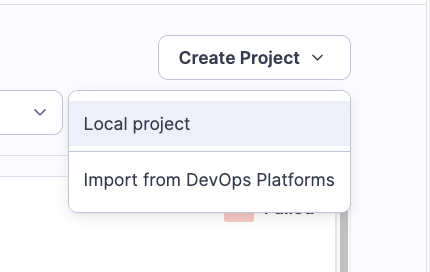
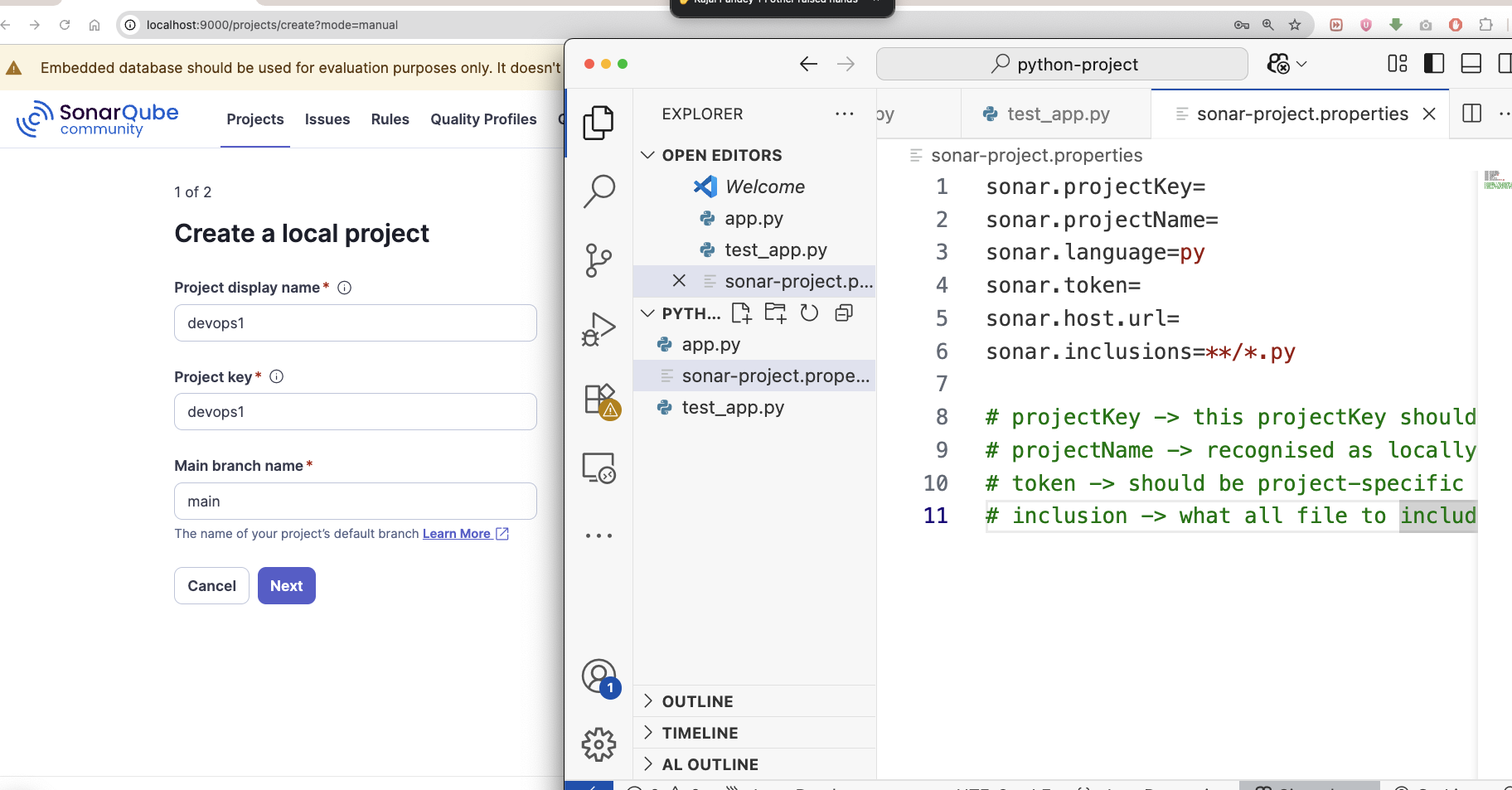




