Strategic Implementation Plan for SonarQube-Based Code Quality Management

1. SonarQube Deployment & Architecture

To meet the client's requirement for visibility, compliance, and audit-ready quality tracking, SonarQube will be deployed in a centralized but scalable architecture. A dedicated SonarQube server will be hosted within the client's secure infrastructure (either on-premises or a cloud environment approved for government healthcare projects). For resilience, the setup will use a high-availability configuration with a PostgreSQL backend for data storage, ensuring audit logs and quality data are securely preserved. Network segregation and TLS encryption will be enforced to comply with healthcare security policies. Developers, QA staff, and project managers will connect to this single SonarQube instance to ensure consistency of analysis across Java, JavaScript, and Python services.

2. Clean Code & "Clean as You Code" Alignment

The implementation will follow the principle of "Clean as You Code" to prevent accumulation of technical debt. This means new or modified code must meet defined quality standards before it can be merged into the main branch. Legacy issues in the existing codebase will be tracked but not block current development; instead, the team will focus on gradually improving older code through refactoring during regular sprint work. This approach balances the need for forward-looking quality with the practical constraints of large legacy systems.

3. Static Code Analysis Pipeline Integration

SonarQube scanners will be integrated into the CI/CD pipelines built on GitHub Actions. Each microservice (Java, JavaScript, Python) will include a step in its build workflow that runs the appropriate SonarQube scanner (e.g., Maven plugin for Java, npm-based scanner for JavaScript, SonarScanner for Python). Pull request checks will be configured so that SonarQube automatically analyzes code changes and provides feedback within GitHub. If quality gates are not met, the workflow will fail, preventing merge into protected branches. This enforces automation without adding significant manual overhead for developers.

4. Quality Profiles and Gates Customization

Out-of-the-box SonarQube rules will be customized to align with the client's internal compliance policies. For example, strict rules will be enforced on security vulnerabilities, code smells, and duplications due to the healthcare system's sensitivity. Custom quality profiles will be created per language, ensuring relevant rule sets for Java (back-end services), JavaScript (front-end and APIs), and Python (data processing scripts). Quality gates will be tailored to mandate zero critical vulnerabilities, a maximum allowable code duplication threshold, and acceptable coverage levels from unit tests. These gates will be validated with stakeholders before final rollout to ensure compliance with audit expectations.

5. User and Role Management Strategy

Role-based access will be configured to align with the project's governance model. Developers will have permissions to analyze code and view results for their services. QA engineers will have access to broader dashboards and the ability to configure quality profiles within their domains. Project managers will be assigned read-only access to portfolio dashboards to track overall progress without altering configurations. SonarQube will be integrated with the client's identity provider (e.g., LDAP or SSO) to simplify authentication and enforce consistent role-based security.

