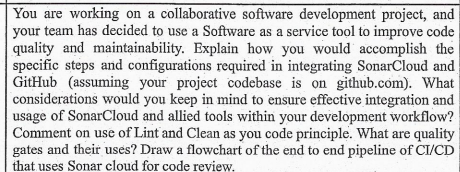
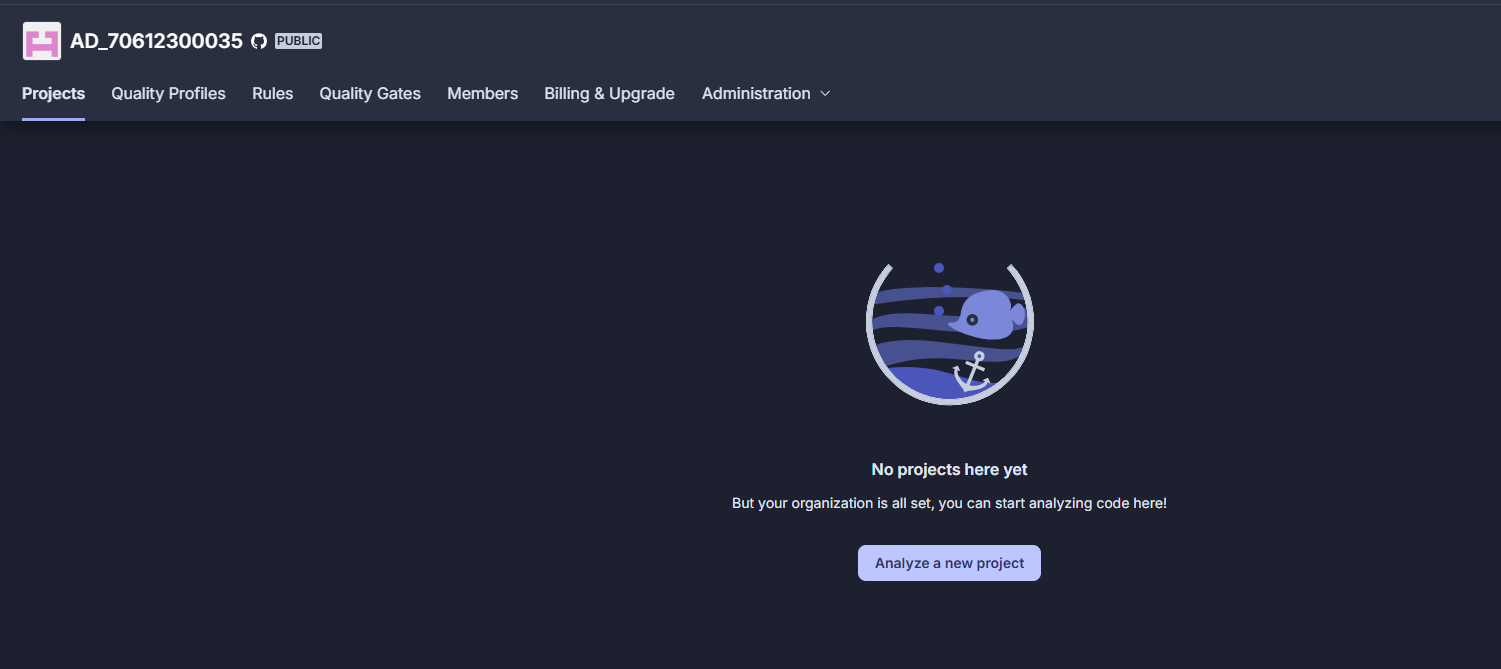
Q3]