So now we have to create a project in sonarqube

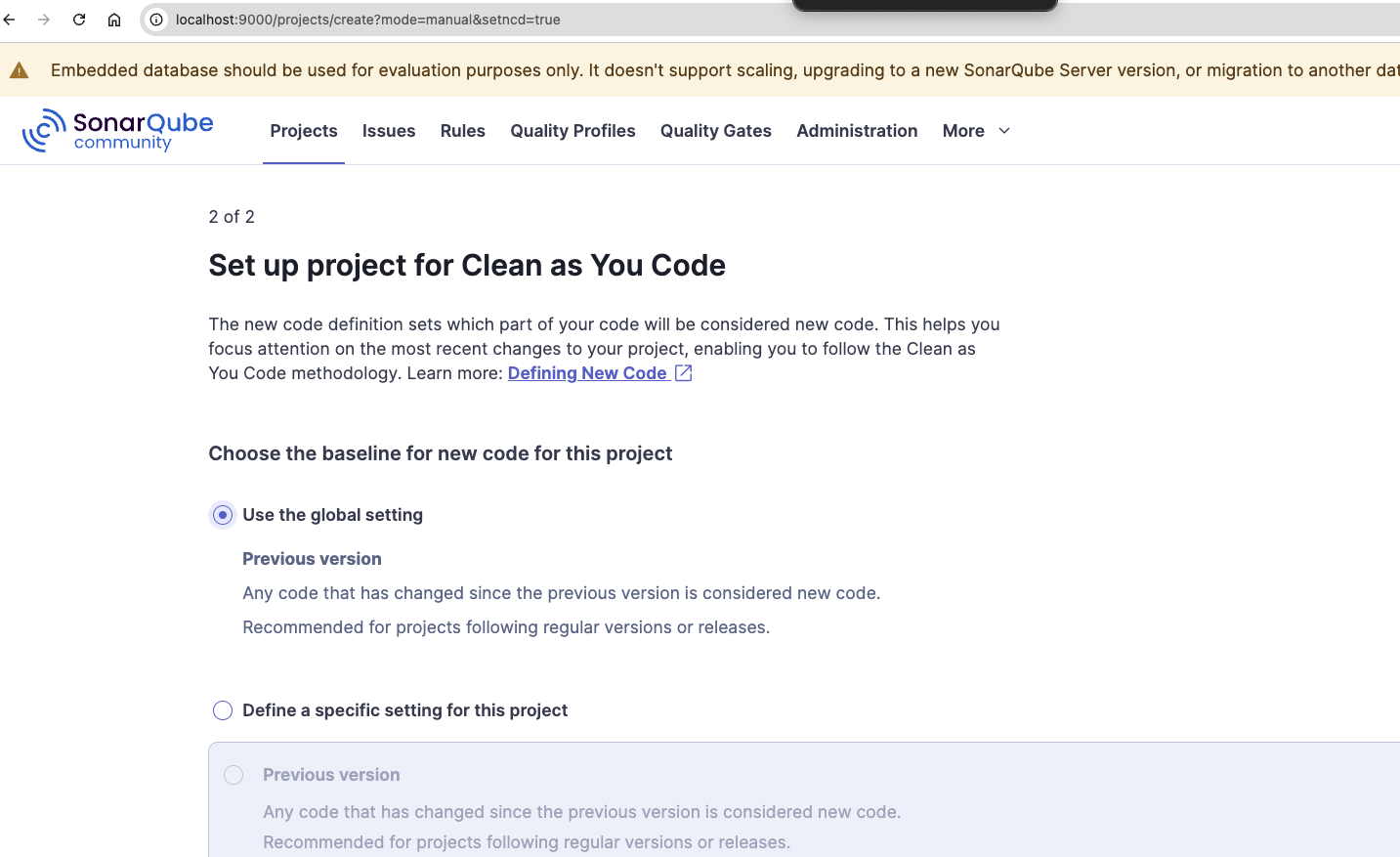






