**Introduction**

1. It is a Quality management tool /Code analyser– Web Based
2. Community [Free source] and Enterprise release from Sonar Source
3. Sonar Qube developed in JAVA
4. Supports close to 20 language
5. Continuous code review
6. Static code review- Automatically
7. Test Reports
8. Code coverage Reports
9. What – Why and How to fix and who created?
10. Easily integrated with Jenkins [Install plugins - sonar qube and sonar scanner]
11. Quality Gates and user can setup his own rules
12. By default port : 9000

**SonarQube Architecture**

1. SonarQube server – Dashboard and Database[reports and timeline stored]
2. Scanner – Agent running in a place where you have a source code
3. Rules/Quality Gates[if code meets the Quality gates , next state can be triggered]

**How it works?**

1. Scanner can be called from local, maven and Jenkins. Agent is a jar file
2. When scanner runs on the source code
3. In a source code is a property file[static property. Scanner reads this file first
4. Scanner downloads the Rules from server and applies on the code
5. Report is generated and moved to the database. Report is shown on the dashboard

**SonarQube Configuration [ Jenkins to SonarQube]**

1. Install **SonarQube** software[May or may not be on Jenkins machine. Only should be on the same network] [bin folder; ./sonar.sh start]
2. Install SonarQube plugin in Jenkins
3. Install **Sonar Scanner** on Jenkins machine only. This acts as an agent to the original SonarQube server From Jenkins , this scanner collects data and sends to Sonar Qube server
4. Configure Sonar Config to Jenkins
5. Set up a job to send details to Sonar Qube
6. Can be downloaded as a file or run as a container
7. SonarQube cannot be run using Root user. You should create a user to install it
8. Configure set of parameter under the job for sonar
9. Manage Jenkins🡪Configure System
10. SonarQube server details
11. Authentication token from SonarQube to Jenkins registry
12. Go to Global Tool Configuration-Give the path of Sonar runner
13. Go to Code review job🡪Configure
14. As a part of code review again you have to compile the code
15. An option: Execute Sonar Qube scanner, add build step, add sonar parameters
16. Run the build and it will send the data to SonarQube
17. Report – Quality Gate status – Bug, code smells, code coverage, duplication
18. In Quality Gate add your own conditions

**Maven – Sonar Integration**

1. Install Sonar Qube
2. Add maven- sonar plugin in pom.xml file
3. Add the test method in your Maven project
4. Make sure SonarQube is up and running
5. Build your Project : Run as 🡪Run Configuration 🡪clean install sonar:sonar – command on Goal field