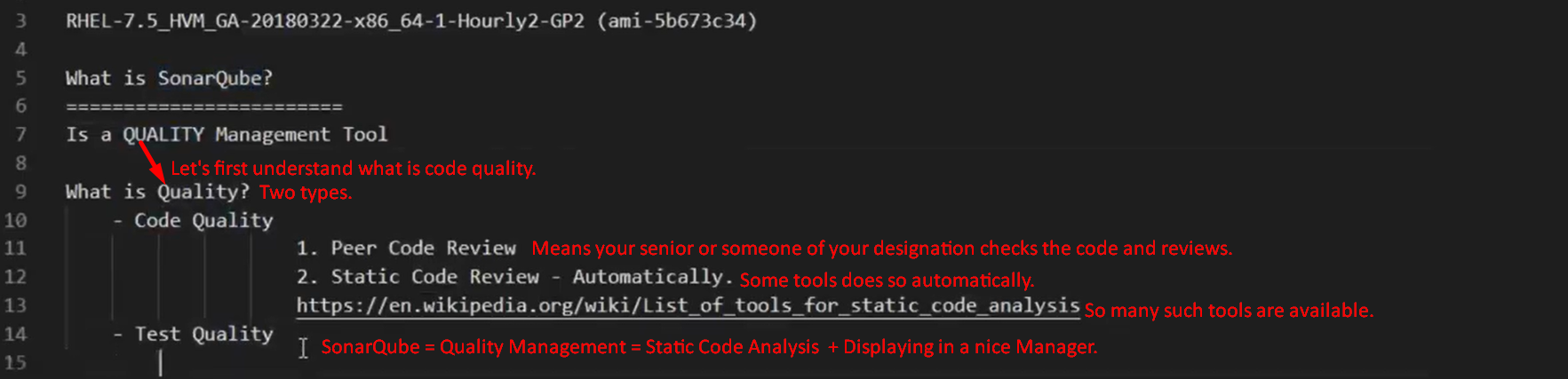
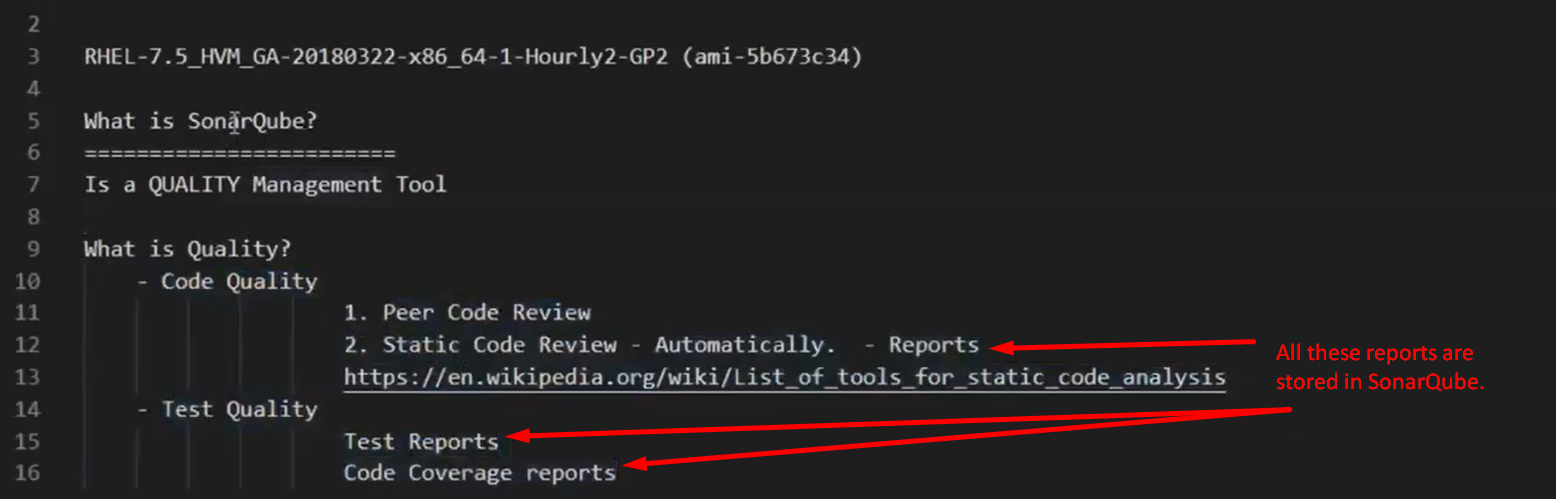
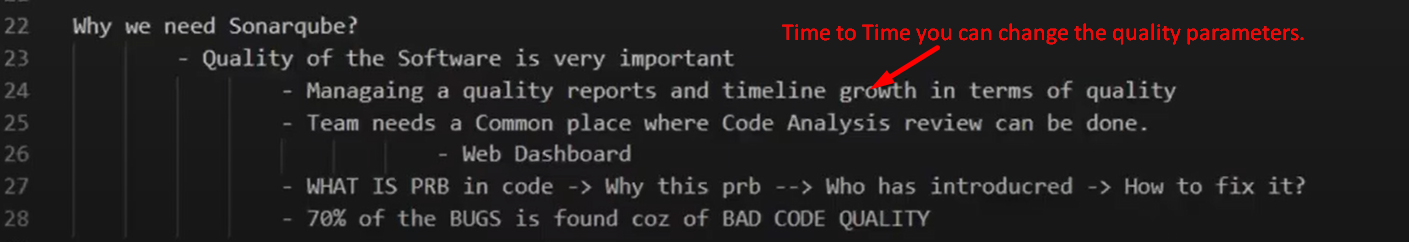
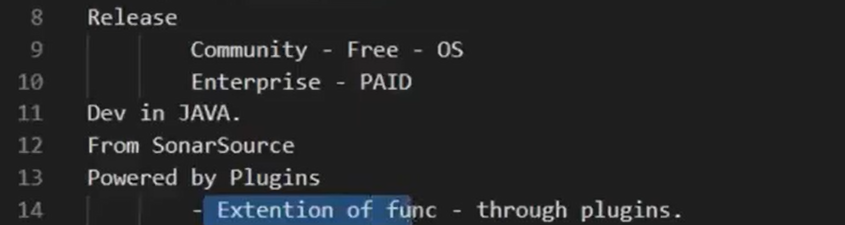
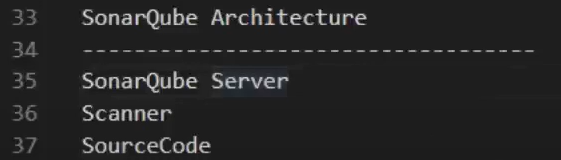
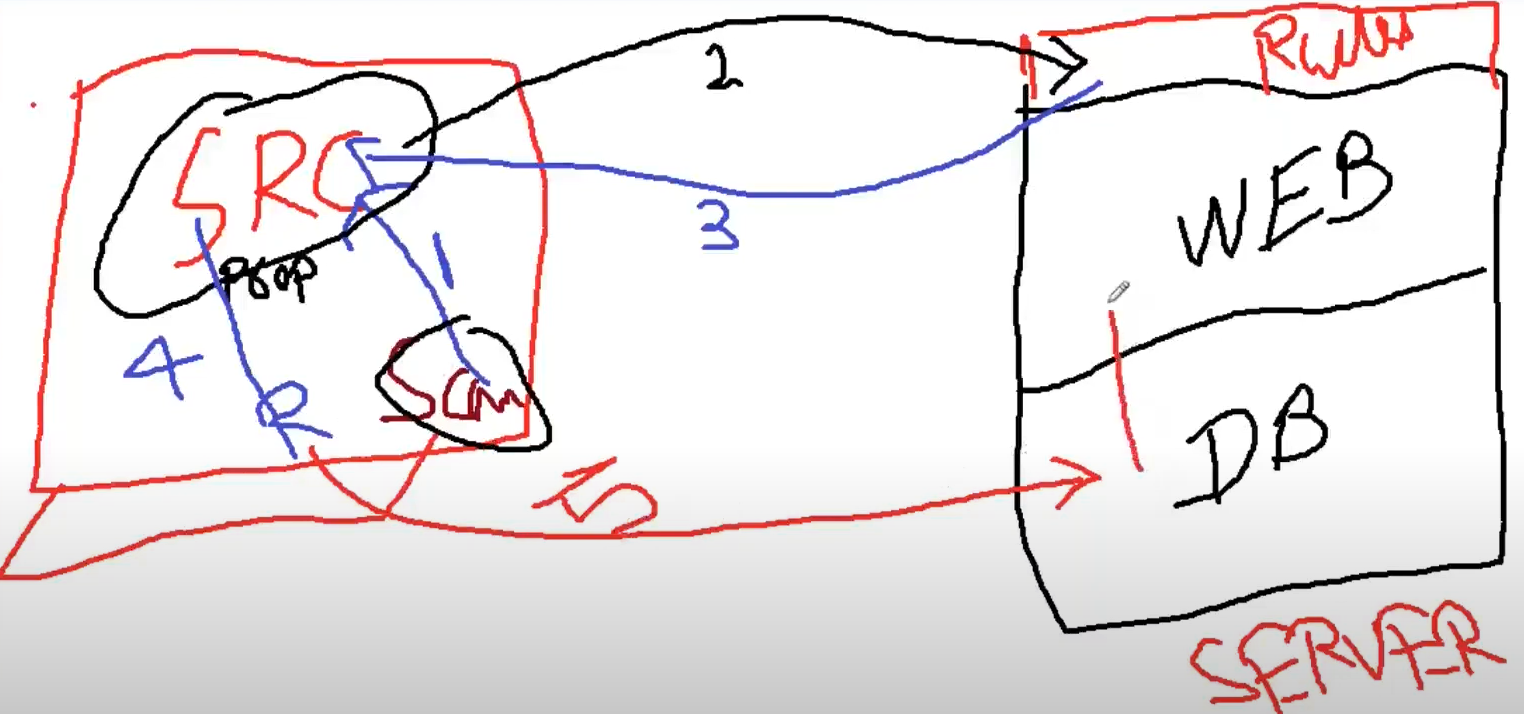
1. <https://www.youtube.com/watch?v=UjCVCvdI73w>
2. 
3. These tools do the code analysis and generate a report.   
   That report is Quality Report for the code.  
   That needs to be managed somewhere.  