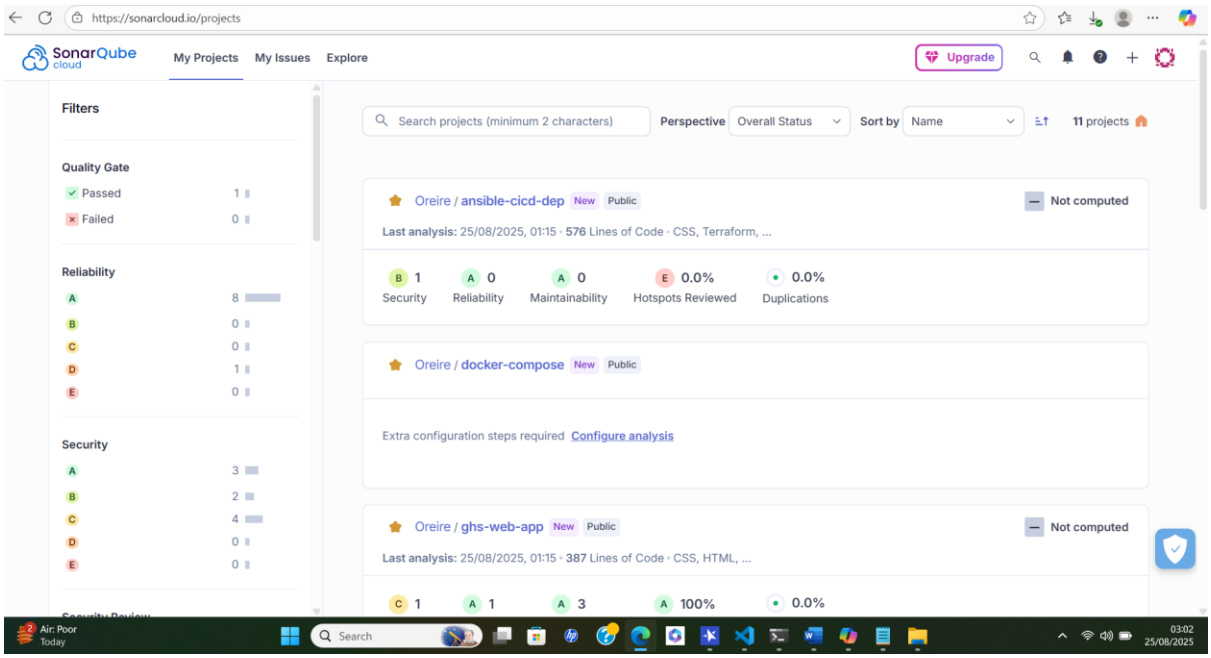
6. Dashboarding and Reporting Use

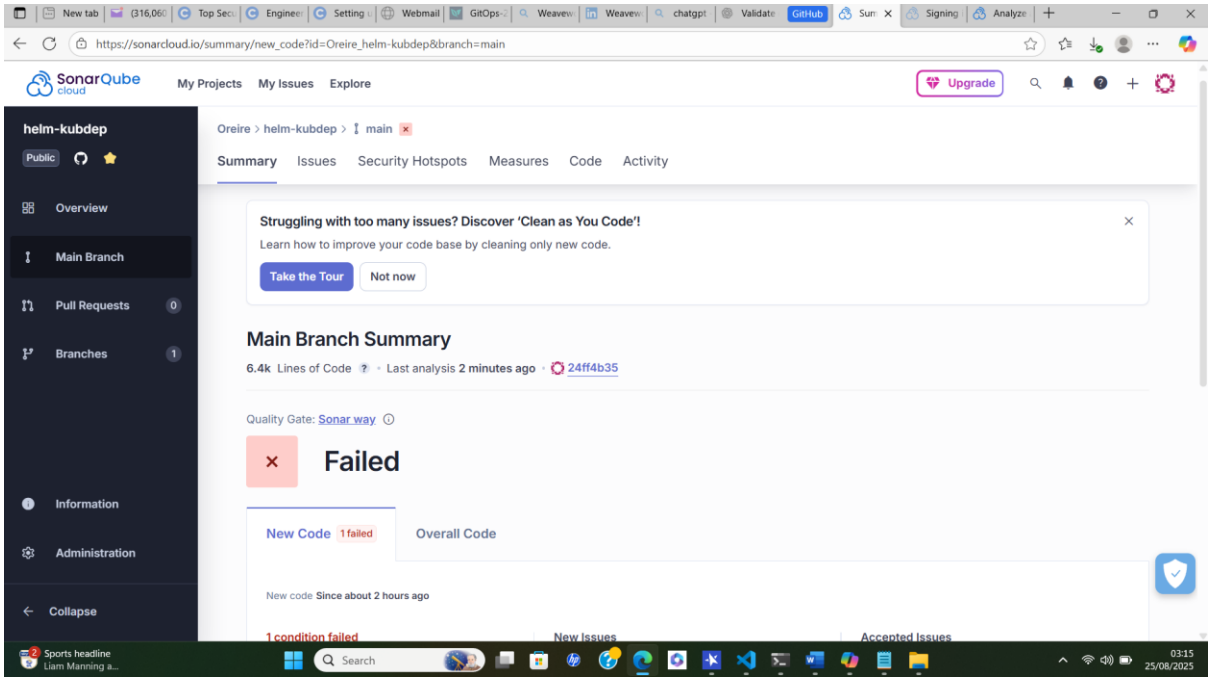
Dashboards will be designed to provide visibility at multiple levels: project, service, and portfolio. Developers can see detailed issue reports for their services, QA can track compliance against quality gates, and project managers can view high-level metrics such as technical debt trends, security posture, and compliance with agreed standards. Automated reports will be scheduled and distributed to stakeholders, ensuring traceability and evidence for audits. Custom widgets and health indicators will highlight hotspots in the codebase, guiding teams toward continuous improvement.

