SonarQube for SAST - DevOps Implementation Guide

What Is SonarQube?

SonarQube is an open-source Static Application Security Testing (SAST) and code quality platform. It scans source code to detect:

- Security vulnerabilities
- Bugs
- Code smells (maintainability issues)
- Code coverage (when combined with test reports)

Why SonarQube for You?

SonarQube fits your setup because:

- Ubuntu VM: Easy Docker deployment or native installation.
- Docker: SonarQube runs well in containers.
- Java Maven Spring Boot: First-class support with Maven plugins.
- GitLab CI/CD: Seamless integration via pipeline jobs.
- Reporting Requirement: Full-featured UI and exportable reports.

Key Concepts You Must Understand

1. Static Code Analysis (SAST): Analyzes source code without running it.
2. Quality Gates: Predefined rules to decide if code passes standards.
3. Code Smells vs. Vulnerabilities: Code smells are bad practices; vulnerabilities are potential exploits.
4. Issues, Rules, and Severity Levels: SonarQube assigns severity levels from info to blocker.

SonarQube for SAST - DevOps Implementation Guide

Architecture Overview

Your Codebase -> Sonar Scanner via Maven -> SonarQube Server -> Web UI / Dashboard.

How SonarQube Works in CI/CD

1. Developer pushes code to GitLab.

2. GitLab pipeline compiles, tests, and runs sonar:sonar via Maven.

3. Scanner analyzes Java code.

4. Results sent to SonarQube server.

5. Reports available in the dashboard.

Reporting & Dashboards

- Summary metrics: bugs, vulnerabilities, code smells.

- Detailed findings: file and line number.

- Security hotspots: areas needing manual review.

- Report export: via API or plugins.

Security-Specific Features (SAST)

- Recognizes known vulnerability patterns.

- Detects flaws like SQL injection, hardcoded credentials.

- Highlights security hotspots for manual review.

SonarQube for SAST - DevOps Implementation Guide

Important Things to Configure

- Project Key: Unique project identifier.
- Authentication Token: Used by the CI pipeline.
- Host URL: SonarQube server address.
- Quality Gate: Defines pass/fail criteria.

Pre-Implementation Checklist

- Deploy SonarQube in Docker (on Ubuntu VM)
- Access SonarQube UI
- Create a project
- Store token in GitLab CI/CD
- Configure Maven plugin
- Add job in .gitlab-ci.yml
- Test pipeline with scan
- Validate results in UI

Common Pitfalls

- Scan upload issues: Check token, URL, connectivity.
- Server not found: Ensure container is running.
- Analysis error: Check Maven logs.
- No vulnerabilities found: Misconfiguration or clean code.

SonarQube for SAST - DevOps Implementation Guide

Summary

SonarQube is a powerful SAST tool for Java and fits perfectly with your tech stack.

It allows automated scanning, detailed reporting, and requires no code modification.