DevOps Shack

TOP 100 SonarQube Questions

And Answers Asked in MNC Interviews

1. What is SonarQube?

Answer:
SonarQube is an open-source platform for continuous code quality inspection. It performs static code analysis to detect bugs, vulnerabilities, code smells, and duplications in multiple programming languages. SonarQube integrates with CI/CD pipelines, ensuring high-quality code delivery.

2. How does SonarQube work?

Answer:
SonarQube works by analyzing source code through a scanner and sending results to the SonarQube server. The workflow involves:

1. **Code Analysis: The scanner runs a static analysis on the codebase.**
2. **Report Submission: The scanner sends results to the SonarQube server.**
3. **Daboard Representation: SonarQube provides detailed insights into issues, duplications, and vulnerabilities.**
4. **Feedback Loop: Developers fix issues, and the cycle repeats.**

3. What are the key features of SonarQube?

Answer:

- **Code Quality Analysis: Identifies bugs, security vulnerabilities, and code smells.**

- **Multiple Language Support: Works with Java, Python, C#, JavaScript, etc.**
- **Continuous Integration Support: Integrates with Jenkins, Azure DevOps, GitHub Actions, etc.**
- **Code Duplication Detection: Finds redundant code segments.**
- **Security Vulnerability Detection: Helps in OWASP Top 10 & SANS security compliance.**
- **Quality Gates: Ensures code meets quality standards before deployment.**
- **Detailed Reports & Daboards: Offers actionable insights via UI and APIs.**

4. What is a SonarQube Quality Gate?

Answer:
A Quality Gate is a set of conditions that code must meet before it is considered "acceptable." It includes:

- **Minimum Code Coverage (e.g., 80% unit test coverage).**
- **No Critical or Blocker Issues (like SQL Injection or XSS).**
- **Maintainability Criteria (low technical debt).**

If the conditions fail, the build is rejected, ensuring only high-quality code is merged.

5. What are SonarQube Rules?

Answer:
SonarQube Rules define coding best practices. They are categorized as:

- **Bug Rules (Detect runtime issues)**
- **Vulnerability Rules (Detect security flaws)**
- **Code Smell Rules (Improve maintainability)**

Rules are applied through Quality Profiles, and projects must adhere to them for good quality assurance.

6. How does SonarQube integrate with Jenkins?

Answer:
SonarQube integrates with Jenkins via the SonarQube Scanner plugin:

1. **Install the SonarQube plugin in Jenkins.**
2. **Configure SonarQube Server in Jenkins settings.**
3. **Use a SonarQube token for authentication.**
4. **Define a Jenkins job and call the `sonar-scanner` in the build script.**
5. **Run the pipeline, and the analysis is sent to SonarQube.**

7. What is a SonarQube Quality Profile?

Answer:
A Quality Profile is a set of rules that define code quality standards for a project. SonarQube provides default profiles, but users can customize profiles based on:

- **Project type**
- **Compliance needs (e.g., OWASP security rules)**
- **Development standards**

8. What is Technical Debt in SonarQube?

Answer:
Technical Debt refers to the amount of effort required to fix code quality issues. SonarQube assigns a debt ratio based on:

- Code smells
- Duplications
- Complex methods
- Poorly structured code

A high technical debt means future maintenance will be expensive.

9. What is the difference between Code Smell, Bug, and Vulnerability in SonarQube?

Answer:

- Bug: A coding error that can cause incorrect behavior at runtime.
- Vulnerability: A security flaw that can be exploited (e.g., SQL Injection).
- Code Smell: Poor coding practice that increases maintenance complexity.

10. How do you run a SonarQube analysis?

Answer:
SonarQube analysis can be run using:

SonarQube Scanner CLI:

```
sonar-scanner -Dsonar.projectKey=my_project
-Dsonar.host.url=http://localhost:9000
-Dsonar.login=my_token
```

Maven Plugin:

```
mvn clean verify sonar:sonar
```

Gradle Plugin:

```
gradle sonarqube
```

11. What are SonarQube Metrics?

Answer:
SonarQube provides various metrics:

- **Reliability: Bugs count.**
- **Security: Vulnerabilities detected.**
- **Maintainability: Code smells and technical debt.**
- **Coverage: Test coverage percentage.**
- **Duplications: Percentage of duplicated code.**

12. How do you enable SonarQube in a CI/CD pipeline?

Answer:

1. **Install the SonarQube Scanner.**
2. **Add SonarQube authentication token in CI/CD.**
3. **Configure SonarQube analysis in Jenkins/Azure DevOps/GitHub Actions.**
4. **Run the pipeline, and SonarQube will analyze the code.**