   So, SonarQube is to take all the analysis and show in a nice manager for all.
4. 
5. Bad Code Quality:
6. Most of the time, bugs are due to bad Code Quality: Such as
   1. Performance Issue.
   2. Security Issue.
   3. Duplicate Issues.
   4. Architecture Issue.
   5. Loading Issue.
7. That is why we use SonarQube.
8. 
9. 

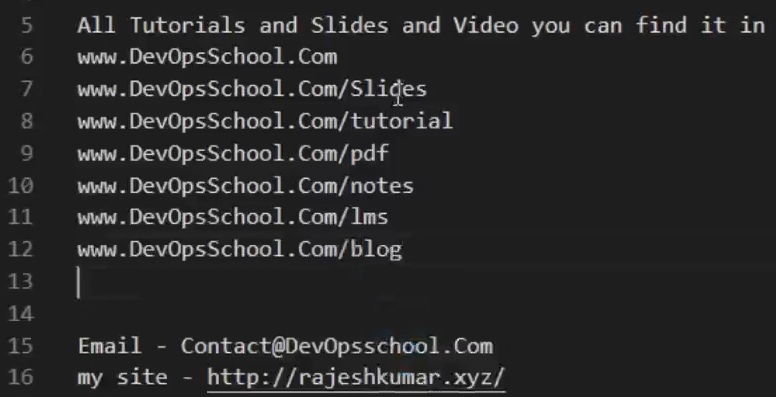
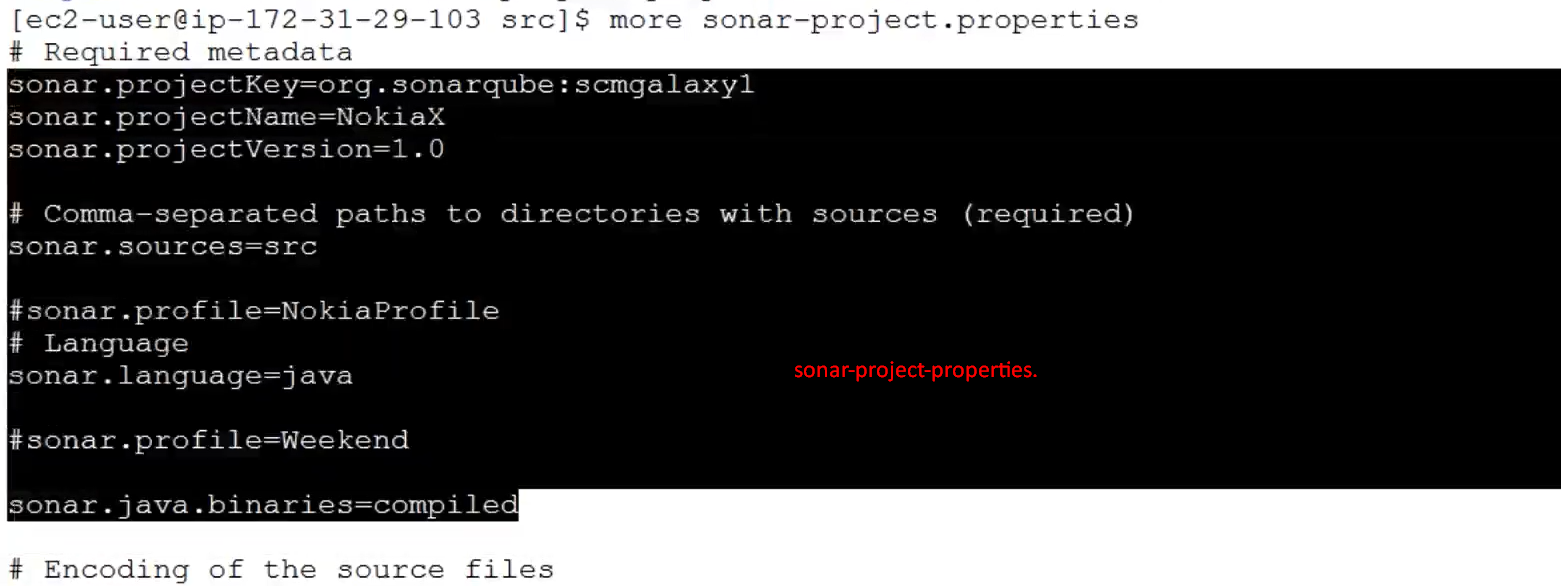
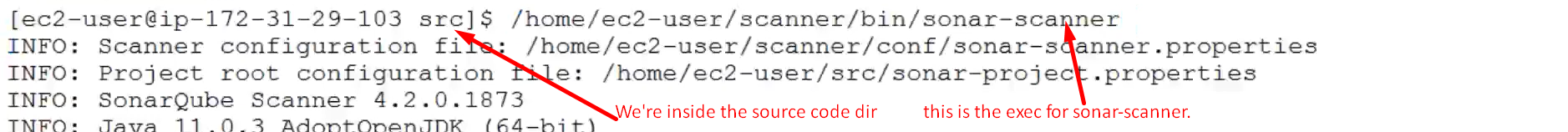
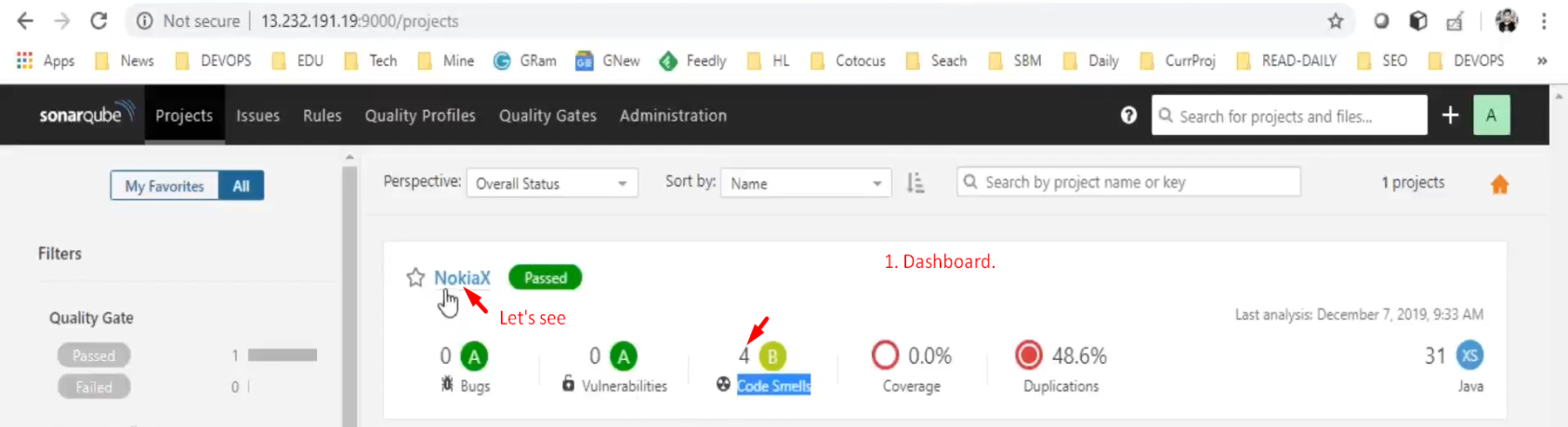