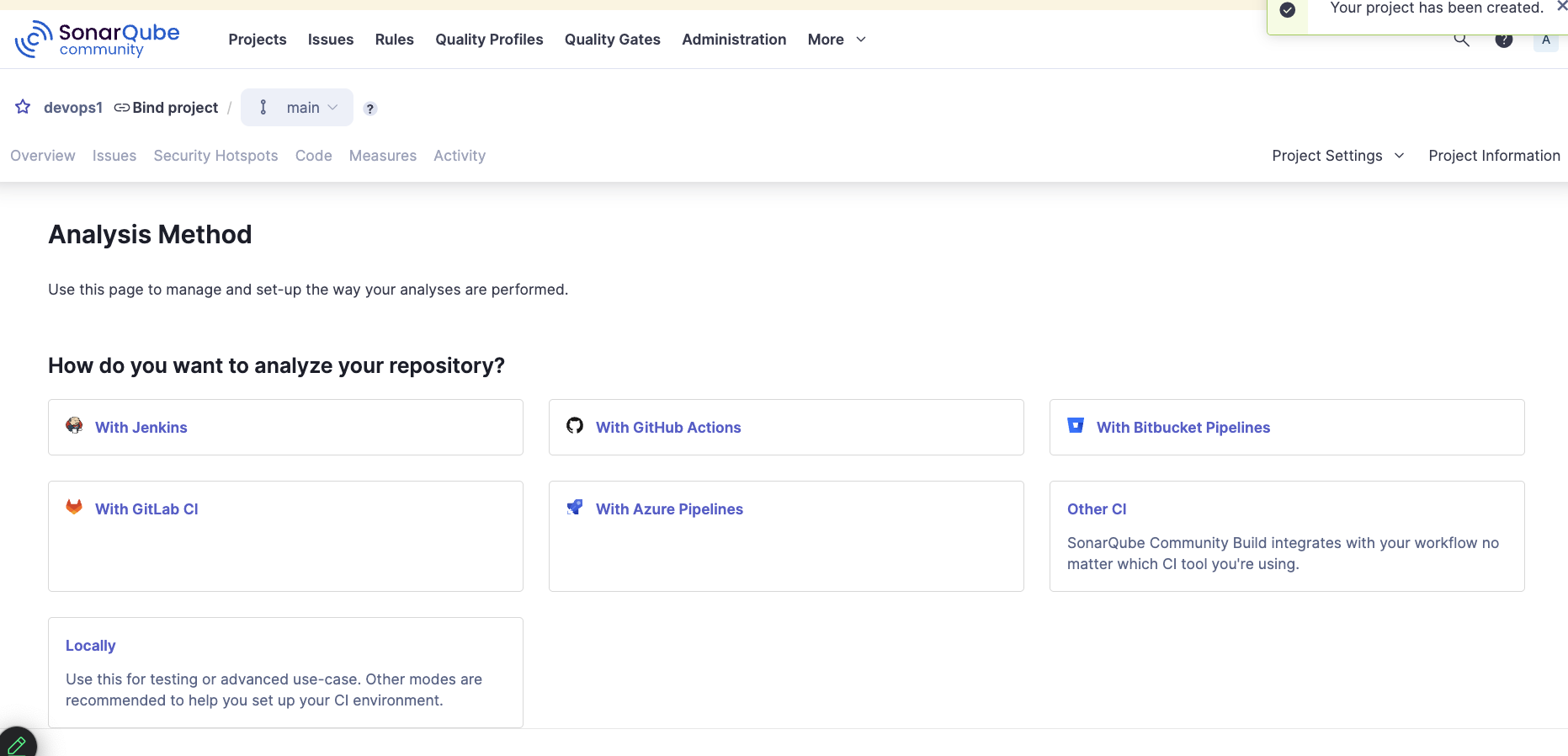


We have to provide the same name as projectKey in props file with seonarqube projectName





