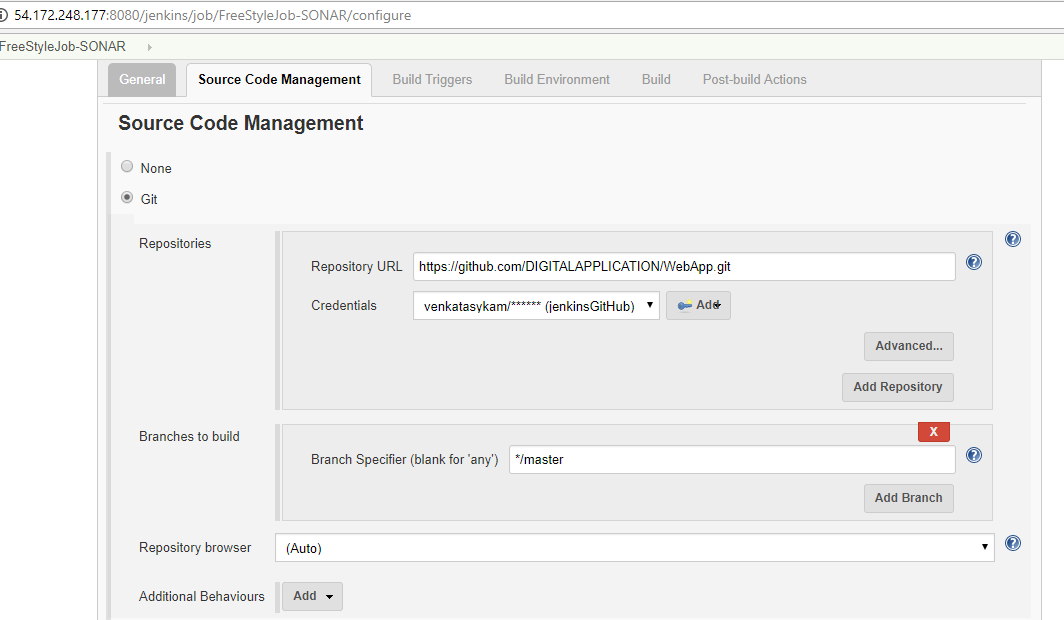
We are going to configure a Jenkins job to run the sonar code analysis and publish the report to sonar dashboard & server. Please find the steps below.

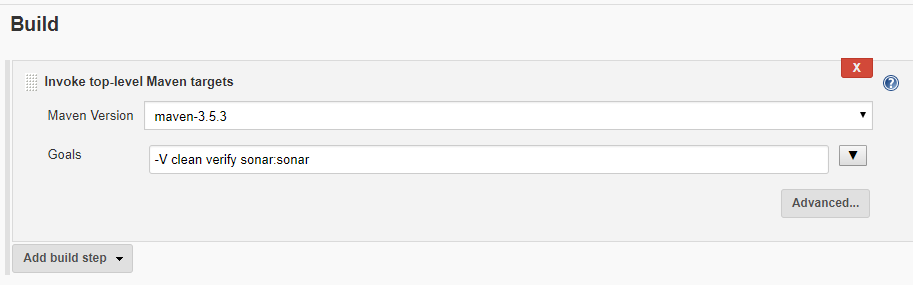
1. Please refer **Java-Tomcat-maven-git-Jenkins.sh** doc which is in Phase-1 & install the required tools on RHEL machine as mentioned over the same doc. (you can choose any type of OS machine if you are familiar with various types of Linux/unix).
2. Configure a Jenkins job as
   1. Checkout the code from GitHub.
   2. Run the maven goals as to compile & package the code.
   3. The maven goal, **sonar:sonar** to analyse the code. This will take care of analyse & uploading the report to sonar server.
3. Check the sonar server/dashboard for any issues/bugs etc in the code.

Create a Jenkins job with the template freestyle/maven project and configure GitHub code repo with credentials. (Please refer Phase-1 docs to know how to configure it.)