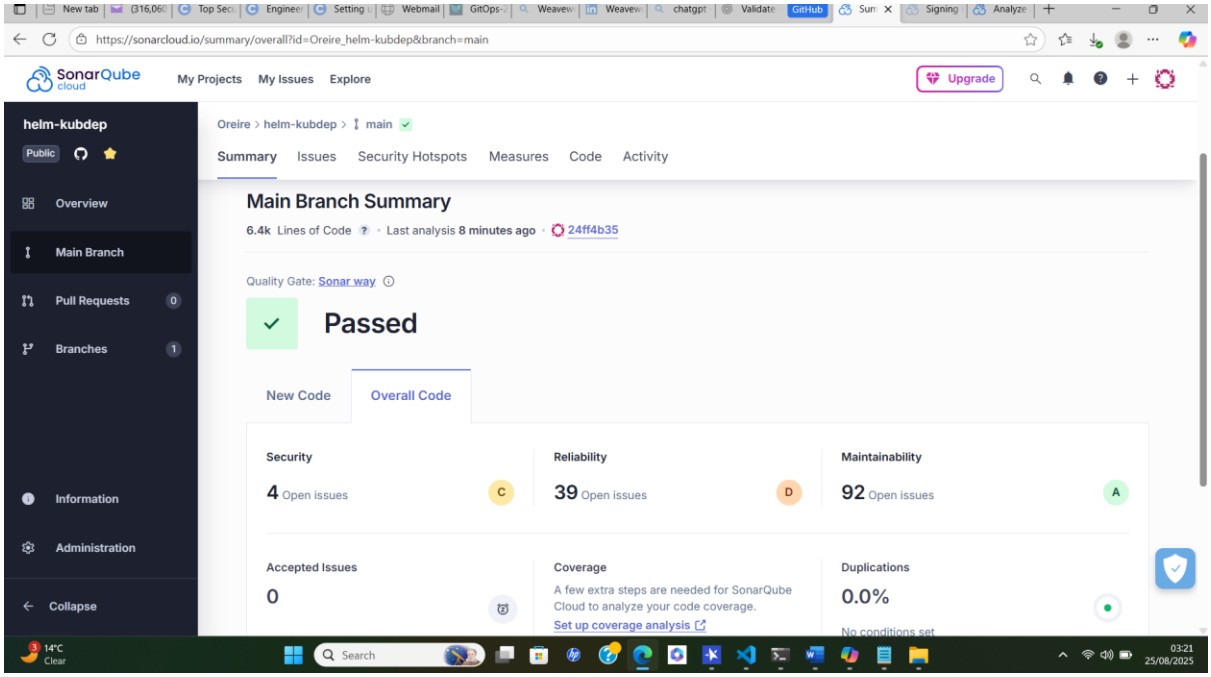
Conclusion

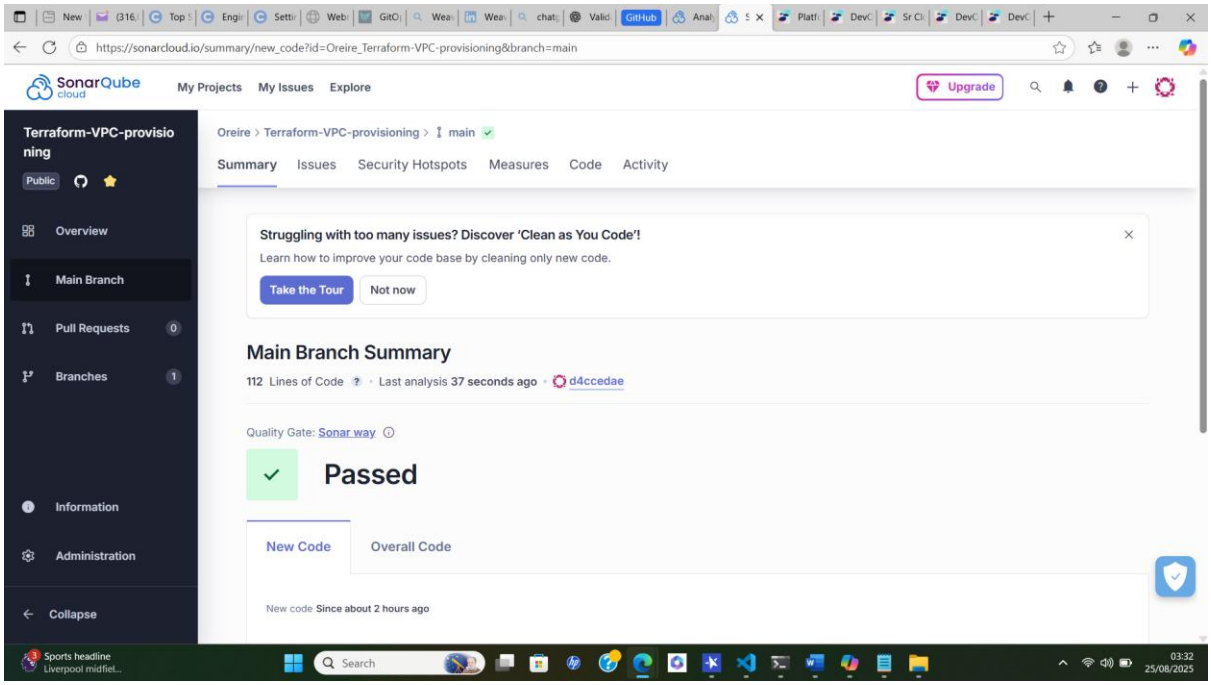
This strategy ensures SonarQube is deployed as a central pillar of code quality management, tightly integrated into the client's development workflows. By combining clean code principles, automated