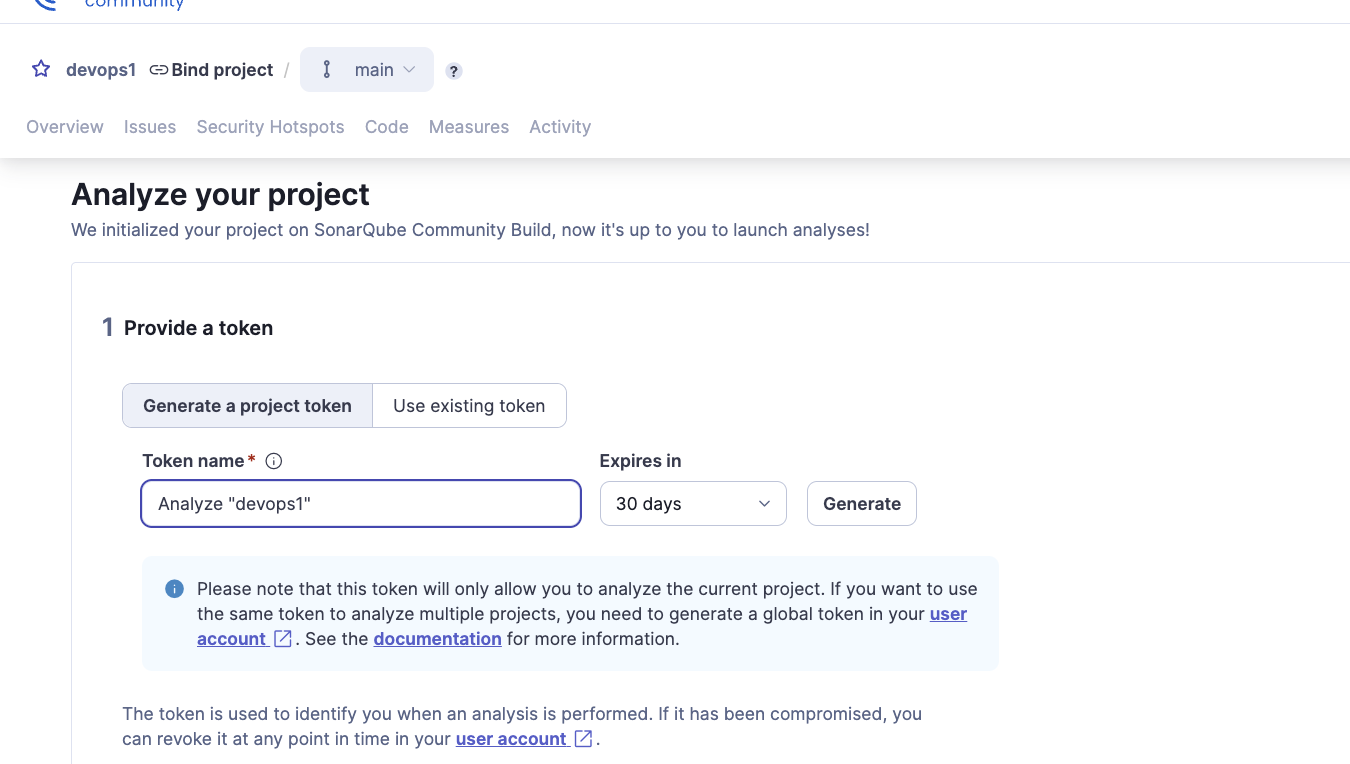
We assumption that: project reporting -> Independent Deployment (LIVE)