13. How does SonarQube handle security vulnerabilities?

Answer:
SonarQube detects security issues like:

- **SQL Injection**

- **Cross-Site Scripting (XSS)**
- **Hardcoded credentials**
- **Insecure APIs It follows OWASP & SANS security best practices to flag vulnerabilities.**

14. What is SonarLint, and how is it different from SonarQube?

Answer:

- **SonarLint is a local IDE extension that gives real-time code analysis.**
- **SonarQube is a server-based tool that performs deeper analysis.**

SonarLint helps developers fix issues before committing the code.

15. What databases does SonarQube support?

Answer:
SonarQube supports:

- **PostgreSQL (Recommended)**
- **MySQL (deprecated in newer versions)**
- **Oracle**
- **Microsoft SQL Server**

16. What is the difference between SonarQube Community, Developer, Enterprise, and Data Center Editions?

Answer:
SonarQube offers different editions based on features and scale:

- **Community Edition (Free): Supports basic code analysis for open-source and small teams.**

- **Developer Edition: Adds branch analysis and deeper security insights.**
- **Enterprise Edition: Includes portfolio management, security reports, and compliance features.**
- **Data Center Edition: High availability, multi-node clustering for large-scale organizations.**

For enterprise-level projects, the Enterprise Edition is recommended.

17. How do you secure SonarQube?

Answer:
To secure SonarQube, follow these best practices:

- **Enable HTTPS for secure communication.**
- **Use authentication & authorization (LDAP, SAML, GitHub OAuth).**
- **Restrict API access with tokens instead of credentials.**
- **Use database encryption to protect sensitive data.**
- **Regularly update SonarQube to patch security vulnerabilities.**

18. What are the default SonarQube Quality Gates?

Answer:
The default SonarQube Quality Gate includes:

- **No blocker or critical issues (bugs & vulnerabilities).**
- **Code coverage ≥ 80%.**
- **Duplicated code ≤ 3%.**
- **Maintainability: No new technical debt.**

Quality Gates ensure only high-quality code is deployed.

19. How do you create a custom Quality Gate in SonarQube?

Answer:

1. **Go to Administration → Quality Gates.**
2. **Click Create and define conditions.**
3. **Set threolds for bugs, security issues, code coverage, duplications, and maintainability.**
4. **Assign the Quality Gate to projects.**

Custom Quality Gates allow you to enforce project-specific coding standards.

20. What is SonarScanner, and how does it work?

Answer:
SonarScanner is the command-line tool that sends code analysis reports to the SonarQube server.

Steps:

Install SonarScanner:

```
sudo apt install sonar-scanner
```

Run analysis:

```
sonar-scanner                    -Dsonar.projectKey=my_project
-Dsonar.host.url=http://localhost:9000
-Dsonar.login=my_token
```

View results on the SonarQube daboard.

It integrates with Jenkins, GitHub Actions, Azure DevOps, etc.

21. How do you analyze multiple branches in SonarQube?

Answer:
For multi-branch analysis, you need Developer or Enterprise Edition.

1. **Enable branch support in SonarQube.**

Use the `sonar.branch.name` parameter:

```
sonar-scanner -Dsonar.branch.name=feature-branch
```

2. **View results for each branch in the SonarQube daboard.**

This helps in tracking quality per branch.

22. How do you integrate SonarQube with GitHub?

Answer:
To integrate SonarQube with GitHub:

1. **Generate a GitHub token with repo and admin access.**
2. **Configure SonarQube Pull Request Decoration in Administration → ALM Integrations.**

Use `sonar.pullrequest.key` in CI/CD:
carp

```
sonar-scanner -Dsonar.pullrequest.key=123
-Dsonar.pullrequest.branch=feature-branch
-Dsonar.pullrequest.base=main
```

3. **SonarQube will comment directly on GitHub PRs.**

23. How do you enforce code coverage in SonarQube?

Answer:

1. **Ensure tests generate coverage reports (JaCoCo, Istanbul, Cobertura).**
2. **Use the `sonar.coverage.exclusions` property for unwanted files.**

Run the scanner with coverage parameters:

```
sonar-scanner -Dsonar.coverage.reportPaths=coverage.
```

3. **Define a Quality Gate rule (e.g., Minimum 80% coverage).**

24. What are SonarQube Hotspots?

Answer:
Hotspots indicate potential security issues that require manual review, such as:

- **Unvalidated user inputs.**
- **Hardcoded credentials.**
- **Weak cryptographic algorithms.**

Unlike vulnerabilities, hotspots are not guaranteed risks, but ould be reviewed.