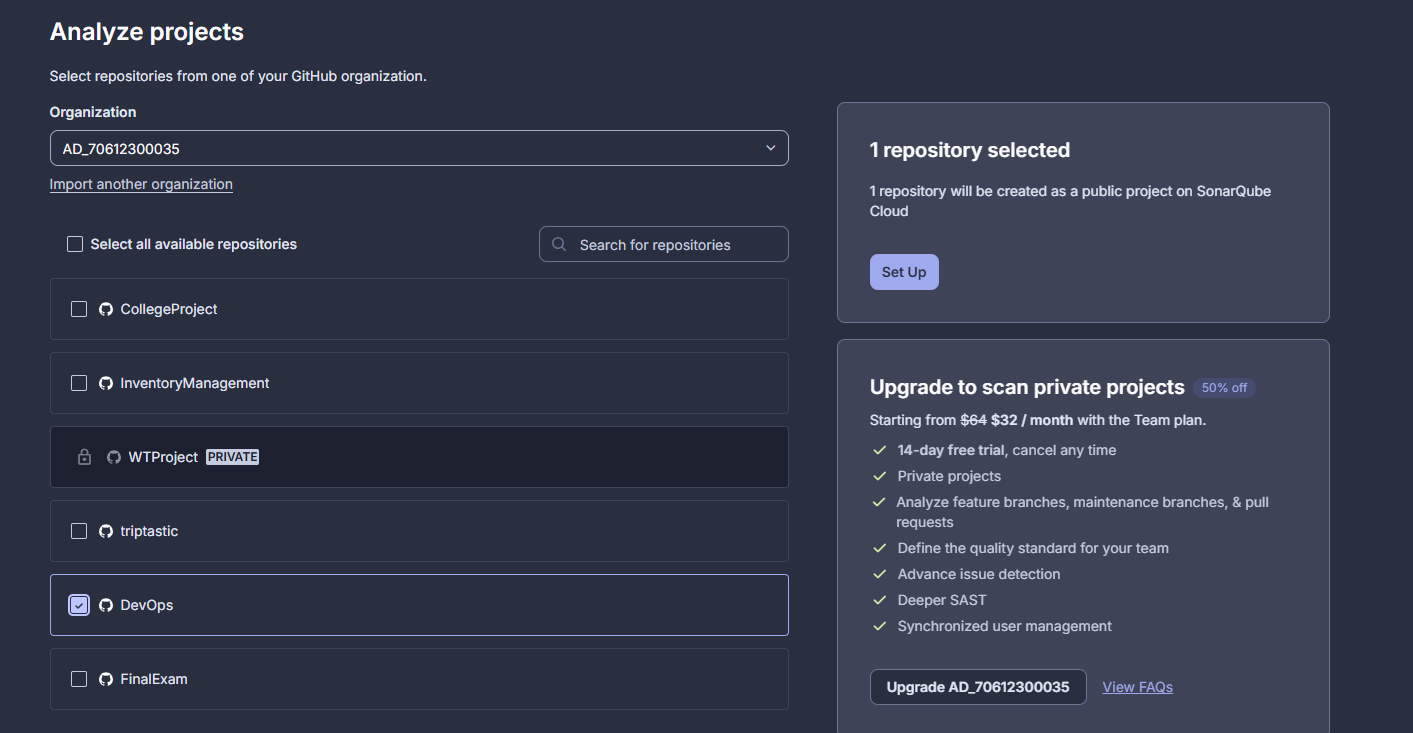


Steps:

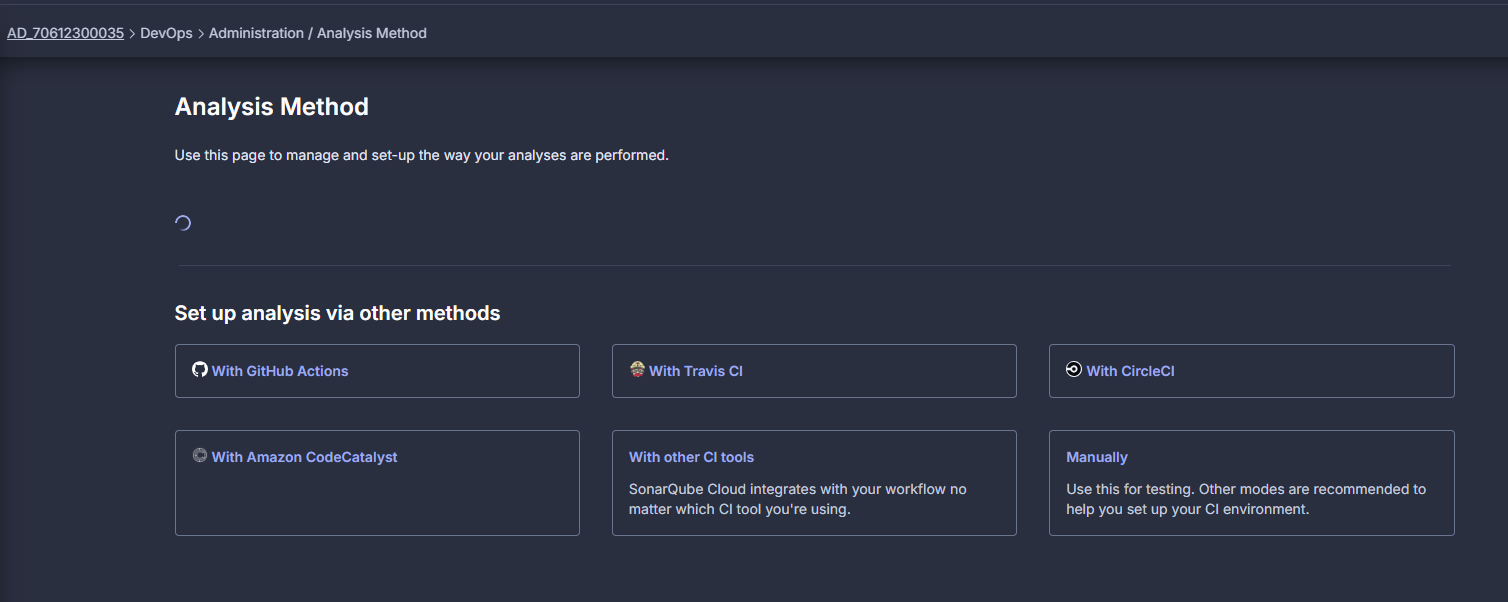
Analyze the Project



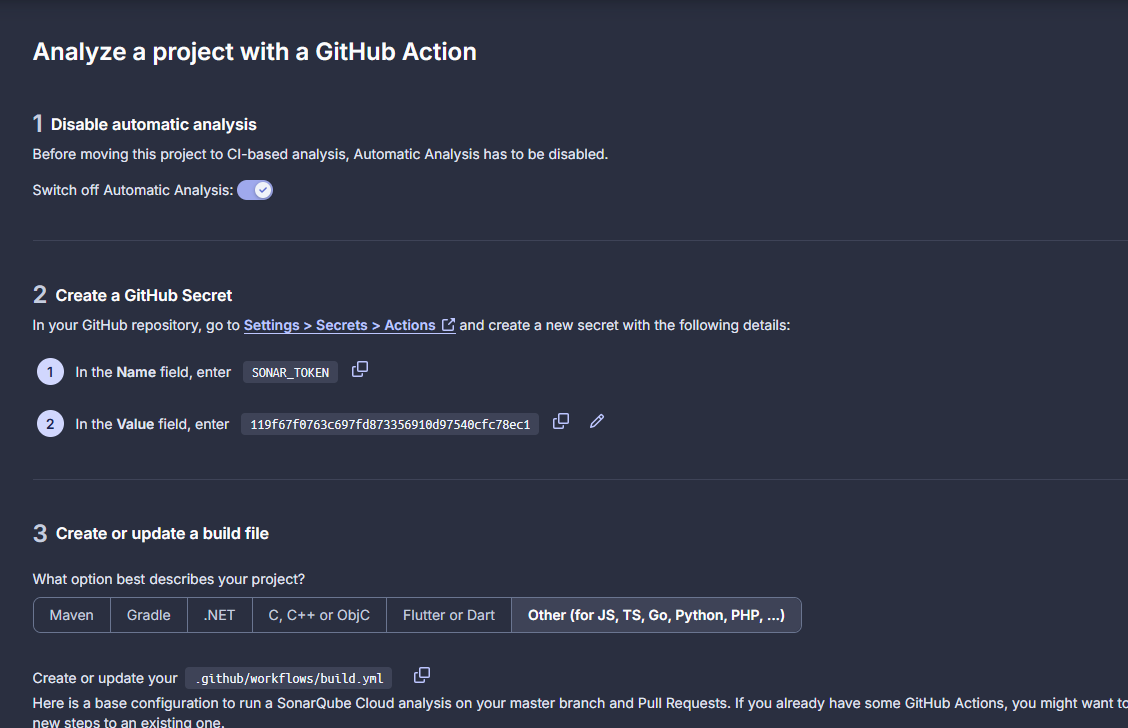
Select a Repository



