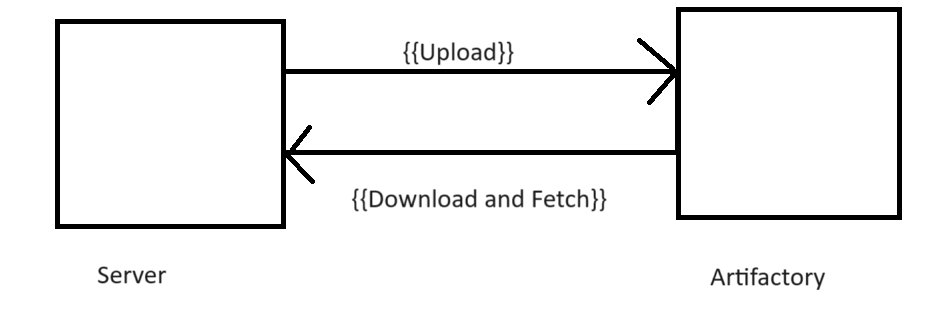
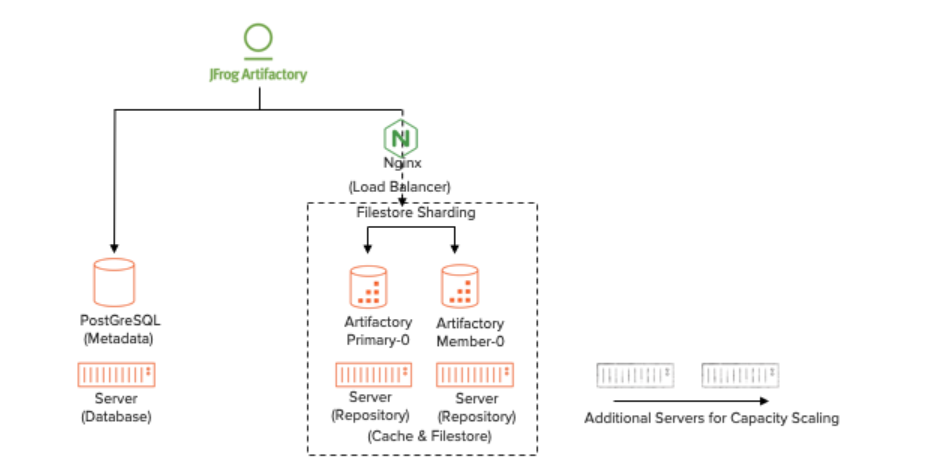
Artifactory: <https://jfrog.com/help/r/get-started-with-the-jfrog-platform/jfrog-artifactory>

Artifactory cheat sheet: <https://jfrog.com/cheat-sheet/jfrog-cli-made-easy/>

Artifactory notes: Devops Specialist\CICD\2- CICD Pipeline\Artifactory management

Artifactory architecture:

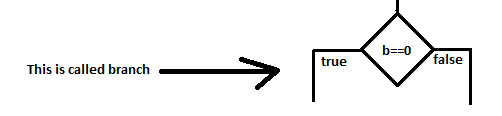




Support link: <https://blog.purestorage.com/purely-technical/jfrog-artifactory-fast-object/>

SonarQube: This tool is used to check code quality.

public int div(int a, int b) { if(b==0) { Throw DivideByZeroException("Kindly pass non zero value") } return a/b; }



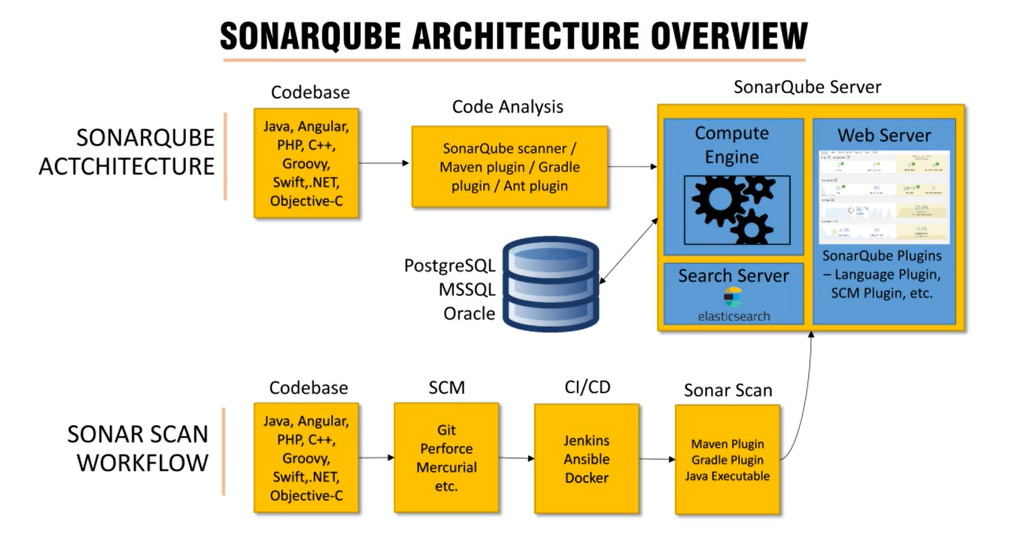
Quality gate:

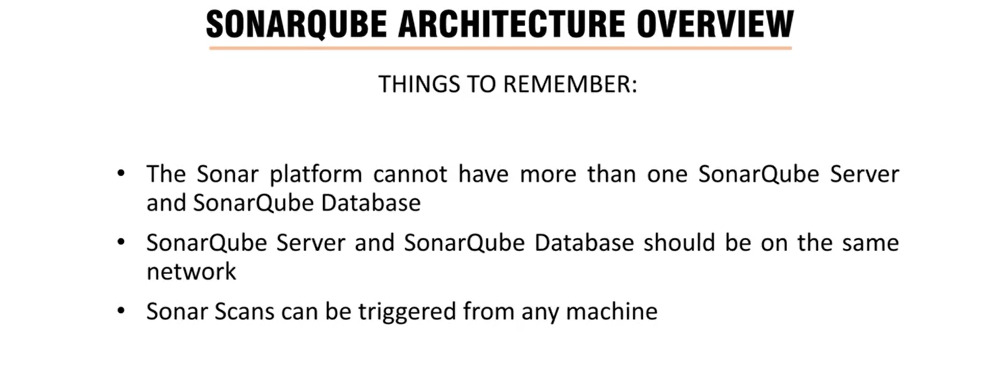
line coverage:>90 // Number of lines are executed during code execution; this should be high.

branch coverage:>90 // branch coverage should be high.

symbol coverage:>90 // are we using variable that we have defined.