**SCM section**:



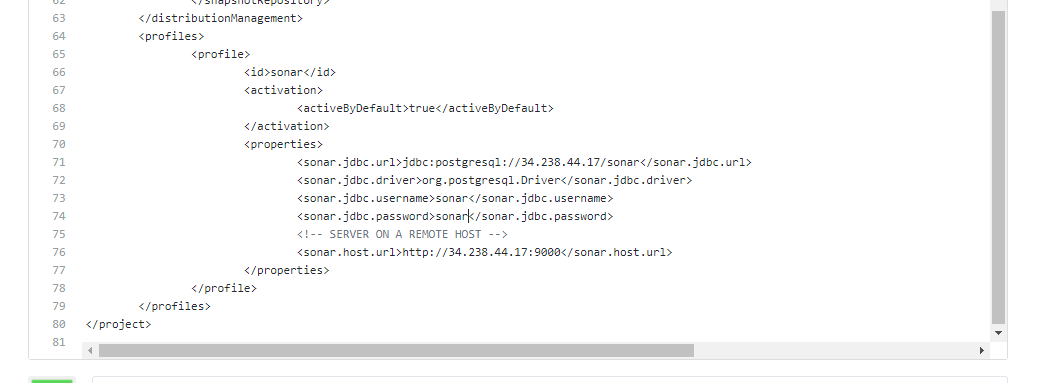
**Build section**:



**Sonar properties**: Jenkins/maven should know the sonar server details, db credentials etc to upload the analysis report to sonar server.

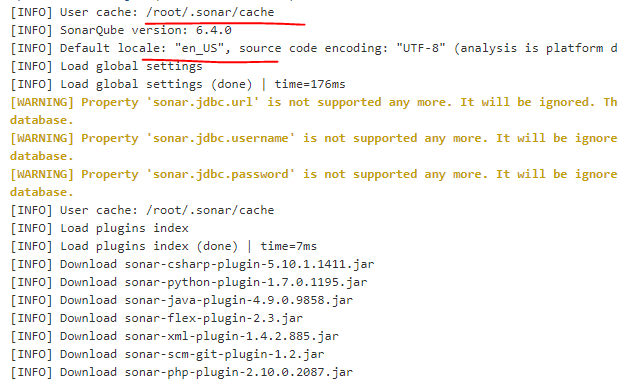
Paste the below snippet either in maven settings.xml file OR project’s pom.xml file. Both the ways it will work. In this example, I am adding the below sonar profile to the project pom.xml file. Please replace with your publicIP address. Make sure the maven profile is <activeByDefault>true</activeByDefault>. If it is false, add the profile details to maven goals i.e., clean verify sonar:sonar -Psonar.

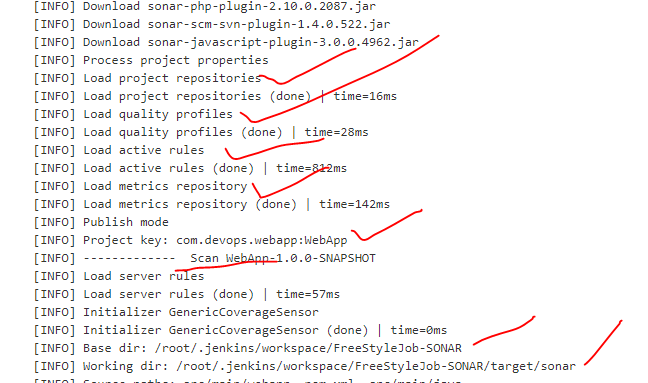
|  |
| --- |
| <profile>  <id>sonar</id>  <activation>  <activeByDefault>true</activeByDefault>  </activation>  <properties>  <sonar.jdbc.url>jdbc:postgresql://<**publicIP**>/sonar</sonar.jdbc.url>  <sonar.jdbc.driver>org.postgresql.Driver</sonar.jdbc.driver>  <sonar.jdbc.username>sonar</sonar.jdbc.username>  <sonar.jdbc.password>sonar</sonar.jdbc.password>  <!-- SERVER ON A REMOTE HOST -->  <sonar.host.url>http://**<publicIP>:**9000</sonar.host.url>  </properties>  </profile> |

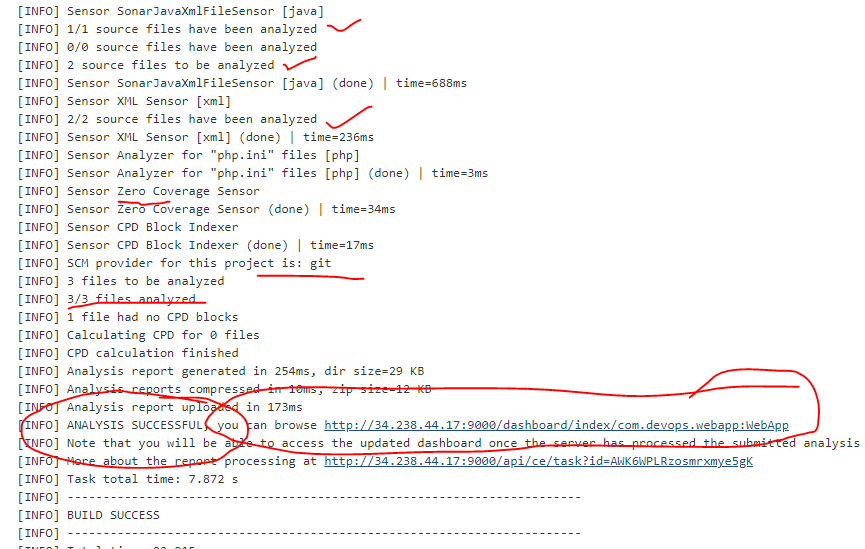


Now, trigger the build and monitor the build console.