25. How do you configure SonarQube for Java projects using Maven?

Answer:

Add SonarQube plugin in `pom.`:

```
<plugin>
```

```
  <groupId>org.sonarsource.scanner.maven</groupId>
  <artifactId>sonar-maven-plugin</artifactId>
  <version>3.9.1.2184</version>
</plugin>
```

Run:

```
mvn clean verify sonar:sonar -Dsonar.projectKey=my_project
```

1. **Results appear in the SonarQube daboard.**

26. What are SonarQube Web APIs used for?

Answer:
SonarQube provides REST APIs to automate tasks such as:

- **Fetching reports:** `/api/issues/search`
- **Triggering scans:** `/api/ce/submit`
- **Managing users:** `/api/users/create`

Example API call:

```
curl -u admin:admin 'http://localhost:9000/api/projects/search'
```

27. How do you configure SonarQube for Python projects?

Answer:

Install `pylint` for static analysis:

```
pip install pylint
```

Generate reports:
lua

```
pylint --output-format=json my_project > pylint-report.json
```

Run SonarQube analysis:

```
sonar-scanner
-Dsonar.python.pylint.reportPaths=pylint-report.json
```

28. How do you analyze code using Dockerized SonarQube?

Answer:

Run SonarQube in Docker:

```
docker run -d --name sonarqube -p 9000:9000 sonarqube:lts
```

Install SonarScanner in the container:

```
docker exec -it sonarqube /bin/ba
```

1. Run analysis inside the container.

29. How do you disable a specific rule in SonarQube?

Answer:

1. **Go to Quality Profiles in SonarQube.**
2. **Search for the rule to disable.**
3. **Click Deactivate or create a custom profile.**

Alternatively, exclude a rule in `sonar-project.properties`:

```
sonar.issue.ignore.multicriteria=rule1,rule2
sonar.issue.ignore.multicriteria.rule1.ruleKey=squid:S001
```

30. What is SonarQube's Leak Period?

Answer:
The Leak Period defines the time range for monitoring new issues.

- **If set to `"previous_version"`, SonarQube compares changes against the last analyzed version.**
- **Helps track recent regressions in code quality.**

31. How do you configure SonarQube with Kubernetes?

Answer:

Deploy SonarQube via Helm:

```
helm install sonarqube sonarqube/sonarqube
```

Expose via Ingress:

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: sonarqube
spec:
  rules:
    - host: sonarqube.example.com
      http:
        paths:
          - path: /
            pathType: Prefix
            backend:
              service:
                name: sonarqube
                port:
                  number: 9000
```

This deploys SonarQube in a scalable Kubernetes environment.

32. How do you configure SonarQube to analyze JavaScript and TypeScript projects?

Answer:

Install the SonarScanner CLI and `eslint`:

```
npm install -g eslint
```

Generate an ESLint report:

```
eslint -f json -o eslint-report.json src/
```

Run SonarScanner with ESLint integration:

```
sonar-scanner
-Dsonar.eslint.reportPaths=eslint-report.json
```

1. **View the analysis results on the SonarQube daboard.**

33. How do you exclude specific files from SonarQube analysis?

Answer:
To exclude files or directories, add them to `sonar-project.properties`:

properties

```
sonar.exclusions=**/node_modules/**,          **/test/**,
**/*.spec.js
```

Alternatively, use the SonarQube UI:

1. **Navigate to Project Settings → Exclusions.**
2. **Add file patterns for exclusion.**

34. How do you enable LDAP authentication in SonarQube?

Answer:

Edit `sonar.properties` to include LDAP configuration:
properties

```
sonar.security.realm=LDAP
```

```
ldap.url=ldap://ldap.example.com:389
ldap.bindDn=cn=admin,dc=example,dc=com
ldap.bindPassword=adminpassword
```

Restart SonarQube:

```
systemctl restart sonarqube
```

This enables centralized authentication via LDAP.

35. What is SonarQube's Code Coverage, and how is it measured?

Answer:
Code coverage in SonarQube measures the percentage of code executed by unit tests. It consists of:

- **Line coverage: % of lines executed by tests.**
- **Branch coverage: % of decision points (if-else, loops) tested.**
- **Condition coverage: Tests Boolean conditions separately.**

Tools like JaCoCo, Cobertura, Istanbul generate reports that SonarQube reads.

Example for Maven + JaCoCo:

```
mvn clean test jacoco:report sonar:sonar
```

36. How do you set up SonarQube with GitLab CI/CD?

Answer:

Add SonarQube service in `gitlab-ci.yml`:

```
sonar_scan:
  image: sonarsource/sonar-scanner-cli
  script:
      - sonar-scanner -Dsonar.host.url=$SONAR_HOST_URL
-Dsonar.login=$SONAR_TOKEN
```

1. Add SONAR_HOST_URL and SONAR_TOKEN in GitLab CI/CD Variables.
2. Run the pipeline to trigger code analysis.

37. What is SonarCloud, and how does it differ from SonarQube?

Answer:

- SonarQube is a self-hosted code analysis tool.
- SonarCloud is a cloud-based service managed by SonarSource.
- SonarCloud supports GitHub, GitLab, Bitbucket & Azure DevOps without local server setup.
- SonarCloud offers automatic updates, unlike SonarQube, which requires manual updates.

38. How do you increase performance in SonarQube for large projects?

Answer:
To improve SonarQube performance:

Increase JVM heap memory in `sonar.properties`:
properties

```
sonar.web.javaOpts=-Xmx4G -Xms2G
-XX:+HeapDumpOnOutOfMemoryError
```

- Use PostgreSQL (faster than MySQL).
- Enable database indexing to speed up queries.
- Use background task parallelism.
- Run incremental analysis instead of full scans.

39. What are SonarQube's default permission levels?

Answer:
SonarQube has default user roles:

- Admin: Full control over SonarQube.
- Project Admin: Manages a specific project.
- User: Can view reports and run analyses.
- Code Viewer: Read-only access to reports.

Permissions are configurable under Administration → Security → Roles.

40. How do you integrate SonarQube with Azure DevOps?

Answer:

1. Install the SonarQube extension in Azure DevOps Marketplace.
2. Add a SonarQube service connection in Azure Pipelines.

Modify `azure-pipelines.yml`:

```
steps:
  - task: SonarQubePrepare@5
    Inputs:
```

```
    SonarQube: 'SonarQube-Service'
    scannerMode: 'CLI'
- task: SonarQubeAnalyze@5
- task: SonarQubePubli@5
```

3. Run the pipeline; results appear in SonarQube & Azure DevOps daboards.

41. How do you configure SonarQube to ignore test files?

Answer:
Add exclusions to `sonar-project.properties`:

properties

```
sonar.test.exclusions=**/tests/**, **/*.spec.js
```

This prevents unit test files from being analyzed as part of main source code.

42. How do you reset an admin password in SonarQube?

Answer:

Run SQL command to reset the admin password:

```
UPDATE users SET crypted_password=NULL, salt=NULL WHERE
login='admin';
```

1. Restart SonarQube and log in with:
    - Username: `admin`
    - Password: `admin`
2. Set a new password immediately.

43. How do you migrate SonarQube to a new server?

Answer:

1. **Back up the database using `pg_dump` or `mysqldump`.**
2. **Install SonarQube on the new server.**

Restore the database:

```
pg_restore -d sonar -U sonar_user sonar_backup.sql
```

3. **Update `sonar.properties` with new database credentials.**
4. **Restart SonarQube.**

44. How do you troubleoot "SonarQube Server is Unavailable" error?

Answer:

- **Check logs: `logs/sonar.log`, `logs/es.log`.**
- **Verify database connectivity.**
- **Ensure ports 9000 and 5432 are open.**
- **Increase Java memory if logs ow OutOfMemoryError.**

45. What is the role of Elasticsearch in SonarQube?

Answer:
Elasticsearch indexes and searches data in SonarQube for fast retrieval.

- **Stores analysis results for quick access.**
- **Powers issue tracking and reporting daboards.**
- **Must be restarted if corrupt (`rm -rf data/es6`).**

46. How do you delete a project in SonarQube?

Answer:

1. **Go to Administration → Projects.**
2. **Select the project and click Delete.**

Alternatively, use API:

```
curl -u admin:admin -X POST "http://localhost:9000/api/projects/delete?project=my_project"
```

47. How do you create a backup of SonarQube analysis results?

Answer:

Backup PostgreSQL database:

```
pg_dump -U sonar -h localhost -d sonarqube -F c -b -v -f sonar_backup.sql
```

Backup the SonarQube config and data:

```
tar -czvf sonar_backup.tar.gz /opt/sonarqube
```

1. **Store backups in cloud storage or external drives.**

48. What is the default port for SonarQube?

Answer:

- SonarQube UI: `9000`
- Elasticsearch: `9001`

Configurable in `sonar.properties`:
properties

```
sonar.web.port=8080
```

49. How do you enforce OWASP security standards in SonarQube?

Answer:

1. Use SonarQube Security Hotspots.
2. Enable OWASP Top 10 rules in Quality Profiles.
3. Integrate SAST tools like Checkmarx.

50. How do you configure SonarQube to analyze C/C++ code?

Answer:
For C/C++ analysis, you need SonarQube Developer or Enterprise Edition and SonarScanner for C/C++ (Build Wrapper).

1. Download and install SonarQube and SonarScanner for C/C++.

Run the Build Wrapper to capture the compilation process:

```
build-wrapper-linux-x86-64 --out-dir bw-output make clean all
```

Run SonarScanner with the captured build data:

```
sonar-scanner
-Dsonar.cfamily.build-wrapper-output=bw-output
```

2. View analysis results in the SonarQube daboard.

51. How does SonarQube handle secrets detection (e.g., API keys, passwords)?

Answer:
SonarQube detects hardcoded secrets like API keys, passwords, and credentials through security rules in Quality Profiles.

1. Enable security rules related to secret detection.
2. Use external secret scanners (e.g., TruffleHog, GitLeaks).

Exclude sensitive files in `sonar-project.properties`:
properties

```
sonar.exclusions=**/config/secrets/**, **/*.env
```

52. How do you analyze Kotlin projects in SonarQube?

Answer:
For Kotlin projects, SonarQube supports static analysis and test coverage.

1. Install the SonarQube Kotlin Plugin.

Run analysis using Gradle:

```
./gradlew                                    sonarqube
-Dsonar.host.url=http://localhost:9000
-Dsonar.login=my_token
```

Use JaCoCo for test coverage:

```
./gradlew jacocoTestReport
```

53. How do you enforce coding standards using SonarQube?

Answer:

1. **Define Quality Profiles with project-specific coding standards.**
2. **Apply Quality Gates to enforce compliance.**
3. **Run SonarLint in IDEs for real-time feedback.**
4. **Reject Pull Requests if code does not meet standards using GitHub/GitLab Integration.**

54. How do you integrate SonarQube with Bitbucket Pipelines?

Answer:

1. **Create a SonarQube token in Bitbucket Repository Variables.**

Modify `bitbucket-pipelines.yml`:

```
pipelines:
  default:
    - step:
        script:
          - pipe: sonarsource/sonarqube-scan:1.0.0
            variables:
                                          SONAR_HOST_URL:
'https://sonarqube.example.com'
              SONAR_TOKEN: $SONAR_TOKEN
```

55. How do you configure SonarQube for Android projects?

Answer:

Add SonarQube Plugin to `build.gradle`:

```
plugins {
    id "org.sonarqube" version "3.3"
}
```

Run analysis:

```
./gradlew sonarqube
```

56. What is SonarQube's Code Duplication Metric?

Answer:

- **Measures duplicate lines in code.**
- **Helps maintain clean, modular code.**
- **Defined as a percentage:**
  - **≤3% duplication is acceptable.**
  - **≥10% duplication indicates poor maintainability.**
- **Reduce duplication using refactoring and modularization.**

57. How do you configure SonarQube logging for debugging?

Answer:

Set logging level in `sonar.properties`:

Properties

```
sonar.log.level=DEBUG
```

1. View logs in:
    - `logs/sonar.log` (SonarQube server logs)
    - `logs/es.log` (Elasticsearch logs)
    - `logs/ce.log` (Compute Engine logs)

58. How do you monitor SonarQube performance?

Answer:

Use SonarQube Web API:

```
curl -u admin:admin "http://localhost:9000/api/system/health"
```

- Integrate Prometheus & Grafana for monitoring.
- Check database response time & JVM memory usage.

59. How do you enforce SonarQube rules in IDEs like IntelliJ, VSCode, Eclipse?

Answer:

1. Install SonarLint Plugin in the IDE.
2. Connect SonarLint to SonarQube for synchronized rules.
3. Enable real-time feedback to catch issues before committing code.

60. How do you integrate SonarQube with Terraform?

Answer:

1. Use `tflint` for Terraform static analysis.

2. Convert TFLint reports into SonarQube-compatible format.
3. Run `sonar-scanner` with Terraform reports.

61. How do you troubleoot false positives in SonarQube?

Answer:

1. Mark them as Won't Fix or False Positive in the UI.
2. Adjust rule threolds in Quality Profiles.
3. Use sonar.issue.ignore.multicriteria to ignore specific cases.

62. How do you generate SonarQube reports automatically?

Answer:
Use the SonarQube API:

```
curl -u admin:admin "http://localhost:9000/api/issues/search?componentKeys=my_project"
```

Or use SonarQube PDF Report Plugin.

63. What is the purpose of SonarQube's Portfolio feature?

Answer:

- Available in Enterprise Edition.
- Aggregates multiple projects into a single daboard.
- Helps track company-wide code quality trends.

64. How do you fix a SonarQube Elasticsearch corruption error?

Answer:

Stop SonarQube:

```
systemctl stop sonarqube
```

Delete Elasticsearch data:

```
rm -rf /opt/sonarqube/data/es6
```

Restart SonarQube:

```
systemctl start sonarqube
```

65. How do you enable SonarQube Dark Mode?

Answer:

1. **Install SonarQube Theme Plugin.**
2. **Select Dark Mode in profile settings.**

66. How do you check SonarQube API health?

Answer:
Run:

```
curl -u admin:admin "http://localhost:9000/api/system/status"
```

- `UP` = Running
- `DOWN` = Issue detected

67. How do you enforce secure coding standards using SonarQube?

Answer:

- **Use OWASP, SANS, PCI DSS security rules.**
- **Enable SonarQube Security Hotspots.**
- **Integrate SAST tools like Checkmarx.**

68. How do you rollback a SonarQube upgrade?

Answer:

1. **Restore database backup.**
2. **Reinstall previous SonarQube version.**
3. **Restore `sonar.properties` from backup.**

69. How do you configure SonarQube in a CI/CD pipeline for microservices?

Answer:

- **Run separate SonarQube analysis for each microservice.**

Define `sonar.projectKey` dynamically:

```
sonar-scanner -Dsonar.projectKey=$MICROSERVICE_NAME
```

70. How do you fix SonarQube's "Database migration failed" error?

Answer:

1. **Check logs for database migration errors.**
2. **Manually apply SQL schema updates.**

3. **Restart SonarQube.**

71. How do you automate SonarQube scans for every Git commit?

Answer:
You can automate SonarQube scans for every Git commit using CI/CD tools like Jenkins, GitHub Actions, or GitLab CI/CD.

Example: GitHub Actions Integration

Create a `.github/workflows/sonarqube.yml` file:

```
name: SonarQube Analysis
on:
  pu:
    branches:
      - main
jobs:
  sonarqube:
    runs-on: ubuntu-latest
    steps:
      - name: Checkout code
        uses: actions/checkout@v3
      - name: SonarQube Scan
        uses: sonarsource/sonarqube-scan-action@master
        env:
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

```
SONAR_HOST_URL: "https://sonarqube.example.com"
```

1. Add SONAR_TOKEN as a GitHub secret.
2. Every commit to main triggers SonarQube analysis.

72. How do you ignore specific rules in SonarQube?

Answer:

Ignore a rule globally in `sonar-project.properties`:
properties

```
sonar.issue.ignore.multicriteria=rule1
sonar.issue.ignore.multicriteria.rule1.ruleKey=squid:S001
```

Ignore a rule in a specific file or directory:
properties

```
sonar.exclusions=**/legacy_code/**
```

Suppress a rule in the source code using annotations:
java

```
@SuppressWarnings("squid:S001")
```

73. How do you integrate SonarQube with Jenkins pipelines?

Answer:

1. **Install the SonarQube Scanner Plugin in Jenkins.**
2. **Configure SonarQube Server in Jenkins settings.**

Add SonarQube analysis in `Jenkinsfile`:

```
pipeline {
    agent any
    stages {
        stage('SonarQube Analysis') {
            steps {
                script {
                    withSonarQubeEnv('SonarQube') {
                        'mvn clean verify sonar:sonar'
                    }
                }
            }
        }
    }
}
```

3. **Run the pipeline and view analysis in SonarQube.**

74. How do you export SonarQube reports as PDF?

Answer:

- Install SonarQube PDF Report Plugin.
- Navigate to Project Daboard → More Actions → Generate PDF Report.

Use SonarQube API:

```
curl -u admin:admin "http://localhost:9000/api/report/export?format=pdf"
```

75. How do you analyze a monorepo with multiple projects in SonarQube?

Answer:
For a monorepo, define multiple projects using `sonar-project.properties`:

properties

```
sonar.projectKey=repo_project1
sonar.sources=src/project1
```

Run analysis separately for each sub-project.