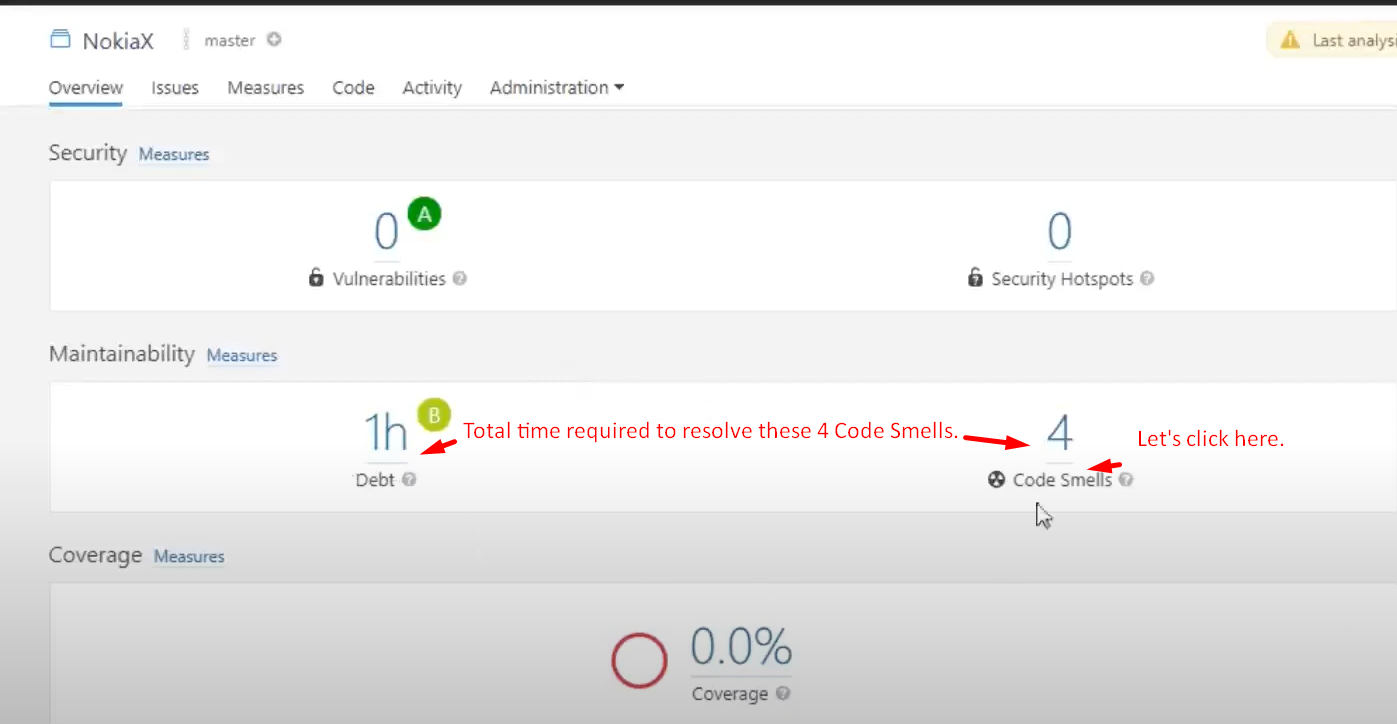
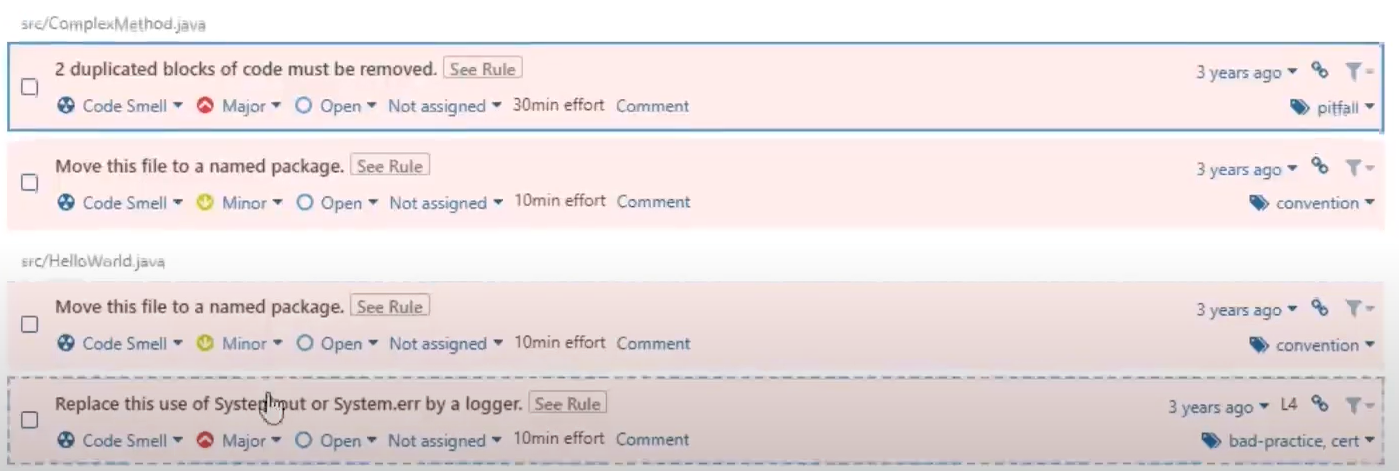
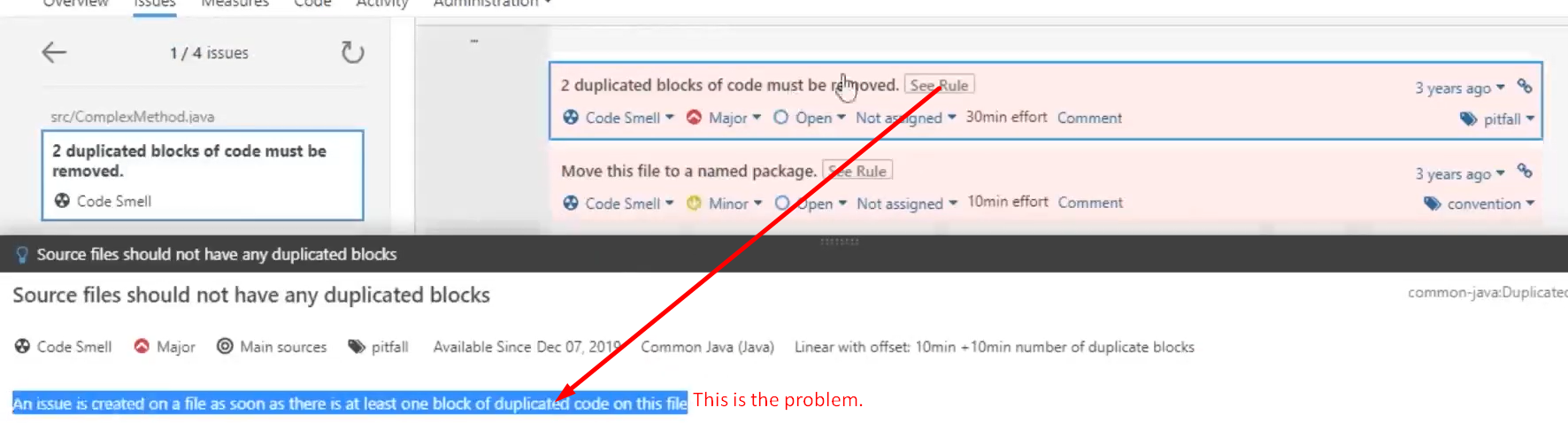
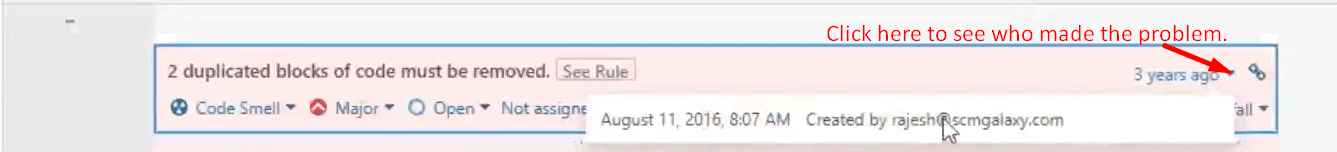
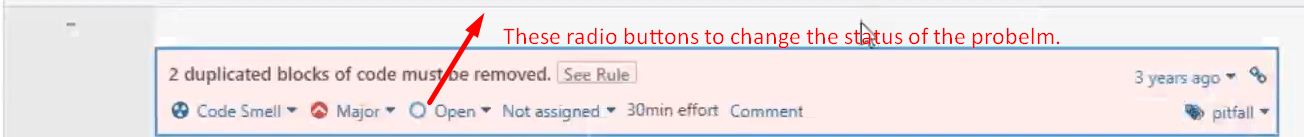
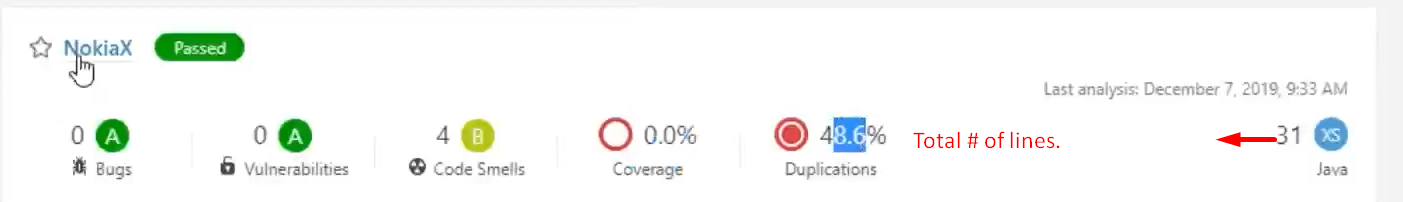
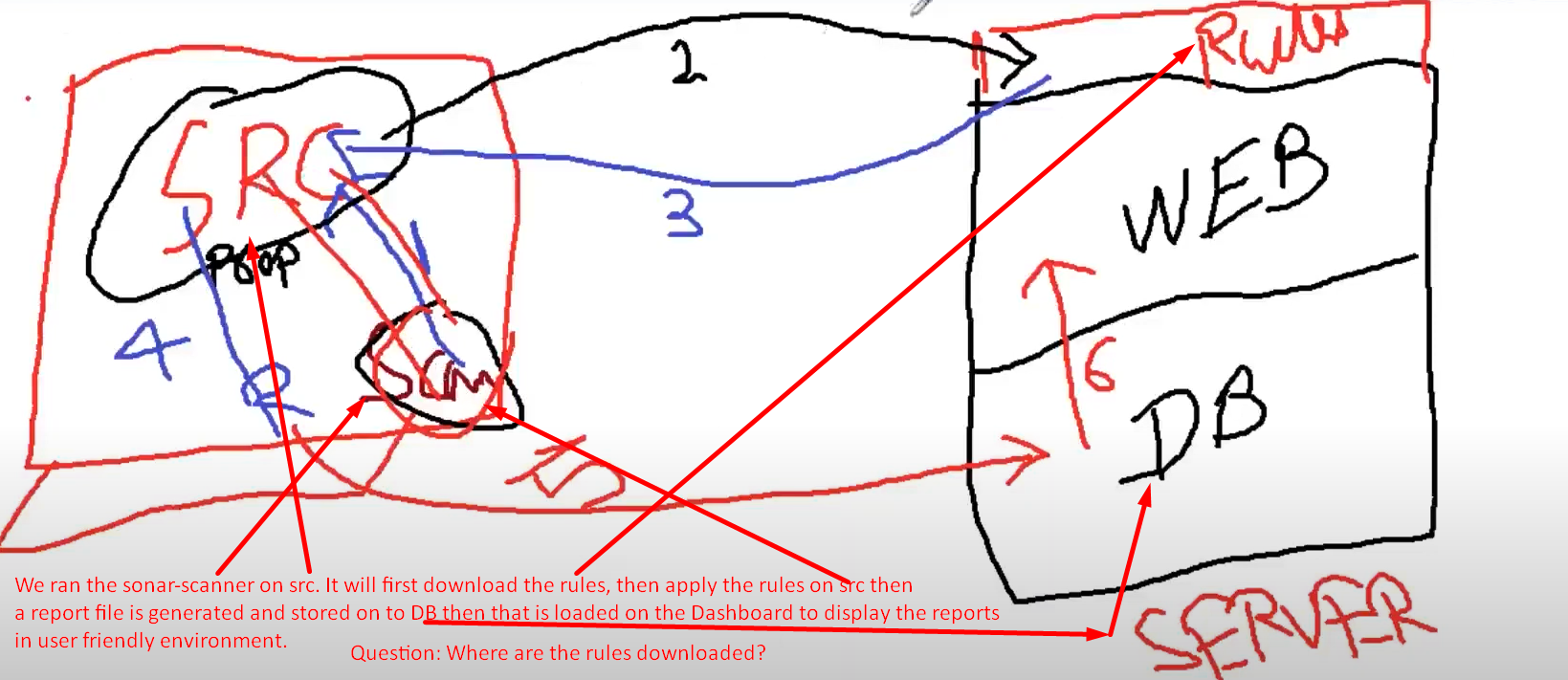
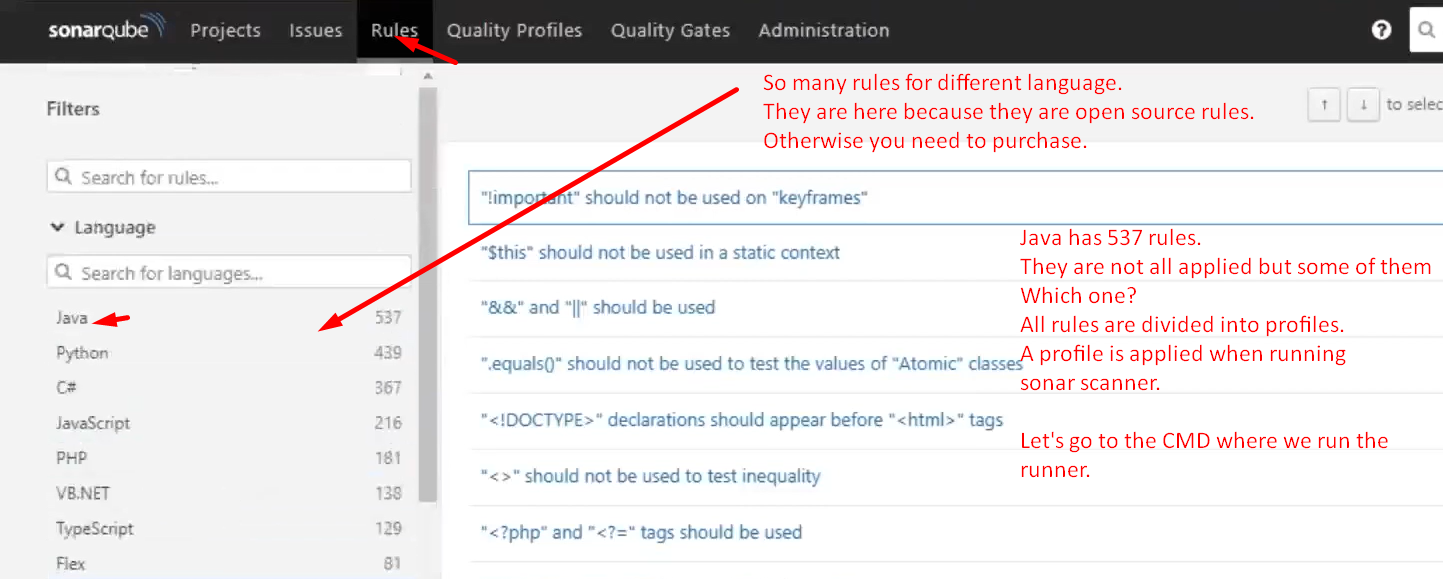
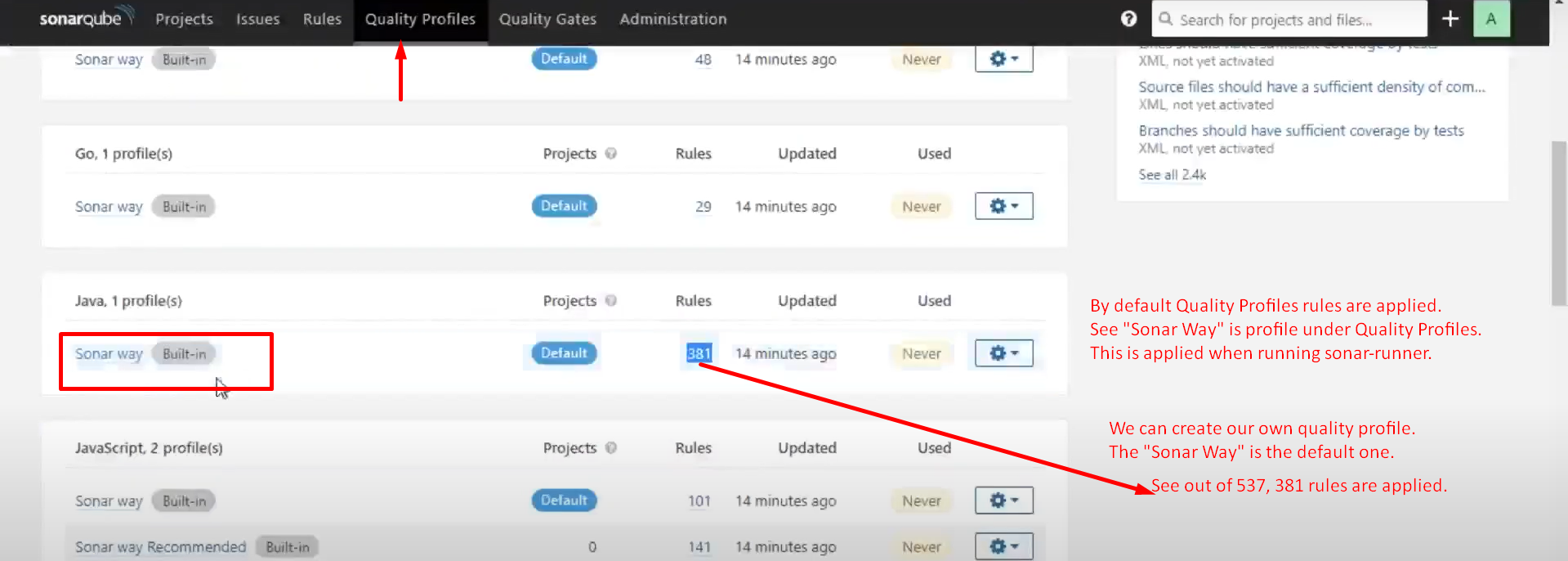
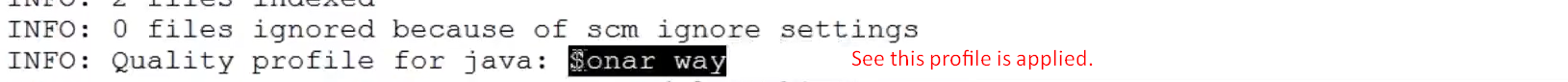
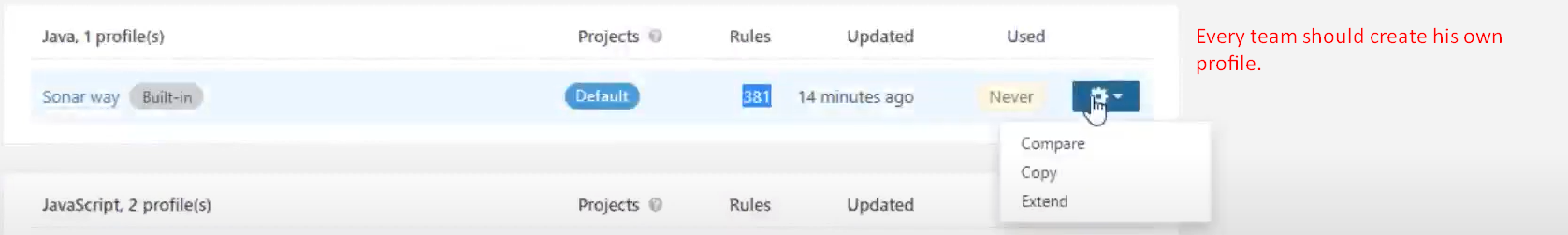
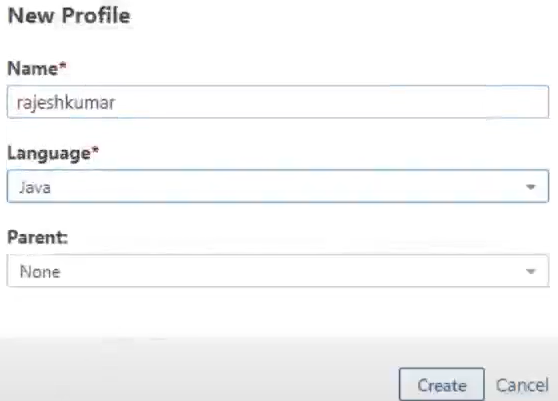