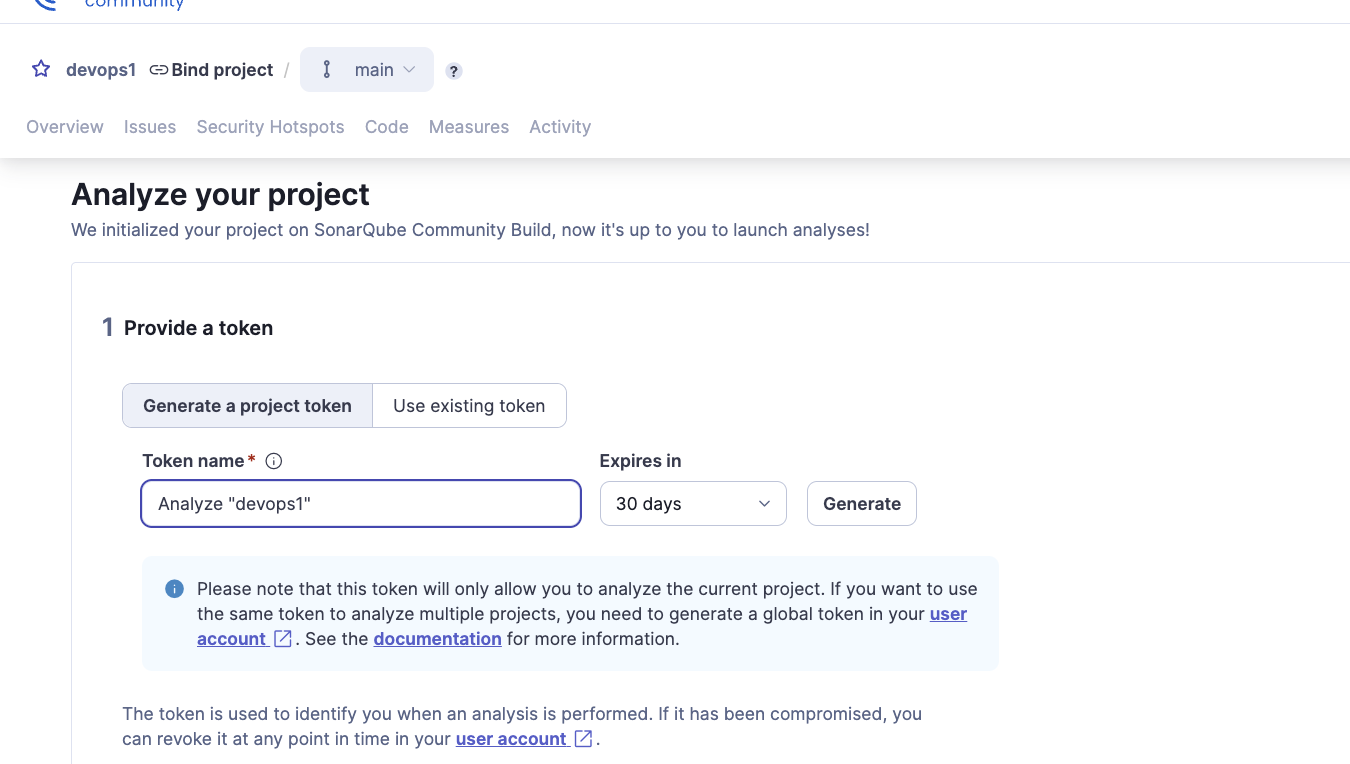


Since we are doing this with Local project:

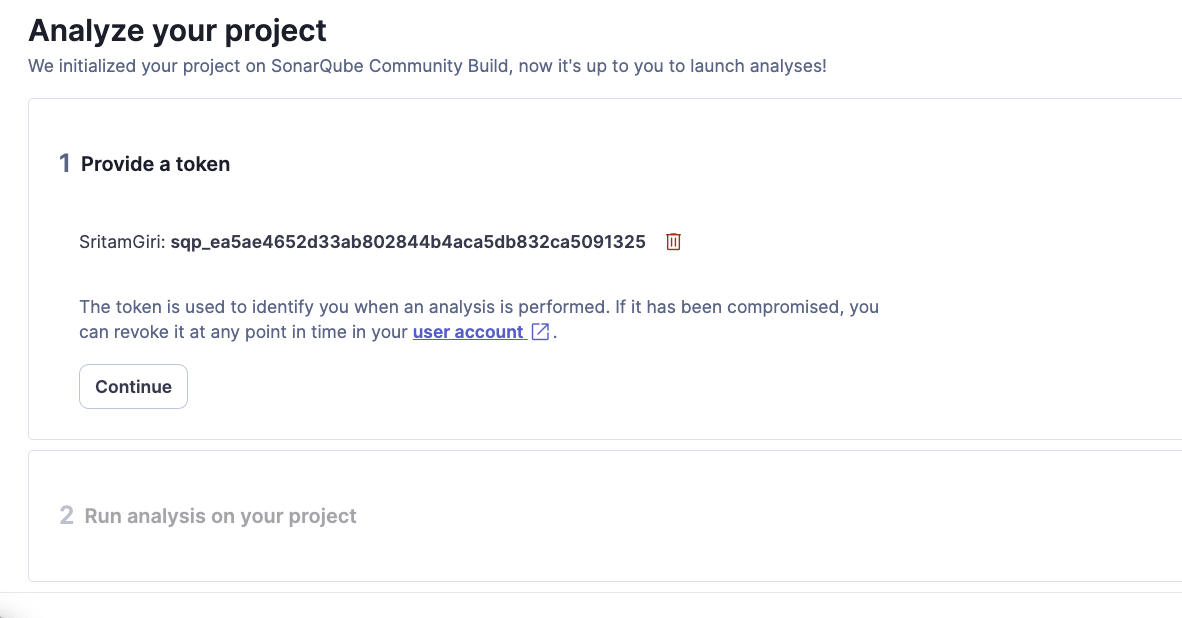
So select the option Locally



