|  |  |
| --- | --- |
| **Name: Yukta Varma** | **Class: MCA Sem III** |
| **Roll No: A079** | **Batch: B2** |

**Practical:** Set up automatic code analysis for a GitHub repository Using SonarCloud, ensuring that a pull request triggers code quality check

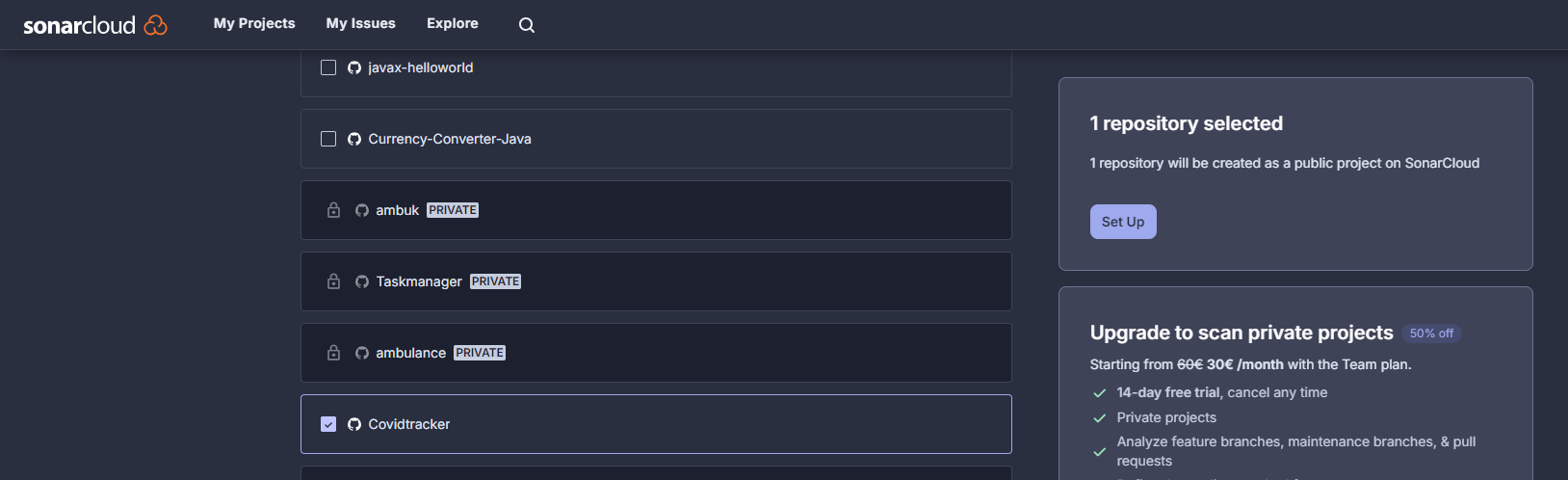
**Prerequisites**

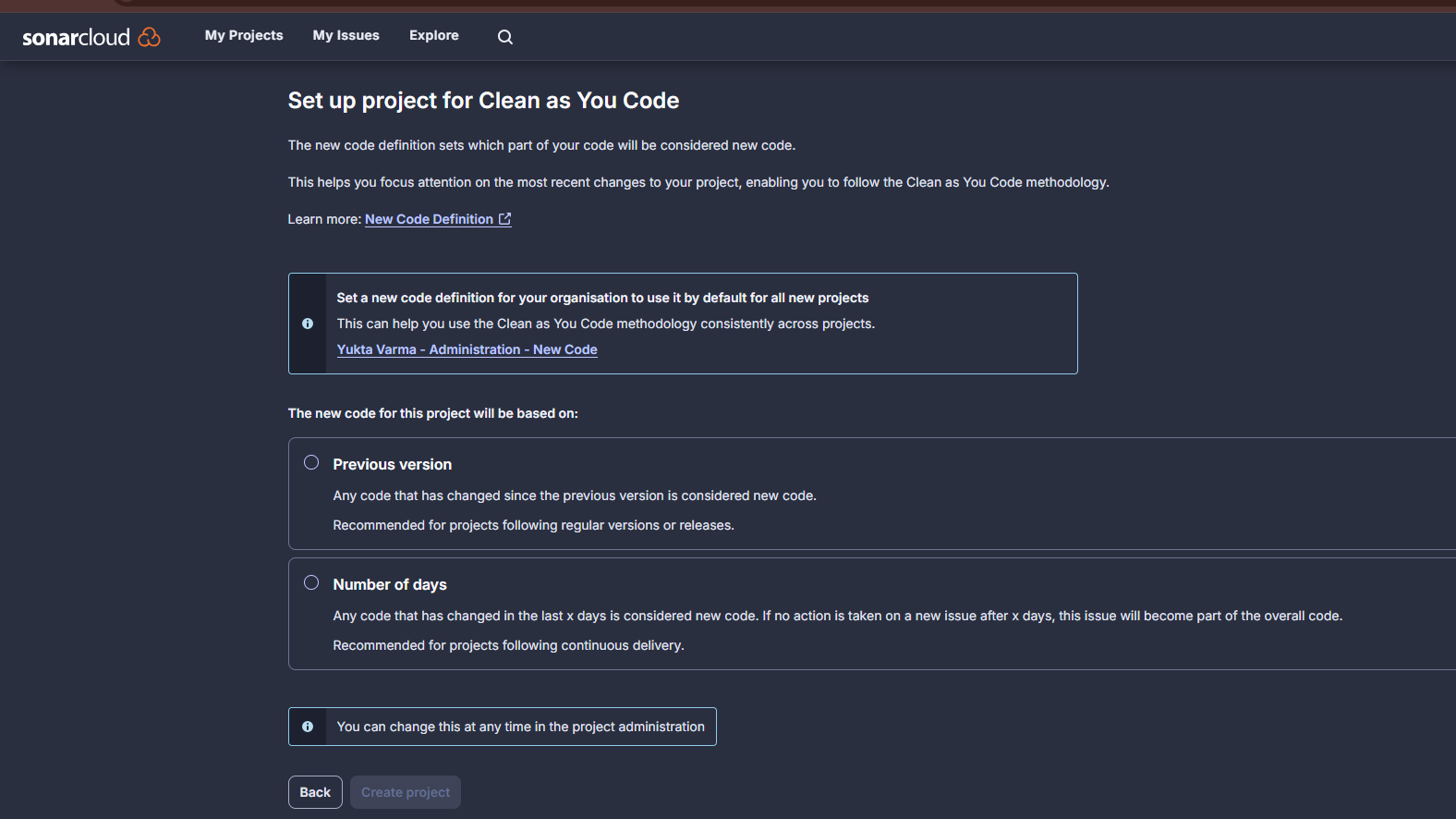
* **GitHub Account**: You need an account on GitHub with access to the repository you want to analyze.
* **SonarCloud Account**: Sign up for SonarCloud using your GitHub account.
* **Admin Permissions**: You must have admin permissions on the GitHub repository you want to integrate with SonarCloud.