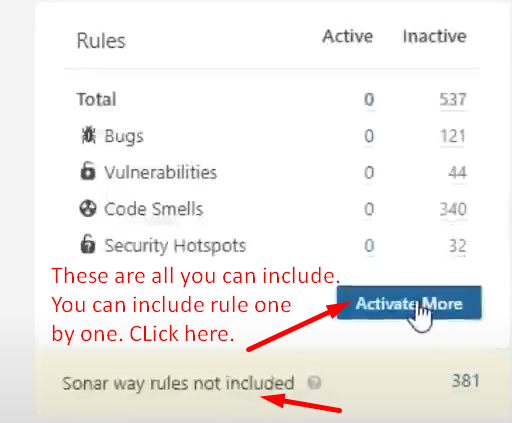
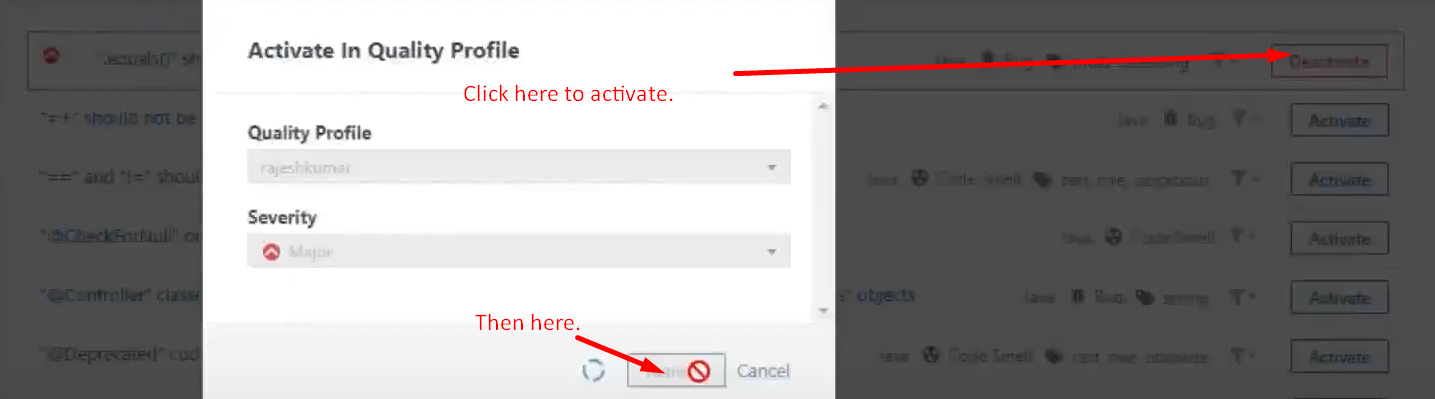
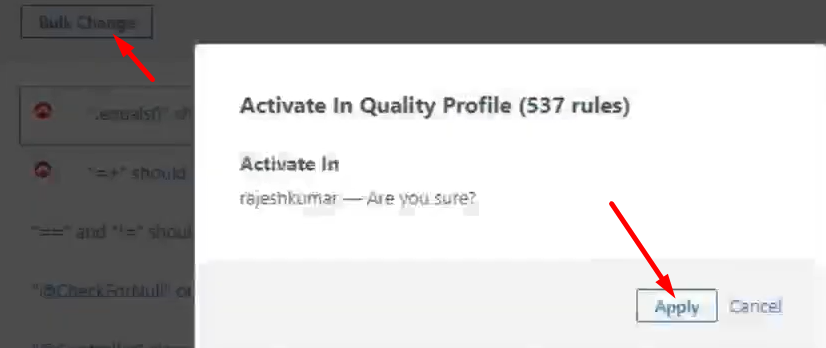
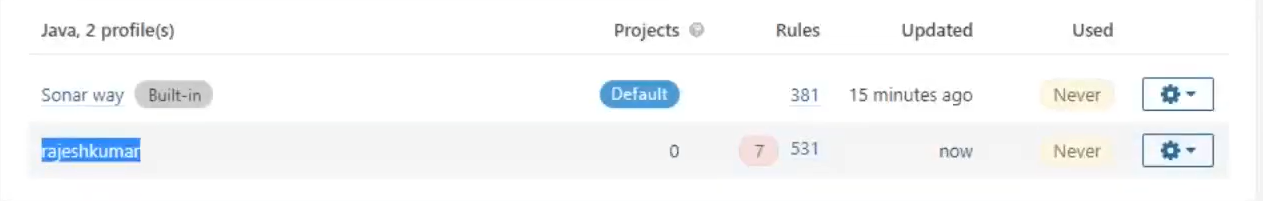
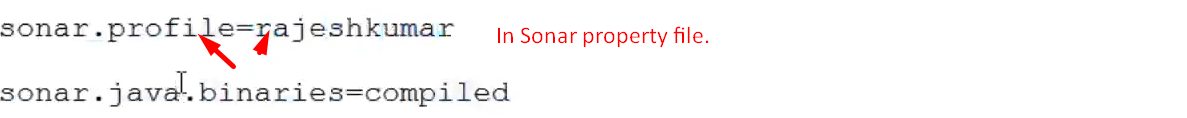
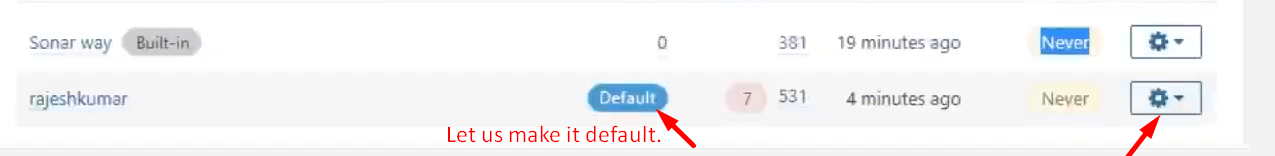
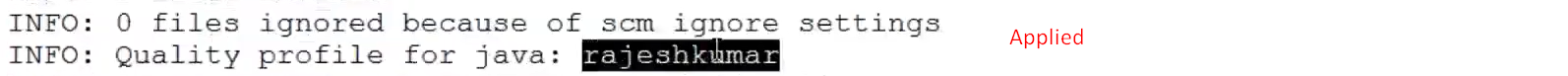
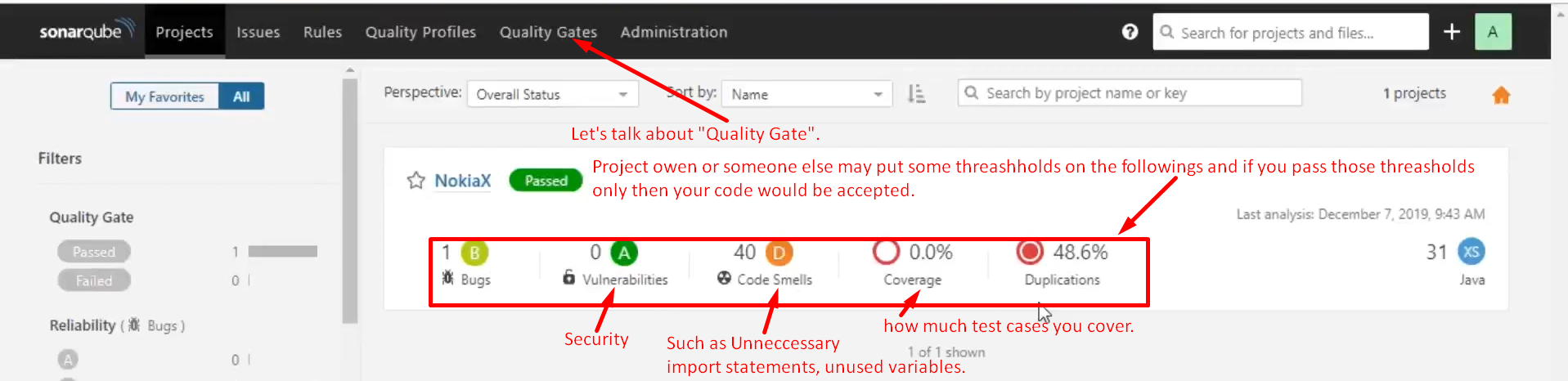
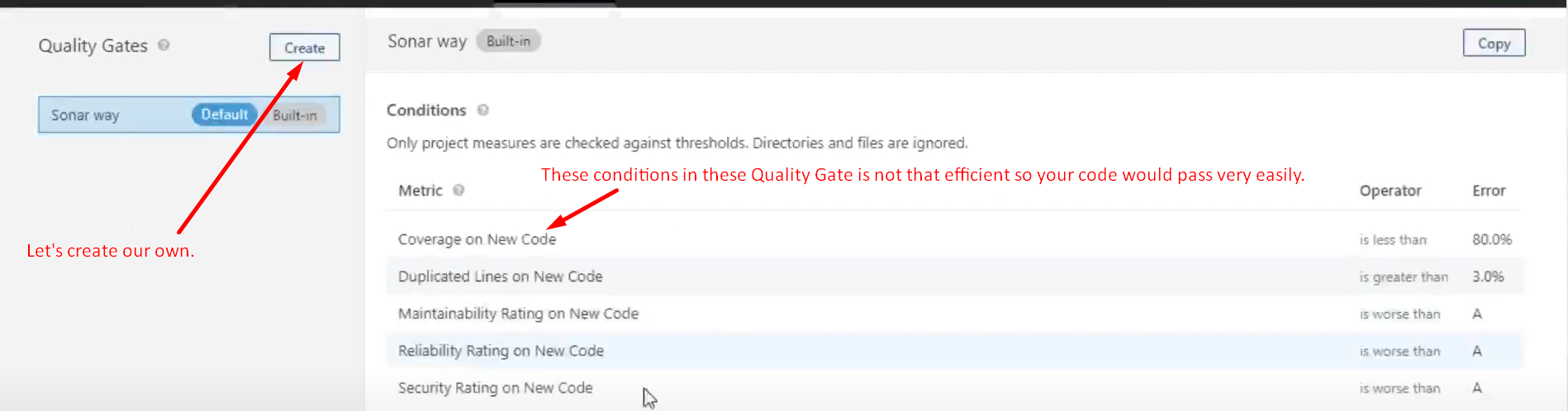
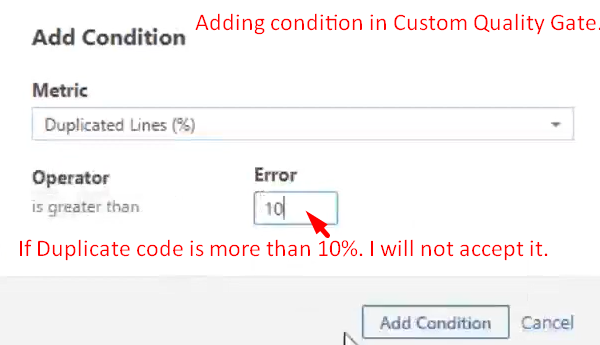
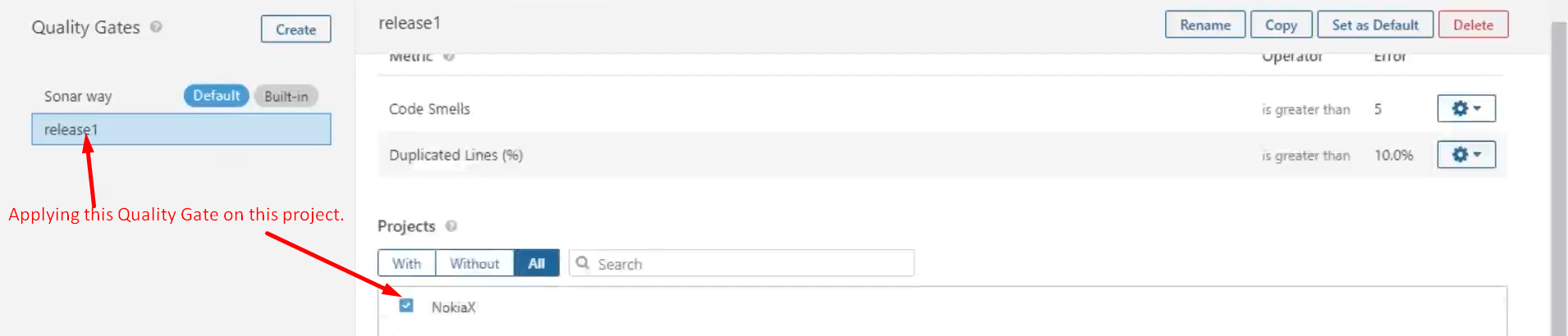