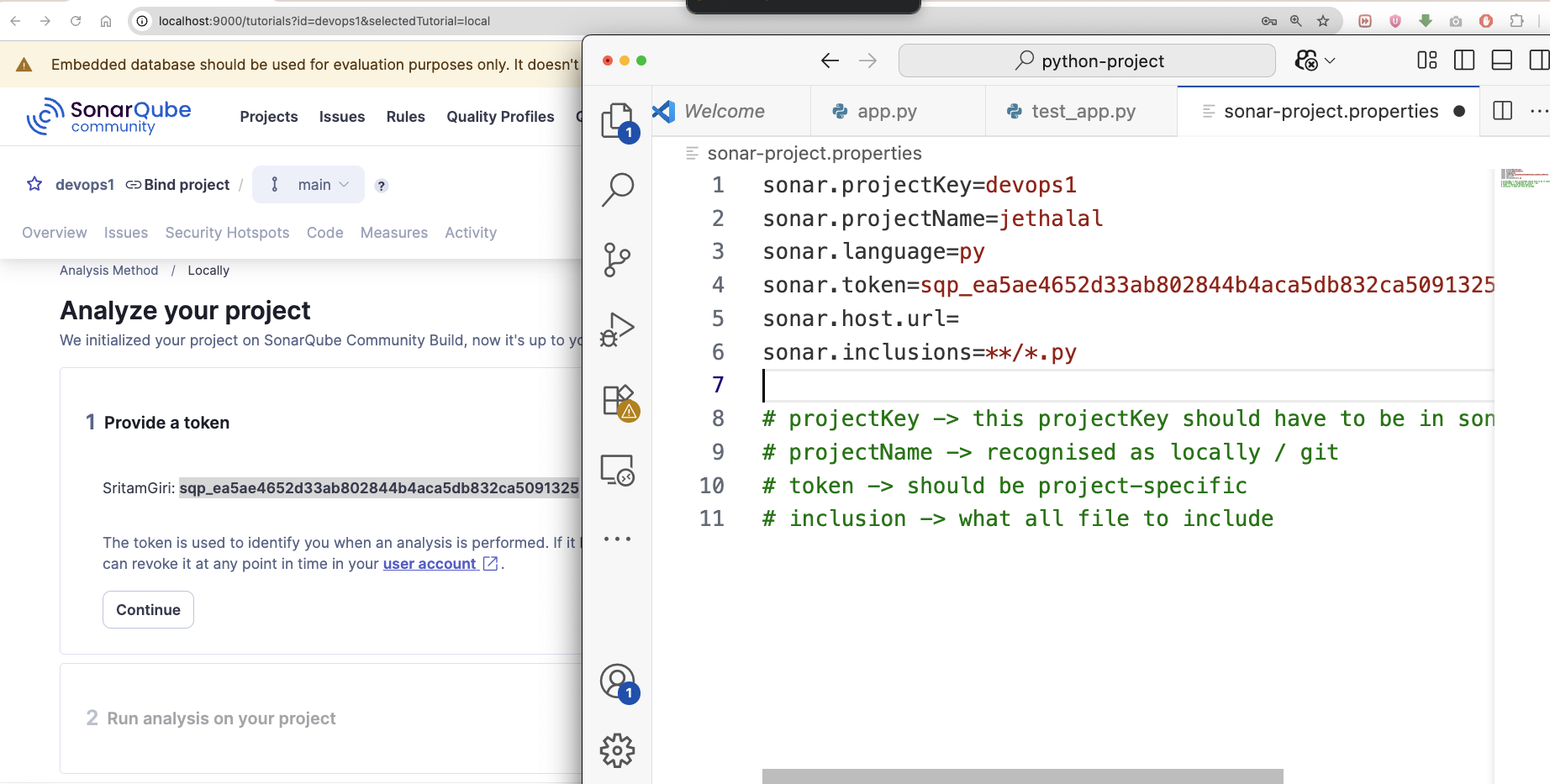
After select locally option: create a token.