Alternatively, use SonarQube Enterprise Edition's Portfolio feature.

76. What is the SonarQube Scanner for MSBuild?

Answer:
SonarScanner for MSBuild is used for .NET and C# projects.
Steps to integrate:

1. Download SonarScanner for MSBuild.

Run the analysis:

```
SonarScanner.MSBuild.exe begin /k:"my_project"
/d:sonar.host.url="http://localhost:9000"
MSBuild.exe /t:Rebuild
SonarScanner.MSBuild.exe end
```

77. How do you analyze Swift code with SonarQube?

Answer:

Install SwiftLint:

```
brew install swiftlint
```

Generate a SwiftLint report:

```
swiftlint --reporter json > swiftlint-report.json
```

Run SonarScanner:

```
sonar-scanner
-Dsonar.swift.swiftlint.reportPaths=swiftlint-report.js
on
```

78. How do you secure SonarQube with HTTPS?

Answer:

Enable HTTPS in `sonar.properties`:
properties

```
sonar.web.https.enabled=true
sonar.web.https.port=9001
sonar.web.https.keyStore=/path/to/keystore.jks
sonar.web.https.keyStorePassword=yourpassword
```

1. **Restart SonarQube.**

79. How do you automate SonarQube user management?

Answer:
Use the SonarQube API to create users:

```
curl -u admin:admin -X POST "http://localhost:9000/api/users/create?login=user1&password=password123"
```

Alternatively, integrate with LDAP or SAML authentication.

80. How do you configure SonarQube to run only on modified files?

Answer:

Use Git diff to find modified files:

```
git diff --name-only origin/main...HEAD > changed_files.txt
```

Pass modified files to SonarScanner:

properties

```
sonar.inclusions=@changed_files.txt
```

81. What is the SonarQube Quality Gate API?

Answer:
The Quality Gate API helps automate checks:

```
curl -u admin:admin "http://localhost:9000/api/qualitygates/project_status?projectKey=my_project"
```

Returns PASS or FAIL based on set conditions.

82. How do you configure SonarQube for Vue.js projects?

Answer:

Install ESLint & Vue.js Plugin:

```
npm install eslint-plugin-vue --save-dev
```

Generate ESLint report:

```
eslint --format json --output-file eslint-report.json src/
```

Run SonarScanner:

```
sonar-scanner
-Dsonar.eslint.reportPaths=eslint-report.json
```

83. How do you troubleoot SonarQube database connection issues?

Answer:

Check `sonar.properties`:
properties

```
sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqu
be
sonar.jdbc.username=sonar
sonar.jdbc.password=sonarpassword
```

Verify database connectivity:

```
psql -U sonar -d sonarqube -h localhost
```

1. **Restart SonarQube.**

84. How do you configure SonarQube for Go (Golang) projects?

Answer:

Install GoLint:

```
go get -u golang.org/x/lint/golint
```

Generate a report:

```
golint ./... > golint-report.txt
```

Run SonarScanner:

```
sonar-scanner
-Dsonar.go.lint.reportPaths=golint-report.txt
```

85. How do you check the history of a project in SonarQube?

Answer:

1. **Use the SonarQube Web UI under Project History.**

Use API:

```
curl -u admin:admin
"http://localhost:9000/api/measures/search_history?comp
onent=my_project&metric=coverage"
```

86. How do you handle SonarQube Elasticsearch “Yellow” or “Red” status?

Answer:

Check logs:

```
tail -f logs/es.log
```

Restart Elasticsearch:

```
systemctl restart sonarqube
```

Increase JVM heap memory in `sonar.properties`:
properties

```
sonar.search.javaOpts=-Xmx2g -Xms2g
```

87. How do you perform a zero-downtime upgrade of SonarQube?

Answer:

1. **Deploy SonarQube Data Center Edition.**
2. **Use rolling updates with a load balancer.**
3. **Perform blue-green deployment of SonarQube servers.**

88. How do you configure SonarQube for React.js projects?

Answer:

Install ESLint:

```
npm install eslint eslint-plugin-react --save-dev
```

Generate an ESLint report:

```
eslint --format json --output-file eslint-report.json
src/
```

Run SonarScanner:

```
sonar-scanner
```

```
-Dsonar.eslint.reportPaths=eslint-report.json
```

1. View the results in the SonarQube daboard.

89. How do you handle false positives in SonarQube reports?

Answer:

1. Mark issues as “False Positive” in the SonarQube UI.
2. Adjust rule severity in Quality Profiles.

Use inline annotations in code:
java

```
@SuppressWarnings("squid:S001")
```

Modify `sonar-project.properties` to ignore certain rules:
properties

```
sonar.issue.ignore.multicriteria.rule1.ruleKey=squid:S001
sonar.issue.ignore.multicriteria.rule1.resourceKey=**/*.java
```

90. How do you automate SonarQube database backups?

Answer:

PostgreSQL Backup:

```
pg_dump -U sonar -h localhost -d sonarqube -F c -b -v
-f sonar_backup.sql
```

Automate with a cron job (`crontab -e`):

```
0 2 * * * pg_dump -U sonar -h localhost -d sonarqube -F
c -b -v -f /backup/sonar_$(date +\%F).sql
```

1. **Store the backup in cloud storage or a remote server.**

91. How do you configure SonarQube for PHP projects?

Answer:

Install PHP CodeSniffer:

```
composer global require "squizlabs/php_codesniffer=*"
```

Run PHP CodeSniffer and generate a report:

```
phpcs --report=json > phpcs-report.json
```

Run SonarScanner:

```
sonar-scanner
-Dsonar.php.codesniffer.reportPaths=phpcs-report.json
```

92. How do you configure SonarQube to detect insecure APIs?

Answer:

1. Enable Security Hotspots in Quality Profiles.
2. Use OWASP and SANS rules for vulnerability detection.

Run SonarQube Security Report API:

```
curl -u admin:admin "http://localhost:9000/api/security_hotspots/search"
```

3. Integrate SAST tools like Checkmarx for deeper security analysis.

93. How do you fix SonarQube “Background Tasks Pending” issue?

Answer:

Restart Compute Engine via API:

```
curl -u admin:admin -X POST "http://localhost:9000/api/ce/execute"
```

Check logs:

```
tail -f logs/ce.log
```

Increase database performance (PostgreSQL):

```
VACUUM FULL;
REINDEX DATABASE sonarqube;
```

94. How do you analyze large codebases efficiently in SonarQube?

Answer:

Run incremental analysis:
properties

```
sonar.scm.exclusions.disabled=true
```

Increase database performance:

```
ALTER DATABASE sonarqube SET work_mem = '256MB';
```

Enable parallel execution:
properties

```
sonar.ce.workerCount=4
```

95. How do you configure SonarQube for Dart/Flutter projects?

Answer:

Install Dart Code Metrics:

```
dart pub global activate dart_code_metrics
```

Generate a report:

```
dart run metrics --report=json > metrics-report.json
```

Run SonarScanner:

```
sonar-scanner
-Dsonar.dart.metrics.reportPaths=metrics-report.json
```

96. How do you customize SonarQube Quality Profiles?

Answer:

1. **Go to SonarQube UI → Quality Profiles.**
2. **Create a new profile or copy an existing one.**
3. **Add or remove rules based on your requirements.**
4. **Assign the custom Quality Profile to projects.**

97. How do you manage SonarQube user roles via API?

Answer:
Create a new user:

```
curl -u admin:admin -X POST
"http://localhost:9000/api/users/create?login=developer
&password=dev123"
```

Assign a role:

```
curl -u admin:admin -X POST
"http://localhost:9000/api/permissions/add_user?login=d
eveloper&permission=scan"
```

98. How do you troubleoot SonarQube not owing issues after analysis?

Answer:

1. **Check the analysis logs for warnings.**

Ensure the project key is correct:

```
sonar-scanner -Dsonar.projectKey=my_project
```

Check database connectivity:

```
SELECT * FROM projects WHERE kee='my_project';
```

Increase Elasticsearch memory:
properties

```
sonar.search.javaOpts=-Xmx4g -Xms4g
```

99. How do you enforce SonarQube rules in pre-commit hooks?

Answer:

Install SonarLint CLI:

```
npm install -g sonarlint
```

Add a pre-commit hook in `.git/hooks/pre-commit`:

```
sonarlint lint --project my_project
if [ $? -ne 0 ]; then
       echo "SonarLint found issues. Fix them before committing."
```

```
    exit 1
fi
```

100. How do you deploy a high-availability SonarQube cluster?

Answer:

1. **Deploy SonarQube Data Center Edition.**
2. **Configure PostgreSQL replication for database HA.**
3. **Deploy multiple SonarQube nodes behind a load balancer.**
4. **Use Kubernetes Helm Charts:**

```
helm install sonarqube sonarqube/sonarqube
```