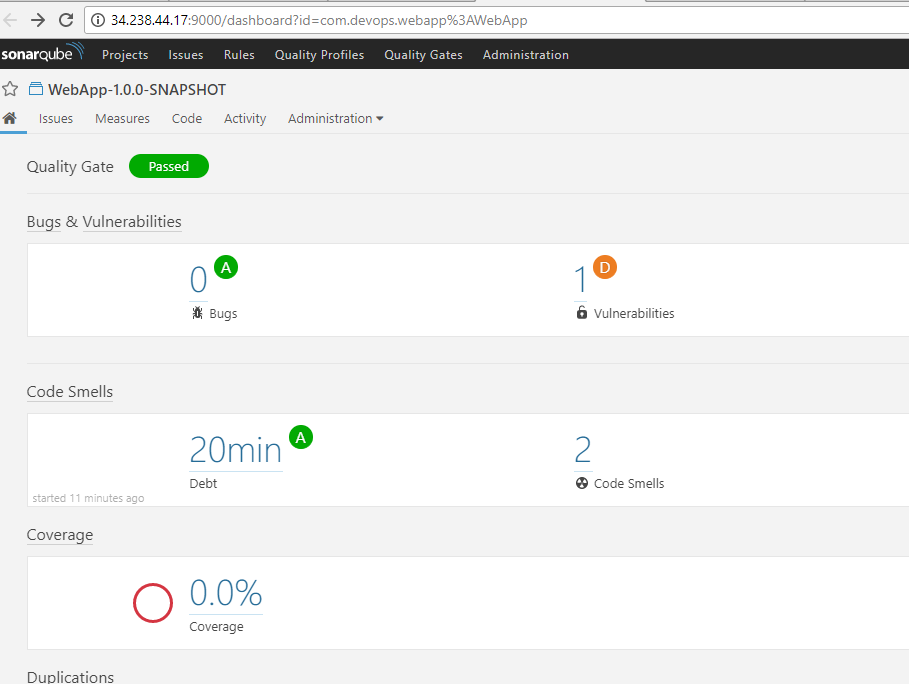








Click on the above link, it will navigate to sonar server/dashboard, and there you can see the code analysis report.



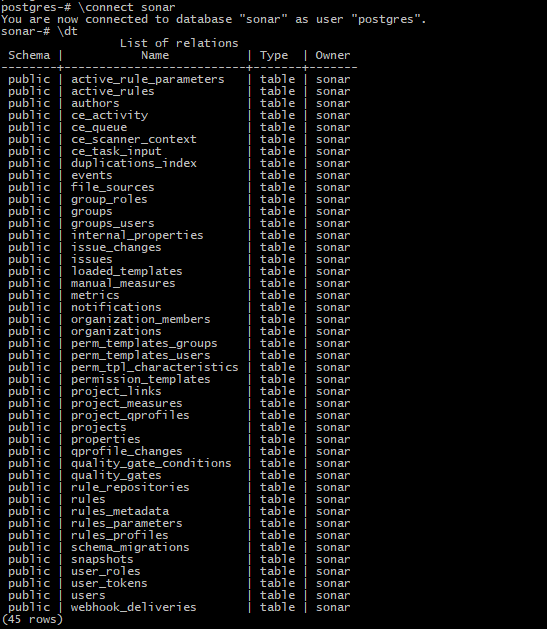
Let’s check the data base, how many default tables created by sonar for the first time.

su – postgres

psql

\connect sonar

\dt



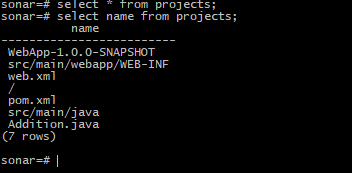


You can check all the tables how data is storing. Here is the result of one query “SELECT \* FROM projects;”