\*\*\* SonarQube is configured in every build



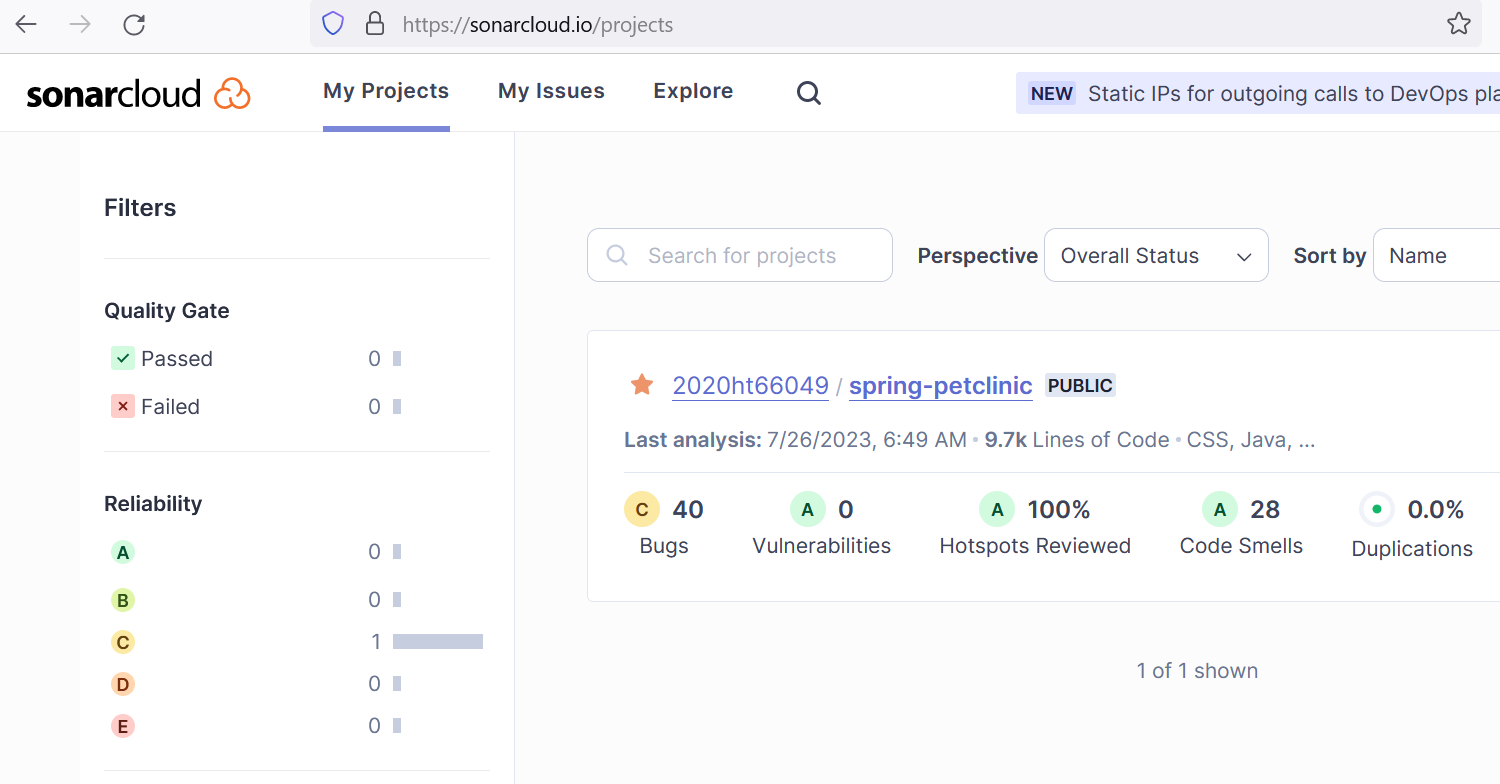


SonarQube Docker installation: <https://hub.docker.com/_/sonarqube>

SonarQube runs with port 9000

SonarCloud: SonarCloud is a cloud-based code analysis service designed to detect coding issues in 26 different programming languages. By integrating directly with your CI pipeline or one of our supported DevOps platforms, your code is checked against an extensive set of rules that cover many attributes of code, such as maintainability, reliability, and security issues on each merge/pull request. As a core element of our Sonar solution, SonarCloud completes the analysis loop to help you deliver clean code that meets high-quality standards.

[Link]: <https://docs.sonarcloud.io/>



Bug: A coding mistake that can lead to an error or unexpected behavior at runtime.

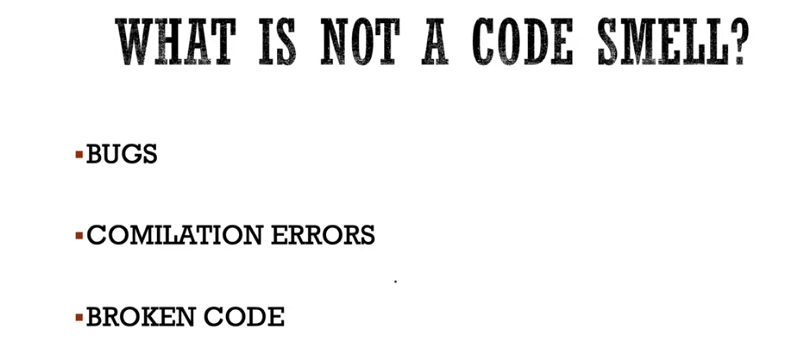
Vulnerability: A point in your code that's open to attack.

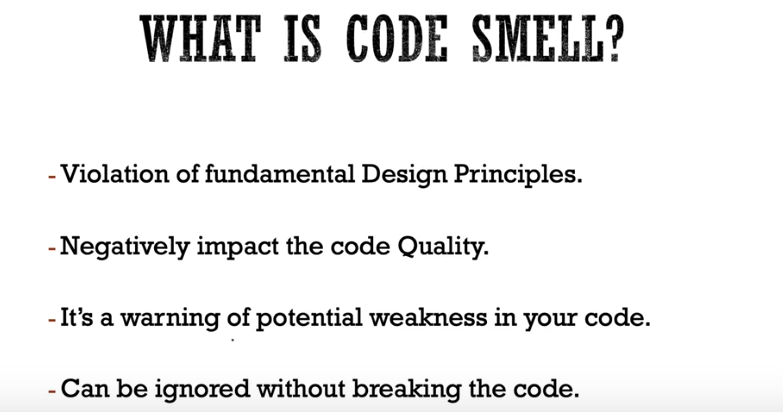


It’s like point from where attacker can attach on your network. [Example, like in home we have lock in the mail door]

Duplication: Code that looks to be duplicated.

Code Smell: A maintainability issue that makes your code confusing and difficult to maintain.





Suppose, we are allocating memory by relloc, malloc and callock but not freeing it.