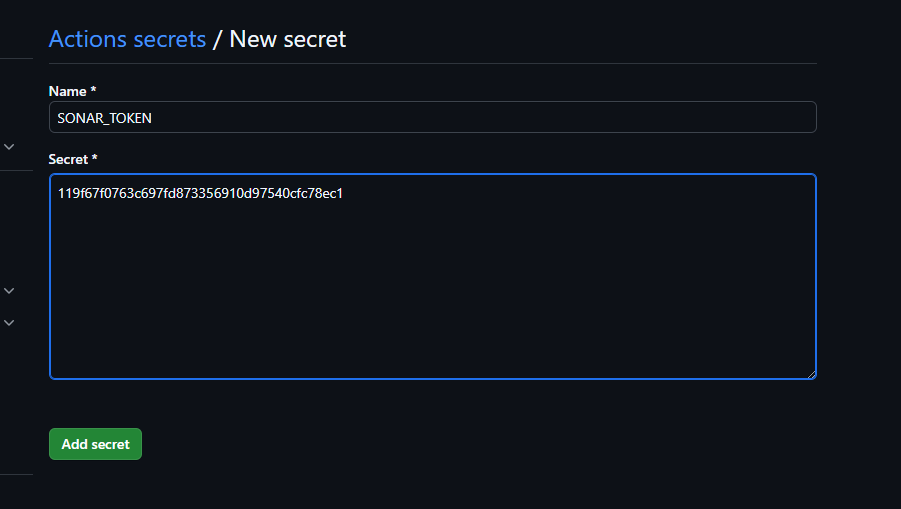
Select With GitHub Action in Administration and Analysis Method