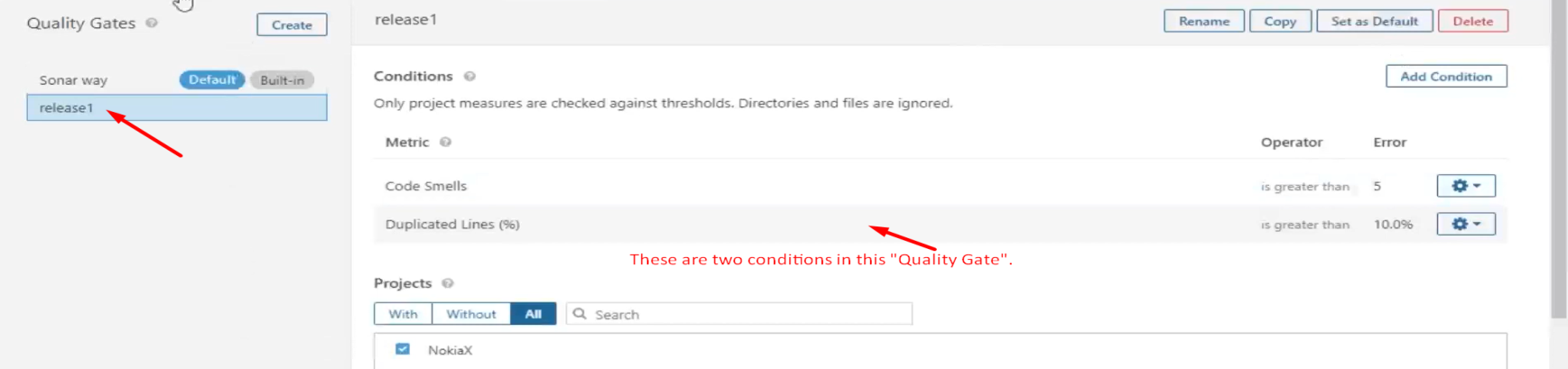
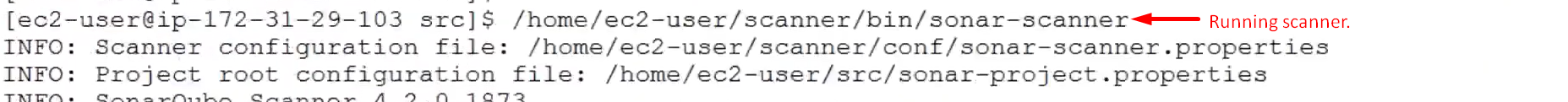
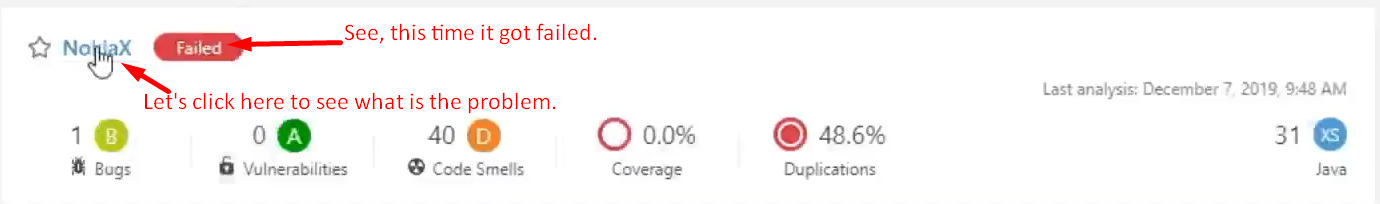
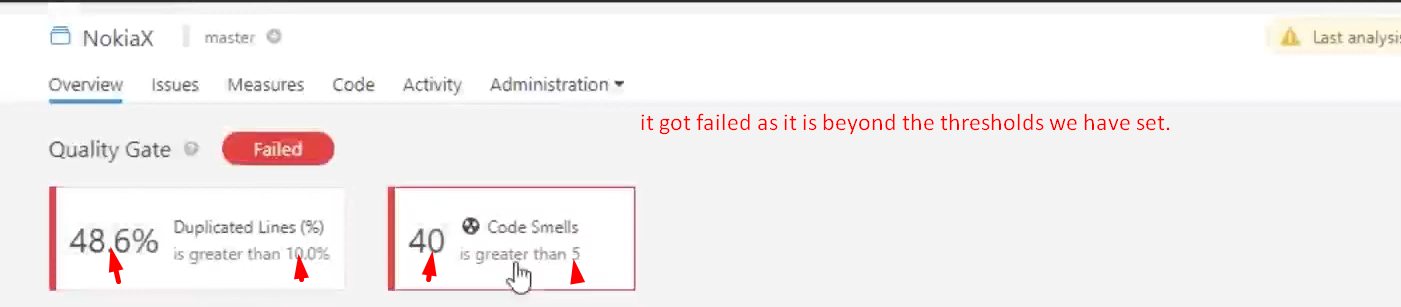
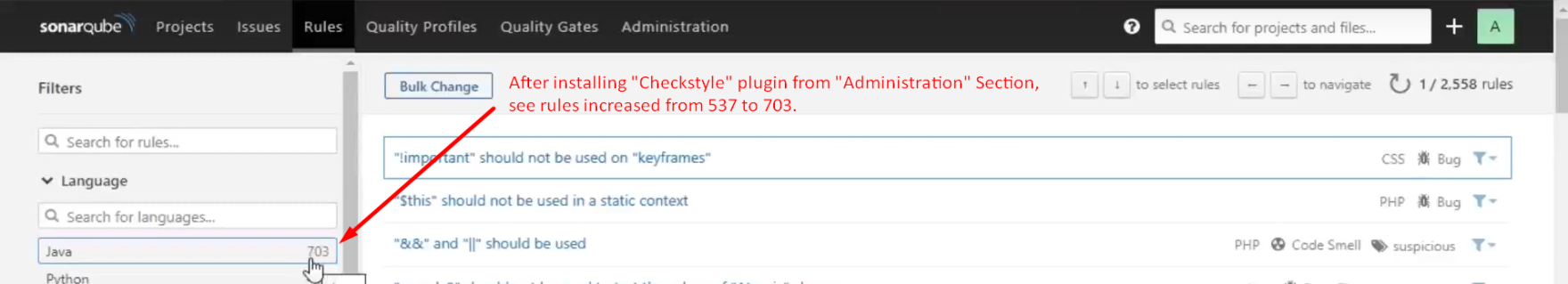
Architecture

1. 
2. SonarQube Server:
   1. Server = Dashboard + Database + Rules
   2. Database: Where all the reports and timeline are stored.
      1. MySQL, H2, Oracle, MSSQL
   3. Dashboard: Where you can see the things.
3. Scanner:
   1. Agent running in a place where you have source code.