### Step-by-Step Setup

#### **1. Create a SonarCloud Account**

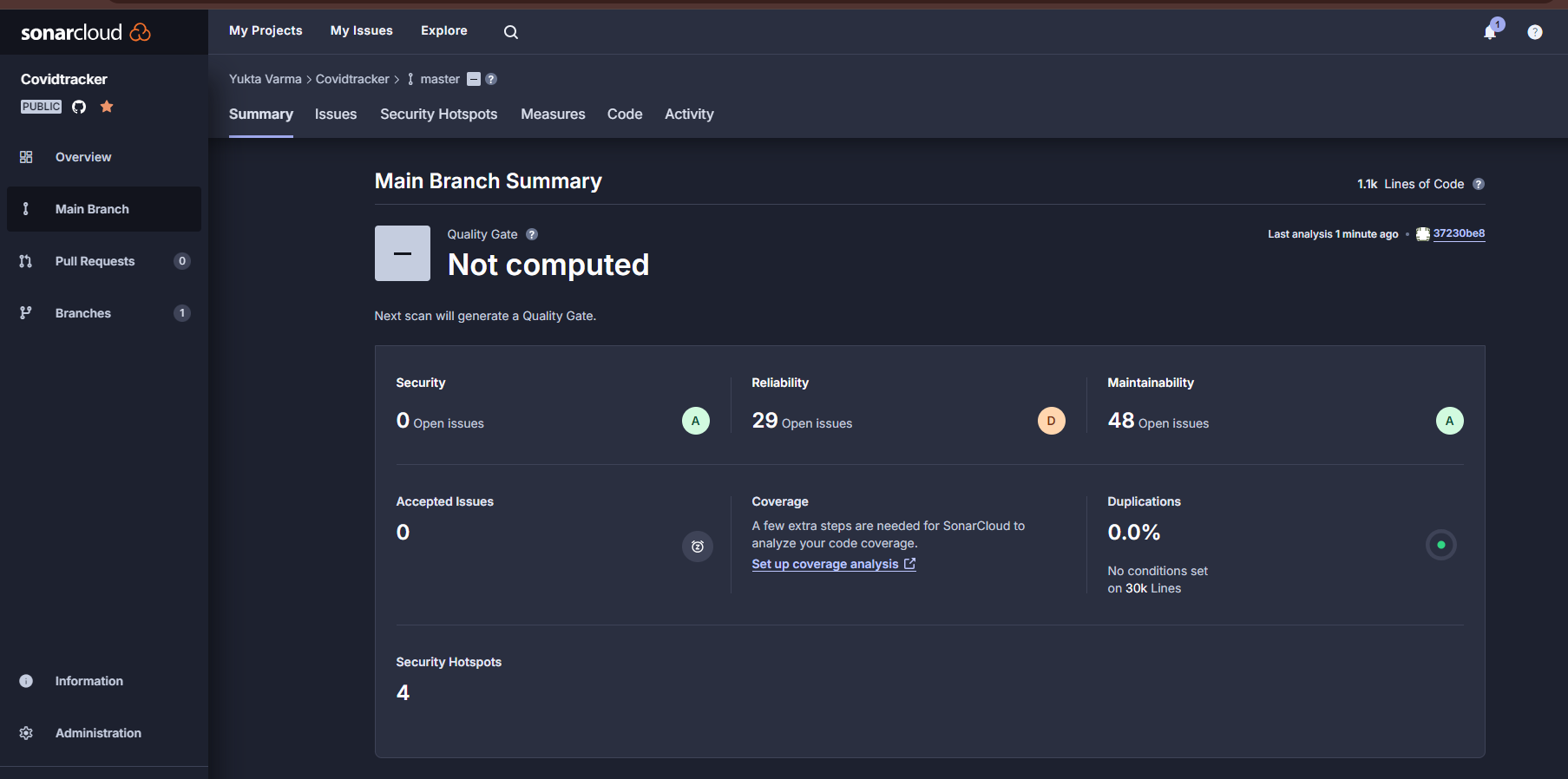
* **Go to SonarCloud**: Visit [SonarCloud](https://sonarcloud.io).
* **Sign Up**: Click on the "Sign up" button and choose the option to sign up using your GitHub account.
* **Grant Access**: After signing up, you will be prompted to authorize SonarCloud to access your GitHub repositories. Grant the necessary permissions.