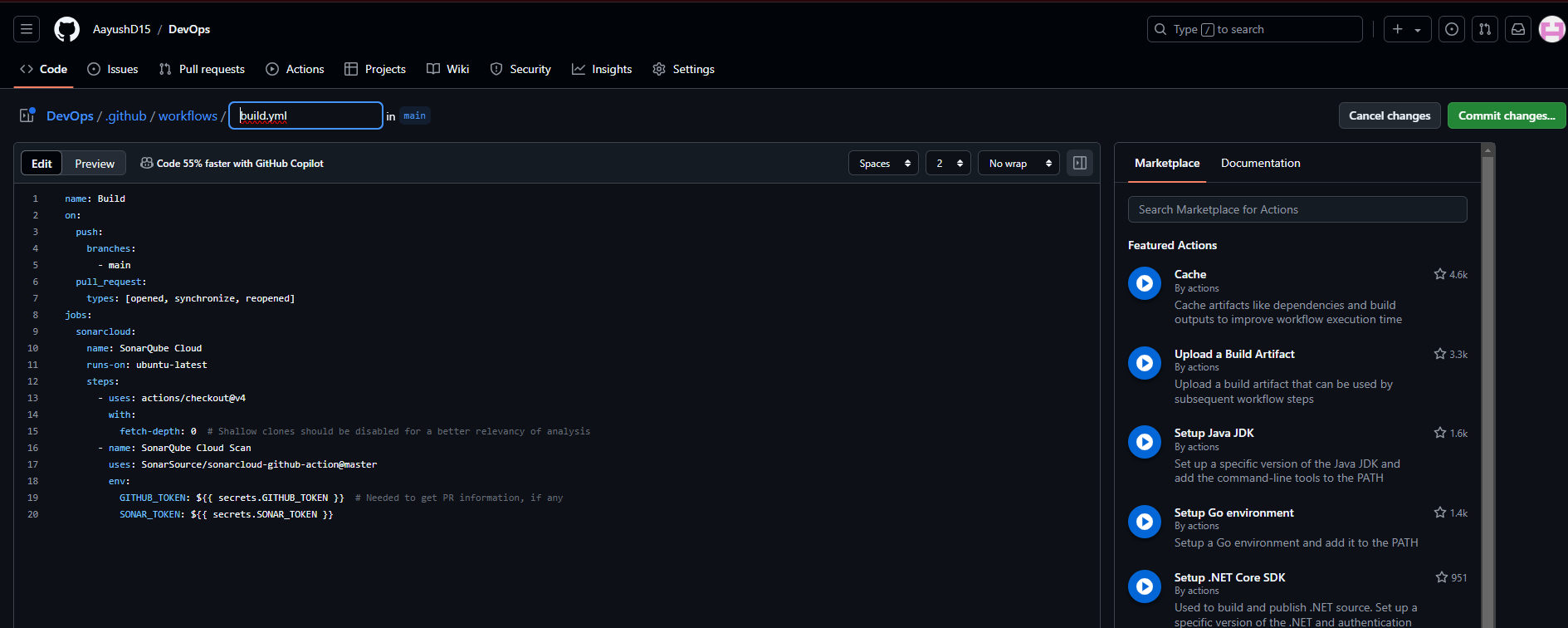


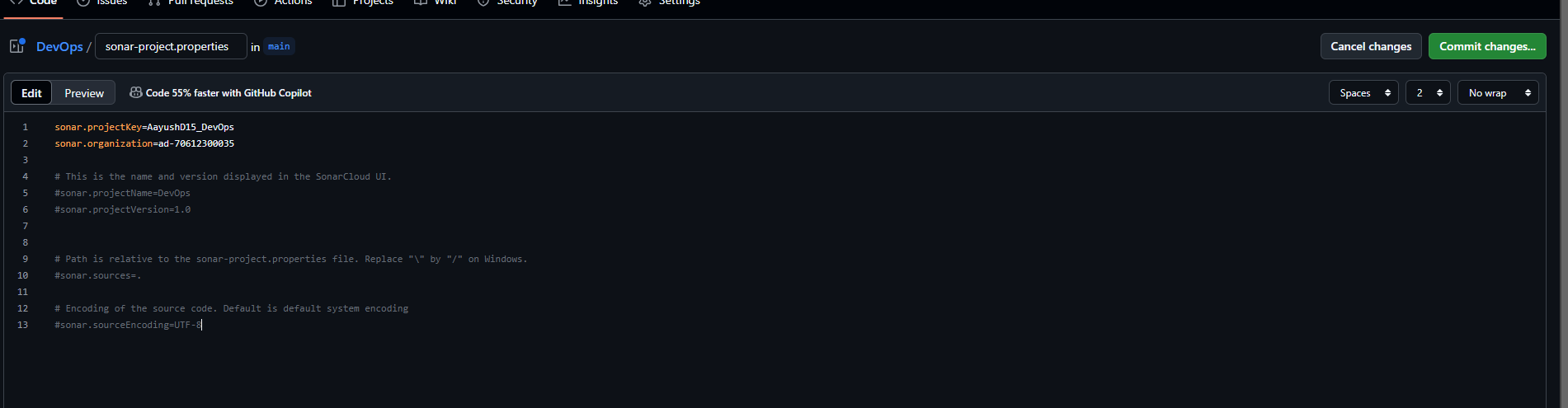
Add the File Configuration in GitHub Repository