static analysis, tailored quality gates, and role-based visibility, the implementation will deliver compliance, transparency, and long-term sustainability in managing technical debt and code quality.

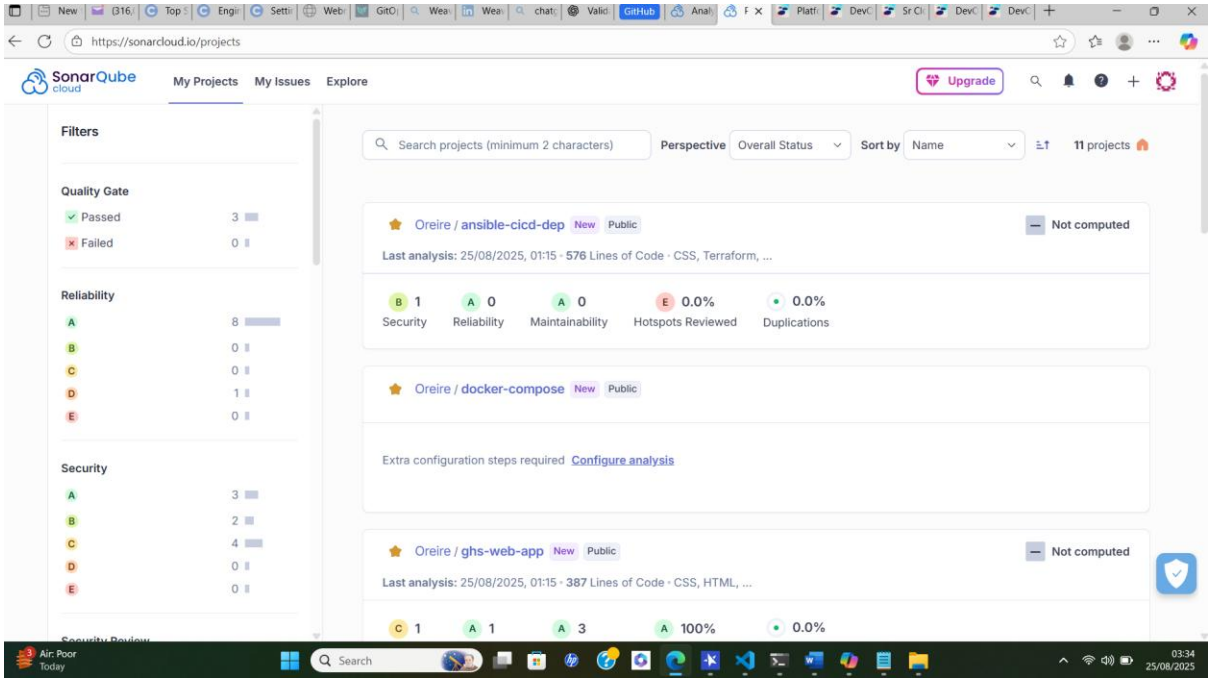
Screenshots of Some Code Analysis Using SonarQube






