** **sonar CHARACTER SET utf8 COLLATE utf8\_general\_ci;**

**CREATE USER '****sonar' IDENTIFIED BY 'Sonar@1234';**

**GRANT ALL ON sonar.\* TO 'sonar'****@'%' IDENTIFIED BY 'Sonar@1234';**

**GRANT ALL ON sonar.\* TO 'sonar'@'localhost' IDENTIFIED BY 'Sonar@1234';**

**FLUSH PRIVILEGES;**

**@'% ---🡪 refer as remote access**

**Login as database user**

**Mysql –u sonar –pSonar@123**

**Show databases;**

**Downlod and Installation of SonarQube:**

**Go to google download sonar qube package from website its in zip**

**Yum install unzip -y**

* Go to **/sonarqube/conf/sonar.properties**,
* In the database section,

**Uncomment the jdbc.username and password lines.**

**Add your db username and password.**

**Uncomment jdbc.db.url line, for which db you are using and save the file.**

* Now, all the sonar metrics will be saved in the database that you selected.
* By default, sonar can’t start as root, you have to start as normal user. You have to add that user to sonar.sh file.
* Go to **/sonarQube/bin/linux64bit/.** Edit the **sonar.sh** and add this.

**RUN\_AS\_USER=username.**

* Give execute permissions to sonar dir.

**chmod –R 777 sonarQube**

* Change the ownership and group of sonar dir to the user, from which you want to start sonar.

**chown –R username sonarQube**

**chgrp –R username sonarQube**

* Start the service to test whether sonar is working (or) not…

**/sonarqube/bin/linux-64bit/sonar sh start –** To start sonar.

**/sonarqube/bin/linux-64bit/sonar.sh stop –** To stop sonar.

**/sonarqube/bin/linux-64bit/sonar.sh restart –** To restart sonar.

**/sonarqube/bin/linux-64bit/sonar.sh console –** To see the output.

* Based on the architecture of your server, the start and stop scripts path will change.
* If you have 64bit system, scripts will be stored in linux-64bit dir.
* If you have 32bit system, scripts will be stored in linux-32bit dir.
* Go to browser, type your ip and sonar port(9000), you can see sonar home page.
* If you want to use your ip (or) hostname instead of localhost in the sonarqube url,
* Go to sonar.properties file,

**In web server section.**

**Sonar.web.host = type your server ip (or) hostname.**

* Restart the sonarqube and now you can access sonar with your ip and sonar port in browser.
* By default, username and password to login to sonar dashboard is **admin** for both.
* You can change the password once you logged into sonar dashboard.

**SONAR-MAVEN INTEGRATION**

* To use sonarqube with maven, you have to install a plugin by making an entry in maven **settings.xml** file.
* Go to settings.xml,
* Under **pluginsGroups** section. Down the second ‘🡪’ mark, paste this.

**<pluginGroup>org.sonarsource.scanner.maven</pluginGroup>**

* In **profiles** section .down the ‘**🡪’** mark and above the activate profiles section, Remove **</profiles>** line and paste this

**<profile>**

**<id>sonar</id>**

**<activation>**

**<activeByDefault>true</activeByDefault>**

**</activation>**

**<properties>**

**<!-- Optional URL to server. Default value is http://localhost:9000 -->**

**<sonar.host.url>**

**http://IP:9000**

**</sonar.host.url>**

**</properties>**

**</profile>**

**</profiles>**

* It will download the plugin from given website above in plugingroup section and it will detect the sonar from the given url.
* After copying this code in settings.xml file, run any maven cmds to see whether we did correct (or) not.
* To test sonar in linux, go to your project **pom.xml** dir and run **mvn sonar:sonar**….It will test the code and give you the errors in the code in sonar GUI dashboard along with your project name.
* Login to your sonar GUI dashboard with your username and password, there you can see your project name. Inside the project name you can see all your code along with errors if there are any.