Add Secret in GiHub



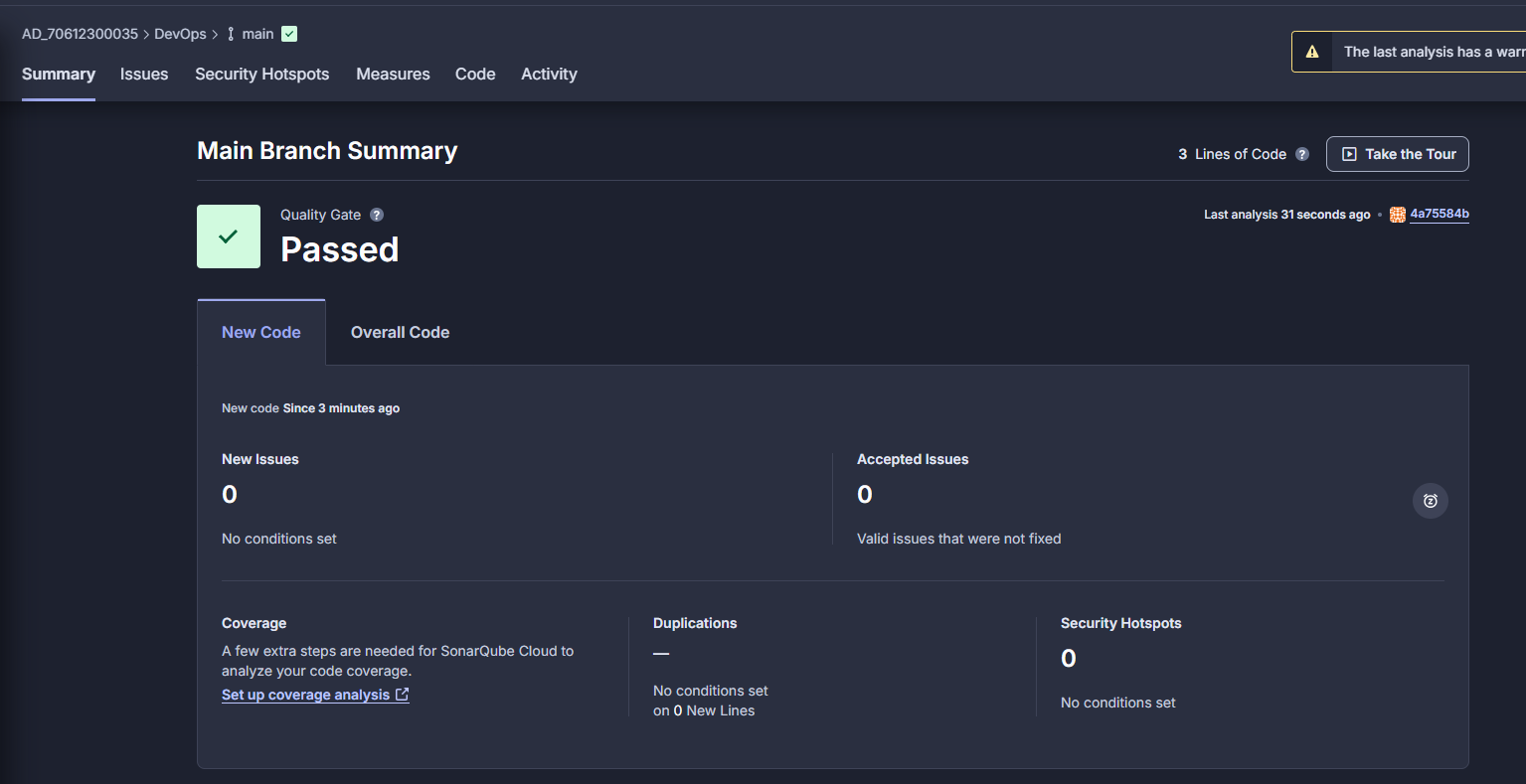




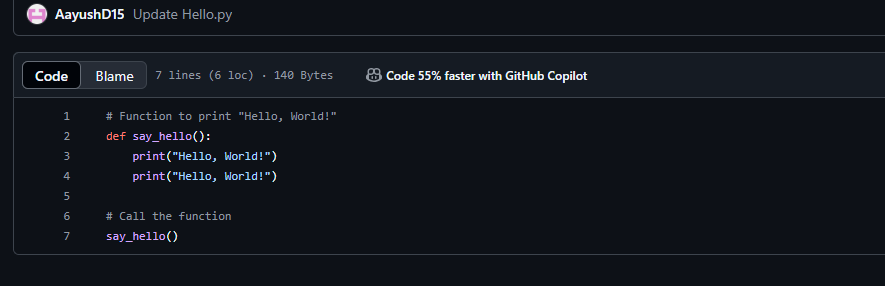
GitHub Repository Will Look like this

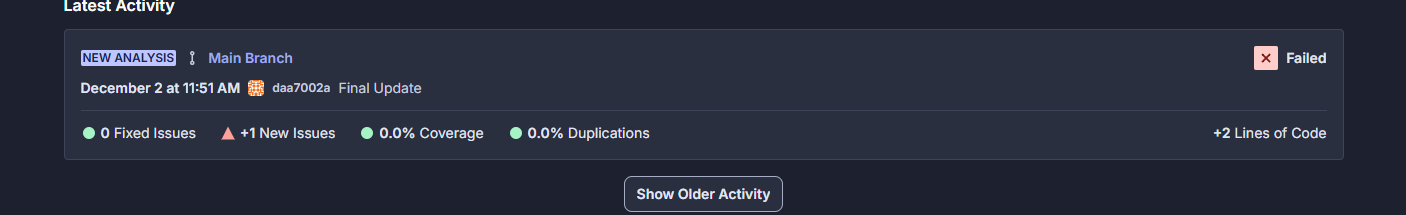


Analysis Output

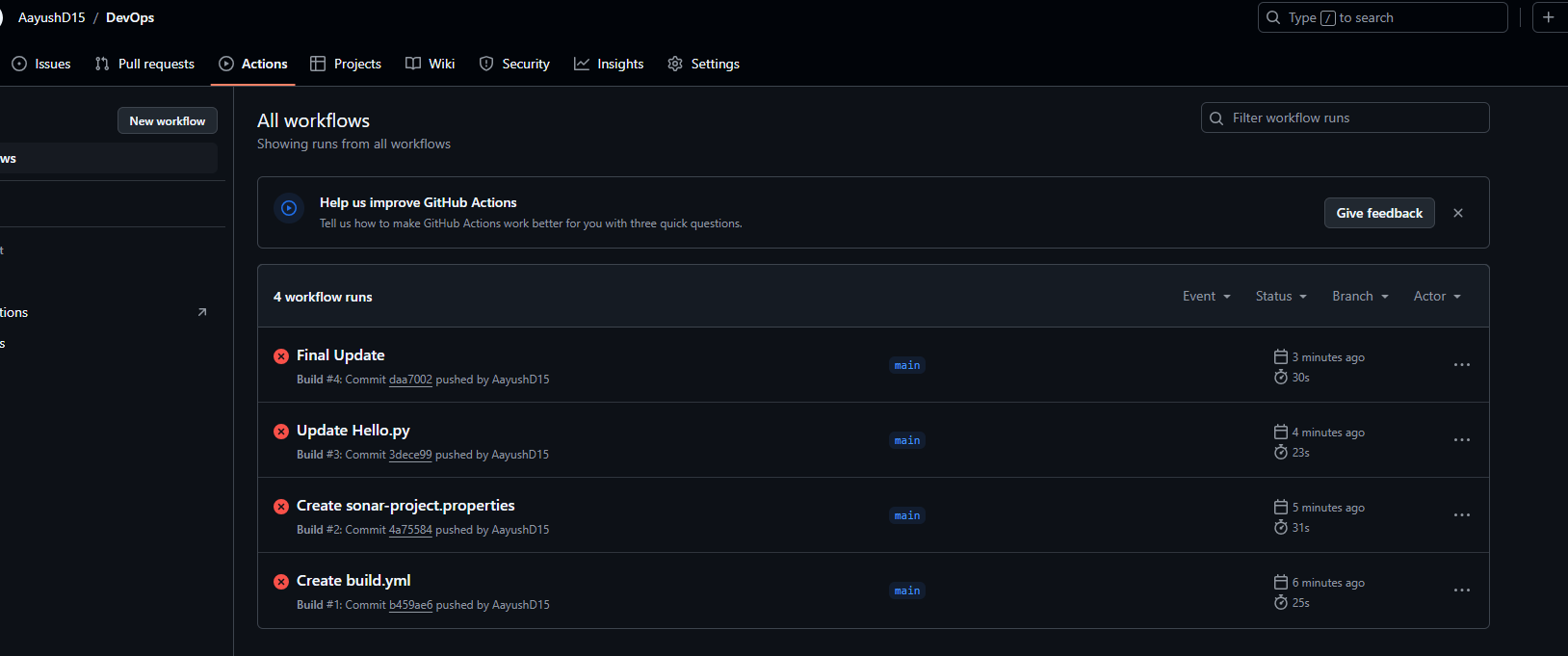


After Making Some Changes in Repository and adding Duplication Line



Analysis Shows Failed as Duplicate Lines are there in Code 

Output of Workflow in our Repository



**Linting and Clean Code**Linting tools analyze code to detect errors, bugs, and style issues. Adopting a "Lint First" approach helps ensure code consistency and catch problems early. The "Clean Code" principle focuses on writing code that is easy to understand, modify, and maintain. Following linting and clean coding practices improves code quality, reduces technical debt, and makes the codebase more robust and scalable.

**Quality Gates**Quality gates are checkpoints in the CI/CD process that enforce defined criteria before allowing an application to progress to the next stage. They ensure code quality, control deployments, maintain consistency, mitigate risks, and promote collaboration. Common quality gate checks include passing tests, maintaining code coverage, adhering to standards, and meeting quality metric thresholds. Quality gates help deliver reliable, high-performing software.

**FlowChart**