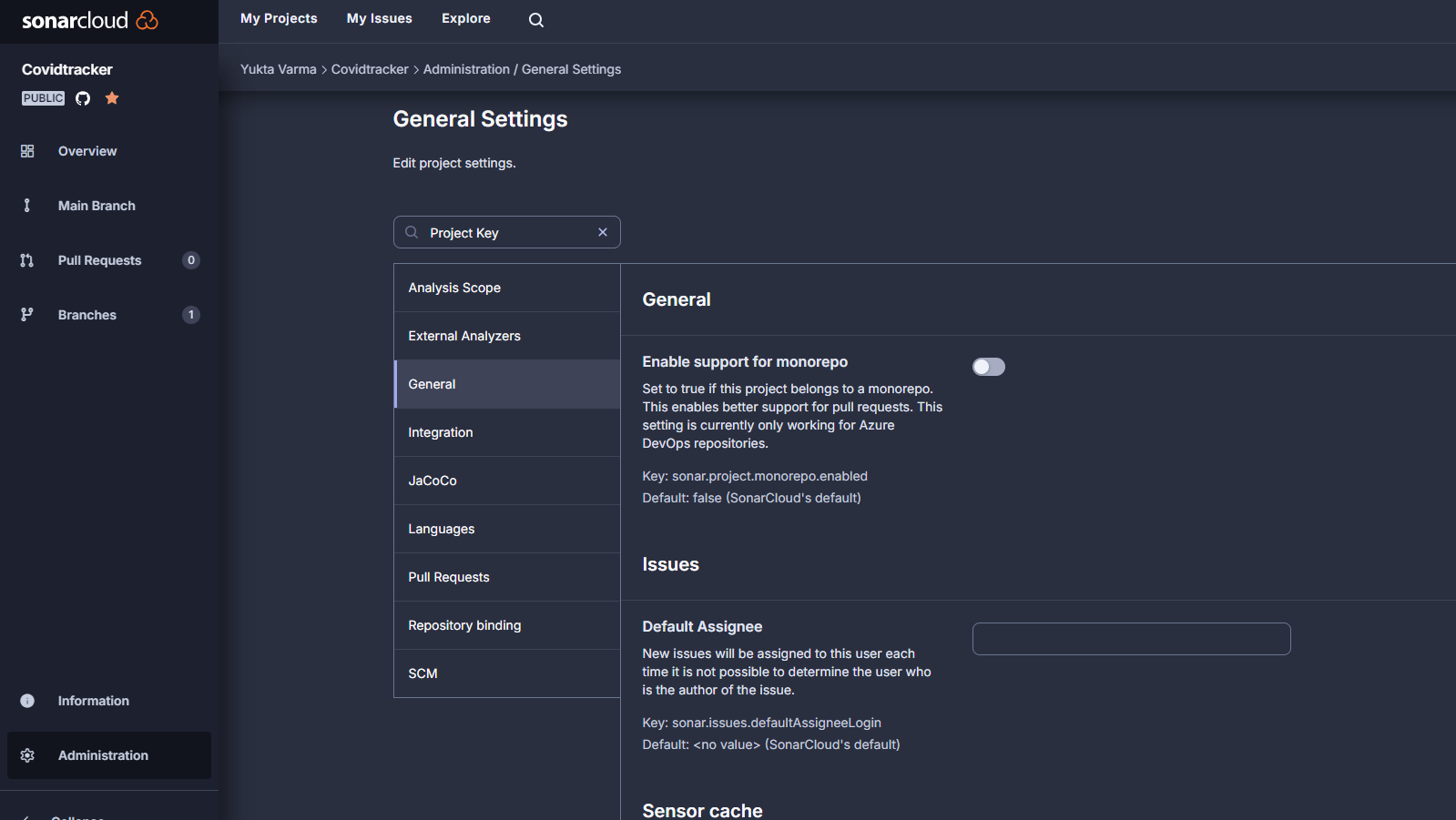


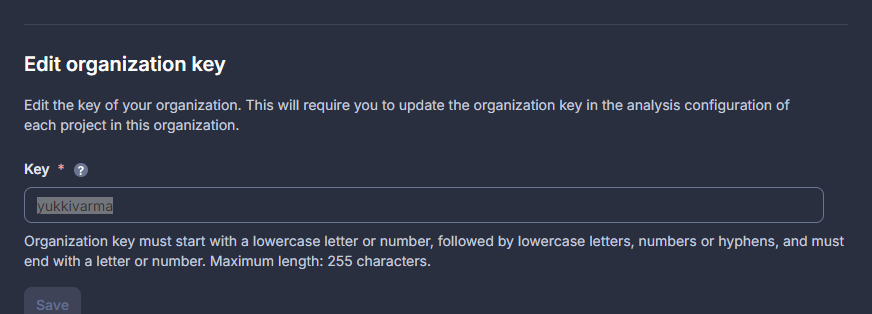


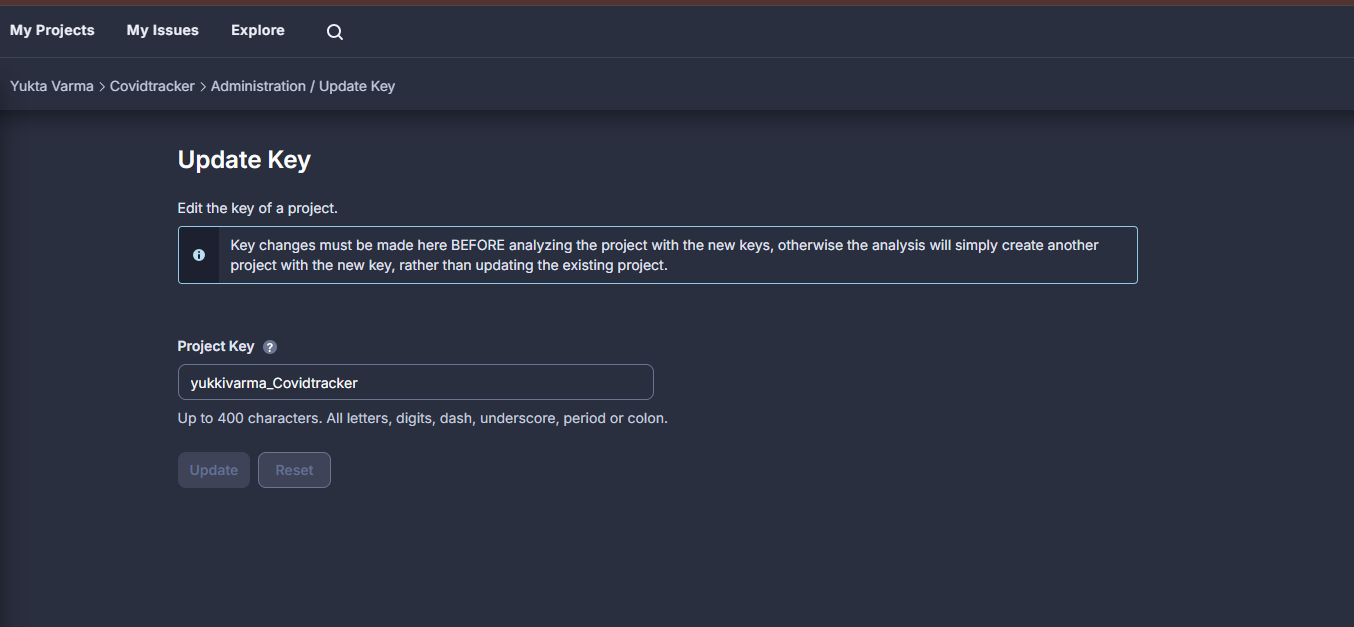
#### **2. Configure SonarCloud Project**

* **Open SonarCloud Dashboard**: After logging in, navigate to the SonarCloud dashboard.
* **Click on Projects**: Look for the "Projects" section on the sidebar.
* **Analyze New Project**: Click on the "Analyze new project" button.
* **Select Your Repository**: Choose GitHub as the organization and select the repository you want to analyze from the list.
* **Grant Access**: Follow the prompts to grant SonarCloud access to your repository.
* **Choose a Project Key**: You can choose a unique project key or use the one generated by SonarCloud by default. Make a note of this key as you will need it later.









yukkivarma\_Covidtracker

#### **3. Set Up SonarCloud Scanner**

* **Create sonar-project.properties File**:
  + In the root directory of your project, create a file named sonar-project.properties.
* **Add Configuration**:
  + Open sonar-project.properties in a text editor and add the following lines, replacing the placeholders with values:

sonar.projectKey= sonar.projectKey=yukkivarma\_Covidtracker

sonar.organization=sonar.organization=yukkivarma

# SonarCloud server URL

sonar.host.url=https://sonarcloud.io

# Define what should be analyzed

sonar.sources=.