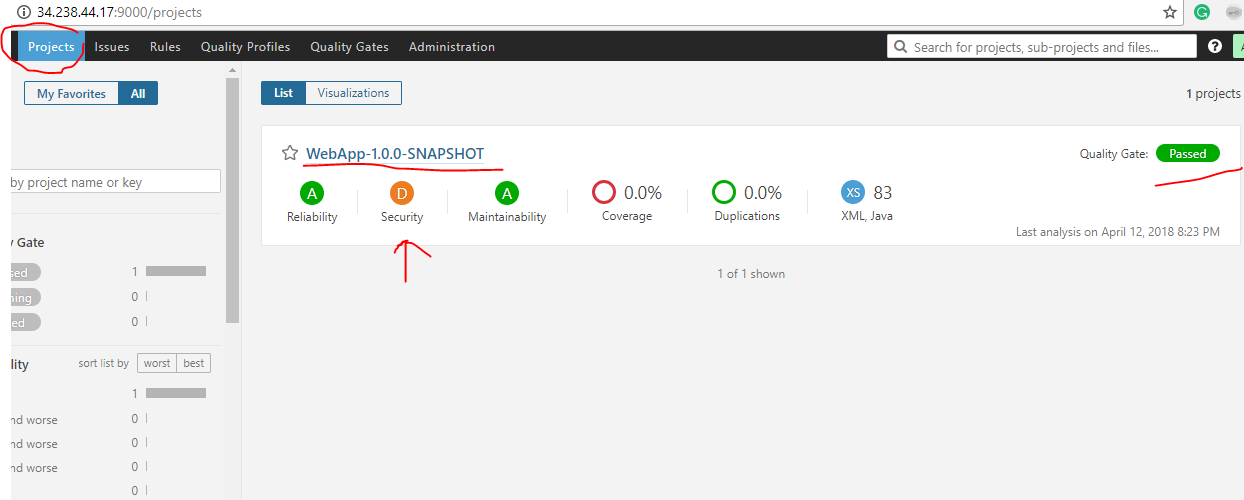
Type “\q” to come out from the table(if the table displays too many rows).

Here is another query “SELECT name FROM projects;” to display only one column ‘name’.

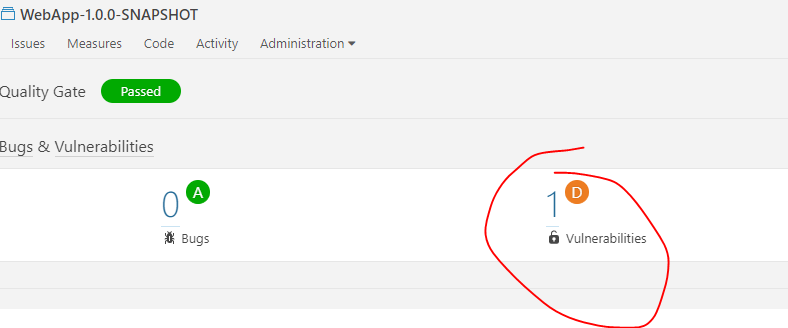


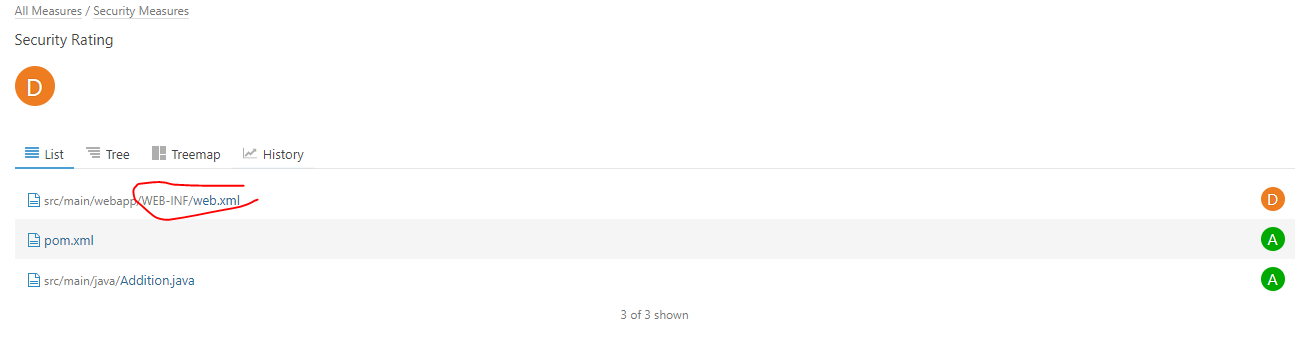
This will prove you that, sonar is connected to PostgreSql successfully & default tables are created by sonar.

Go to Sonar dashboard & click on projects, you will see the list of projects analysed. So far we build only one project. There is one security violation, let’s check the full error details. Click on the project and see the error.