**SONAR-JENKINS INTEGRATION**

* You can integrate sonarqube with Jenkins for continuous testing. It will test the code quality right after the build job is finished. It will show errors in the code in sonar GUI.
* First, you have to download **sonar-scanner plugin** (sonarqube in old versions).
* Afetr installing the plugin. Go to configure system,

**In sonarqube servers section.**

**Click add sonarqube.**

**Type name.**

**Sonar server url.**

**Sonar version.**

**Sonar account username and password** if it asks (by default, **admin**).

**Click, Save.**

* Go to Global tool configuration,

**In sonar installations section.**

**Click add sonarqubescanner.**

**Type a name.**

**Give Sonarqube home path (if you installed sonar scanner in your server).**

**If not, click install automatically and select the scanner version to install.**

**Click, save.**

* We have configured everything with sonar and Jenkins.
* After configuring sonar in Jenkins, while creating a project.
* In the post build step, select **execute sonarqube scanner.**
* In the analysis properties section, we have to mention analysis parameters, based on these parameters, sonar will scan our project.

**sonar.host.url=**http://yourip:9000/

**sonar.projectName=**myproject (your project name to display in sonar UI)

**sonar.projectKey=**myproject (can be anything, must be unique)

**sonar.projectVersion=**1.0 (can be anything)

**sonar.sources=**path/to/your/code (your code path)

**sonar.language=**java (in my case)

**sonar.login=**username (to login to sonar UI)

**sonar.password=**password (to login to sonar UI)

**sonar.projectBaseDir=**Use this if you need analysis should takes place in different dir rather than default dir(default=/.jenkins/workspace/project). Give write permissions to this directory. It is where sonar.working.directory will be created.

* Click, save and build the project.
* Now, once the build finished successfully, it starts sonar analsys and push the code errors and bugs to sonar dashboard.

**QUALITY PROFILES**

* Quality profile service is central for sonarqube. This is where you define your requirements by defining set of **rules** which need to analyze on top of the particular language.
* Every language has default sonar profile (sonar way), which is used while analyzing.
* You can create your own profile and add your custom rules to that profile to analyze the code.
* You can create as many profiles as you want in sonarqube.
* To create a profile.

**Click on quality profiles tab.**

**Click create.**

**Give a name to profile.**

**Select language to assign the role.**

* Once after the profile creation is finished, you will get a new profile with zero rules.
* You can activate the rules inside the new profile that you have created.
* You can associate projects with this profile. After associating, while analyzing the project, this profile will be executed aganaist the project.
* To associate a profile to project.

**Go to profile to associate.**

**Click manage projects.**

**Click ALL**

**Select projects to associate with this profile.**

**Click, close.**

* You can take backup of a profile by clicking on the backup. It will download an xml file with all the rules.
* You can restore the profile by uploading the xml file to sonar.
* You can copy a profile and create a new profile with the same rules as copied one.
* You can activate and deactivate the rules in profiles as per our requirement.
* There is third-party plugin called find bugs-plugin, we can use this plugin to analyze the project based on this plugin rules.
* We have to download find bugs-plugin from sonar market place and restart the server.
* Go to quality profiles and check for findbugs to know whether find bugs is successfully installed (or) not.
* Once findbugs downloaded, you can set the plugin as default analyzer for the projects. And for every time we analyze the code, it will take the rules form findbugs-plugin.
* You have to select the projects to use this findbugs-plugin for analyzing.

**QUALITY GATES**

* Quality gates are simple Boolean thresholds set on project measures. A project must pass each and every thresholds in order to pass the quality gate as a whole. It defines the quality of your project.
* By default, only one quality gate will be there and used for all the projects, you can create a new one and use it.
* You can create new quality gates and add conditions to that gates to check the projects. you have to assign this quality gate to project, which will use while analyzing the project. If the conditions are passed as we specified, then sonar passes quality gate for the project.
* If the conditions are not met with quality gate, it fails the quality gate for the project and it means there is something wrong in your project code.

**To create a new gate.**

**Click on quality gate tab.**

**Click create.**

**Give a name to gate.**

**Set conditions to gate.**

**Select project to associate with gate.**