# Optional: path to test files

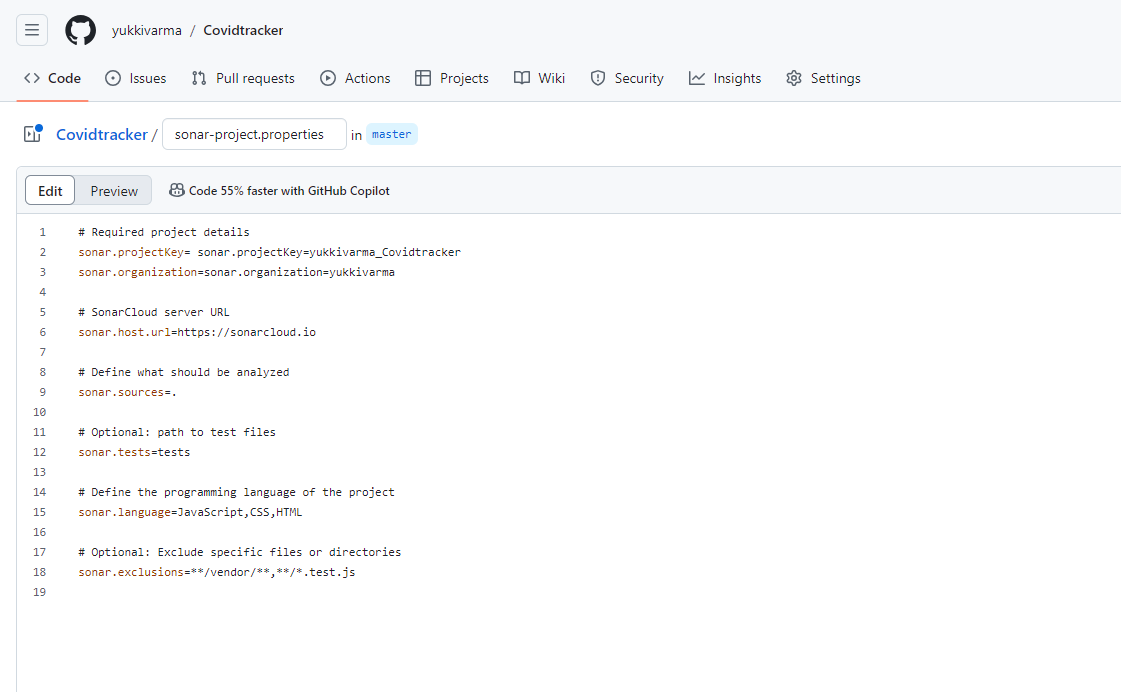
sonar.tests=tests

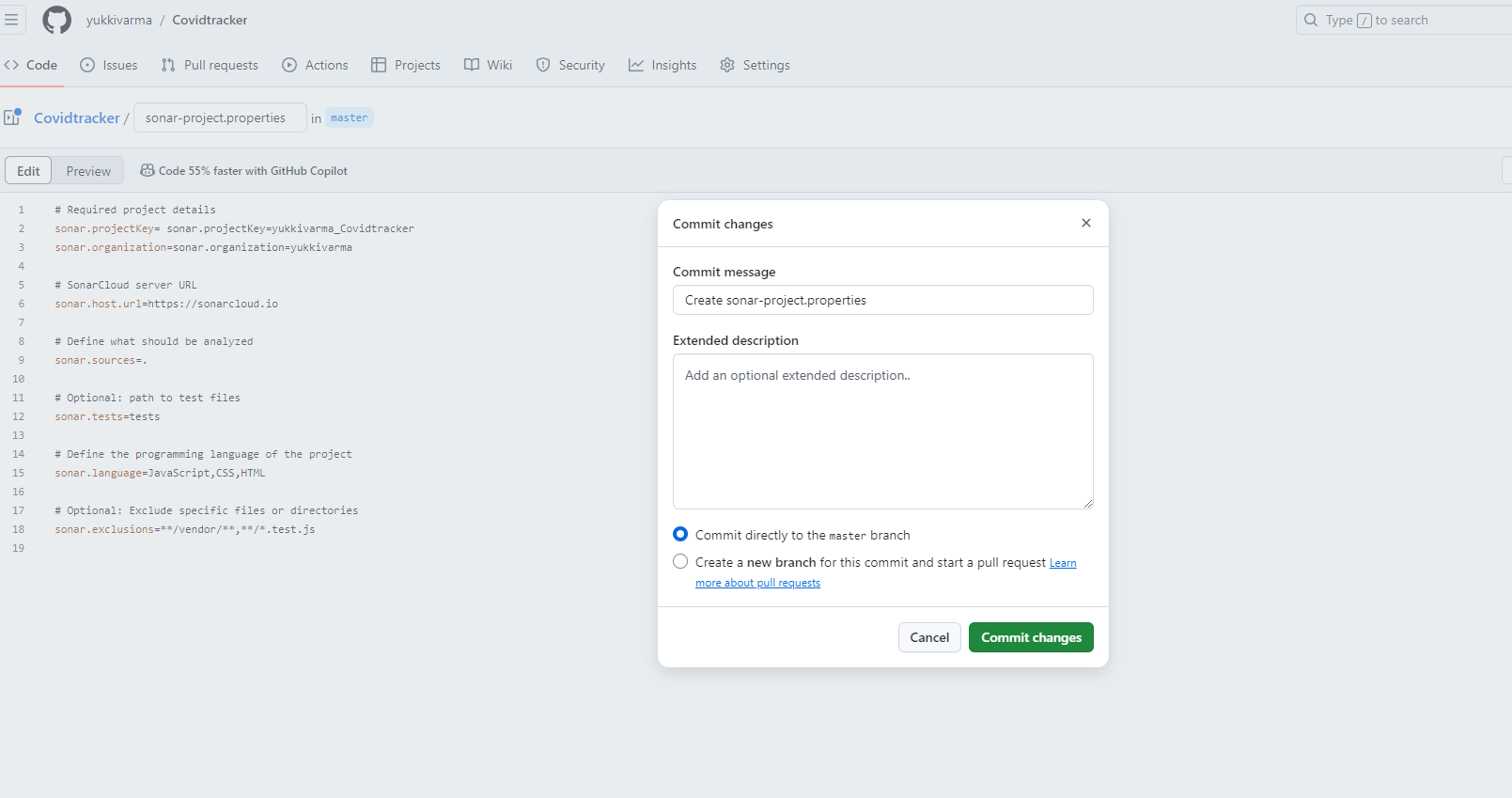
# Define the programming language of the project