https://sonarcloud.io/summary/overall?id=Oreire_helm-kubdep&branch=main
SonarQube cloud
My Projects
My Issues
Explore
Upgrade
helm-kubdep
Public
Overview
Main Branch
Pull Requests
Branches
Information
Administration
Collapse
Oreire > helm-kubdep > main
Summary
Issues
Security Hotspots
Measures
Code
Activity
Main Branch Summary
6.4k Lines of Code
Last analysis 8 minutes ago
24ff4b35
Quality Gate: Sonar way
Passed
New Code
Overall Code
Security
4 Open issues
C
Reliability
39 Open issues
D
Maintainability
92 Open issues
A
Accepted Issues
0
Coverage
A few extra steps are needed for SonarQube Cloud to analyze your code coverage.
Set up coverage analysis
Duplications
0.0%
No conditions set
14°C
Clear
Search
03:21
25/08/2025


https://sonarcloud.io/summary/new_code?id=Oreire_Terraform-VPC-provisioning&branch=main
SonarQube cloud
My Projects
My Issues
Explore
Upgrade
Terraform-VPC-provisioning
Public
Overview
Main Branch
Pull Requests
Branches
Information
Administration
Collapse
Oreire > Terraform-VPC-provisioning > main
Summary
Issues
Security Hotspots
Measures
Code
Activity
Struggling with too many issues? Discover 'Clean as You Code'!
Learn how to improve your code base by cleaning only new code.
Take the Tour
Not now
Main Branch Summary
112 Lines of Code
Last analysis 37 seconds ago
d4ccedae
Quality Gate: Sonar way
Passed
New Code
Overall Code
New code Since about 2 hours ago
Sports headline
Liverpool midfiel...
Search
03:32
25/08/2025


https://sonarcloud.io/projects
SonarQube cloud
My Projects
My Issues
Explore
Upgrade
Filters
Quality Gate
Passed 3
Failed 0
Reliability
A 8
B 0
C 0
D 1
E 0
Security
A 3
B 2
C 4
D 0
E 0
Search projects (minimum 2 characters)
Perspective
Overall Status
Sort by
Name
11 projects
Oreire / ansible-cicd-dep
New
Public
Not computed
Last analysis: 25/08/2025, 01:15 · 576 Lines of Code · CSS, Terraform, ...
B 1 Security
A 0 Reliability
A 0 Maintainability
E 0.0% Hotspots Reviewed
0.0% Duplications
Oreire / docker-compose
New
Public
Extra configuration steps required
Configure analysis
Oreire / ghs-web-app
New
Public
Not computed
Last analysis: 25/08/2025, 01:15 · 387 Lines of Code · CSS, HTML, ...
C 1
A 1
A 3
A 100%
0.0%
Air: Poor
Today
Search
03:34
25/08/2025