   2. This is a jar file.
   3. You can call it from
      1. Command Line.
      2. Maven
      3. Gradle
      4. Jenkins
      5. msbuild
4. SourceCode:
   1. Code Base
5. What is the flow?
   1. First scanner is run on source code. 🡸 This is manual approach.
   2. There is a property file having all the mete-data info about your code such as
      1. The Language in which the source code is.
      2. The IP address of the SonarQube Server.
      3. What is a project key?
      4. The dir having the source code.
      5. This property file is generated manually when running from CMD or otherwise automatically by maven or Gradle.
   3. This property file is read by the scanner and reaches the Server using the IP.
   4. Then download the rules applicable which are applicable on this source code.

Rules are kinds of checklist where it reads each and every line of code and tell you what is wrong?

* 1. Then it will apply the rules.
  2. Then it will generate a report.
  3. This report is moved onto the DB on SonarQube Server.
  4. Then from DB, the report is displayed on Dashboard on Server.



1. 
2. 
3. Now we have run the “Runner” on Source Code.
4. 
5. 
6. 
7. 
8. 
9. 
10. 
11. 
12. 
13. 
14. 
15. 
16. 
17. 
18. 
19. 
20. 
21. 
22. 
23. 
24. It is not working as default. Let’s make it default from the Dashboard.  
    
25. 
26. 
27. 
28. 
29. 
30. 
31. 
32. 
33. 
34. 
35. 
36. 
37. 