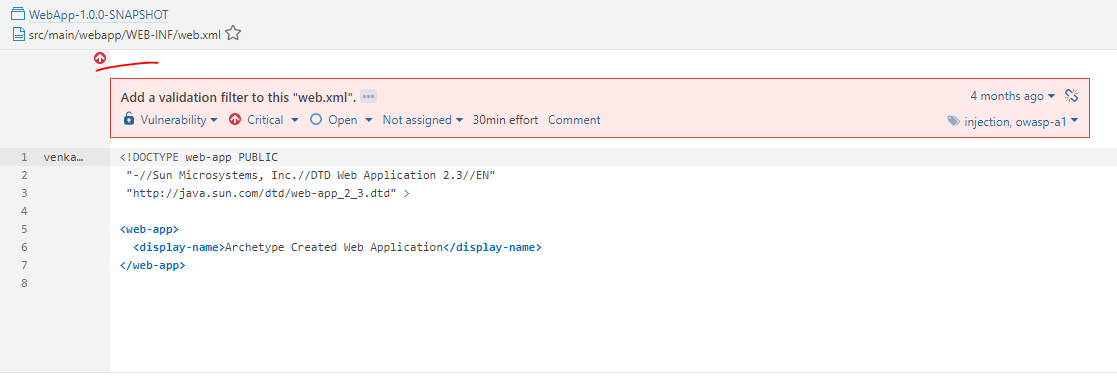


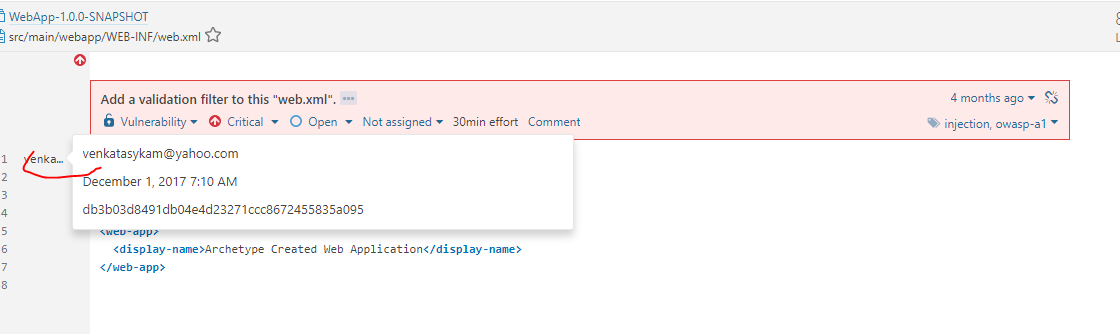
Click on it.



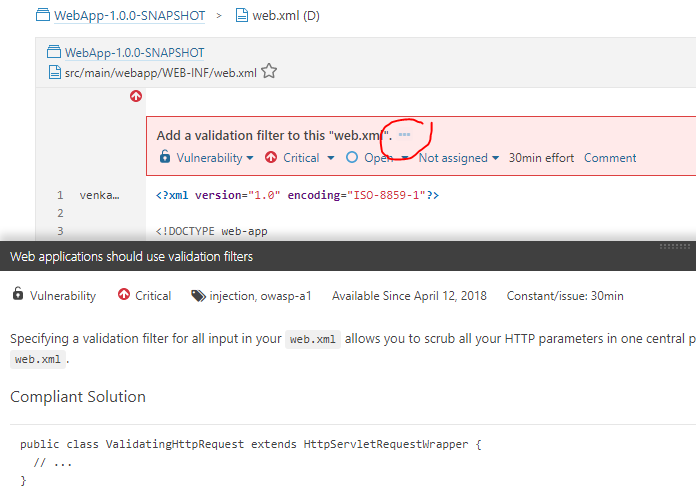


This  is the critical violation.

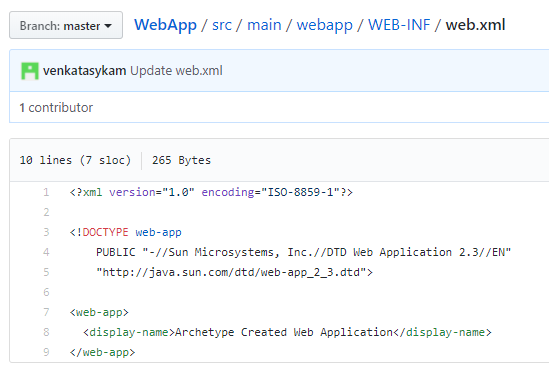




Click on three dots ... to see the full error log.



This is the original web.xml file.



Since there is a critical violation but sonar analysis is success & Jenkins job is success.