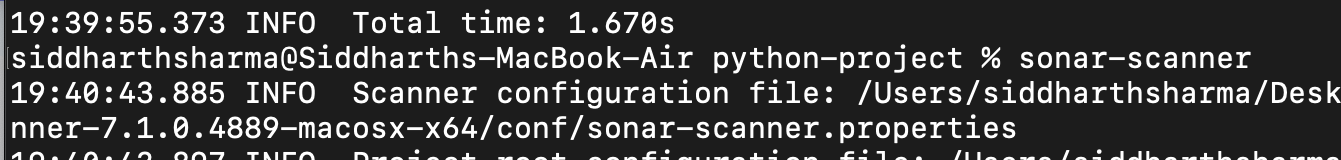


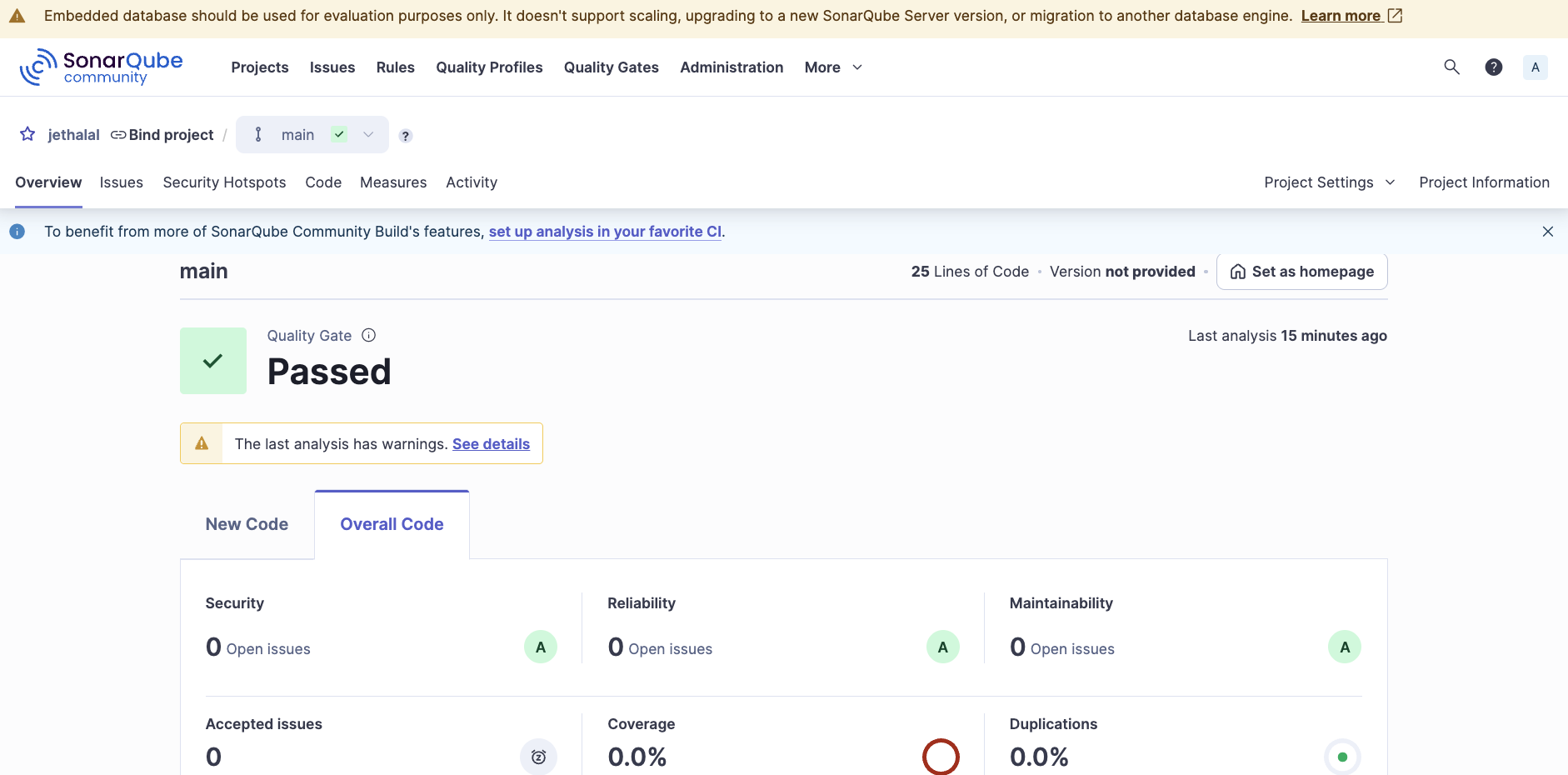
Copy and Paste the Token to you SonarProperties file:





Run the scanner with project;





**1. Lines of Code (LOC)**

* Actual number of **non-comment** and **non-blank** lines of code.

**2. Bugs**

* Defects in code that could cause runtime errors or incorrect behavior.
* Example: Null pointer dereference, array index out of bounds.

**3. Vulnerabilities**

* **Security-related issues** in the code that might be exploited.
* Example: SQL Injection, hardcoded passwords.

**4. Code Smells**

* Maintainability issues that **do not affect correctness** but make the code harder to understand and maintain.
* Example: Long methods, duplicated code, too many parameters.

**5. Security Hotspots**

* Code sections **that require manual review** to determine if there's a security risk.
* Not necessarily vulnerabilities unless confirmed.

**Quality Gates**

**6. Quality Gate**

* A set of **thresholds** defined on key metrics to decide whether a project passes or fails.
* Example: "Code coverage on new code must be > 80%".

**7. Passed/Failed**

* Indicates whether your project **met or failed** the conditions in the Quality Gate.

**Coverage**

* **Percentage of code lines tested by unit tests**.
* Formula: (Lines covered by tests / Total lines to cover) \* 100

**15. Line Coverage**

* % of lines covered by tests (subset of overall coverage).

**16. Branch Coverage**

* % of code branches (if/else, switch) covered by tests.

**Issues & Rules**

**27. Issues**

* Reported problems (bugs, vulnerabilities, code smells).

**28. Rules**

* Predefined or custom **code quality rules** used to analyze your code.

**29. Effort to Fix**

* Estimated time required to **resolve a specific issue**.