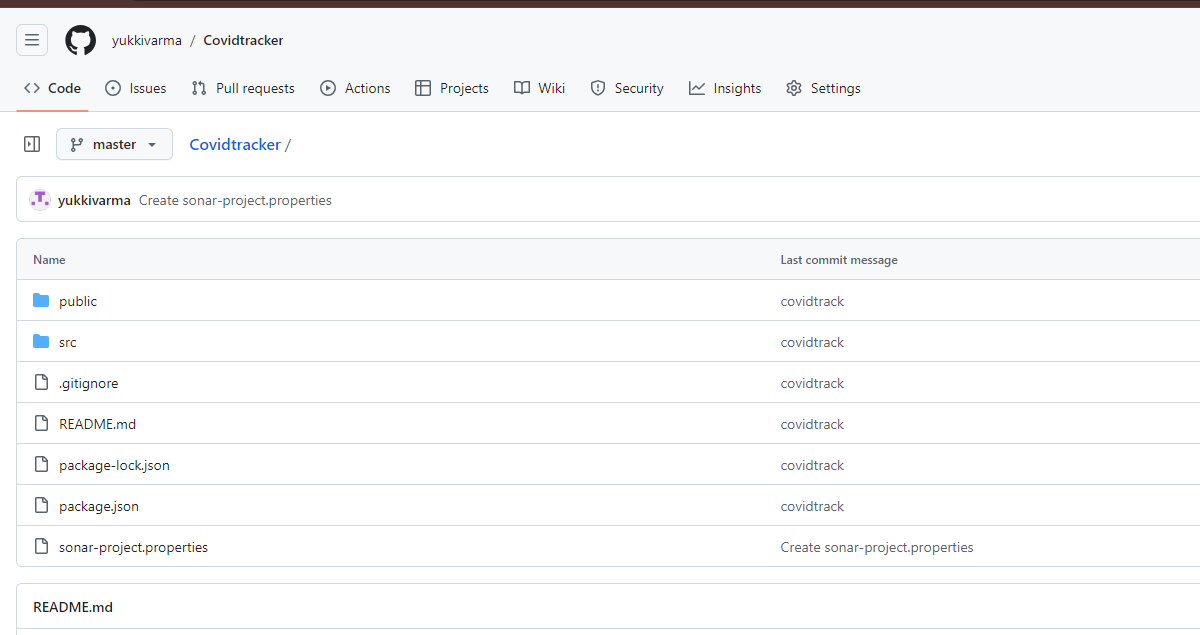
sonar.language=JavaScript,CSS,HTML

# Optional: Exclude specific files or directories

sonar.exclusions=\*\*/vendor/\*\*,\*\*/\*.test.js

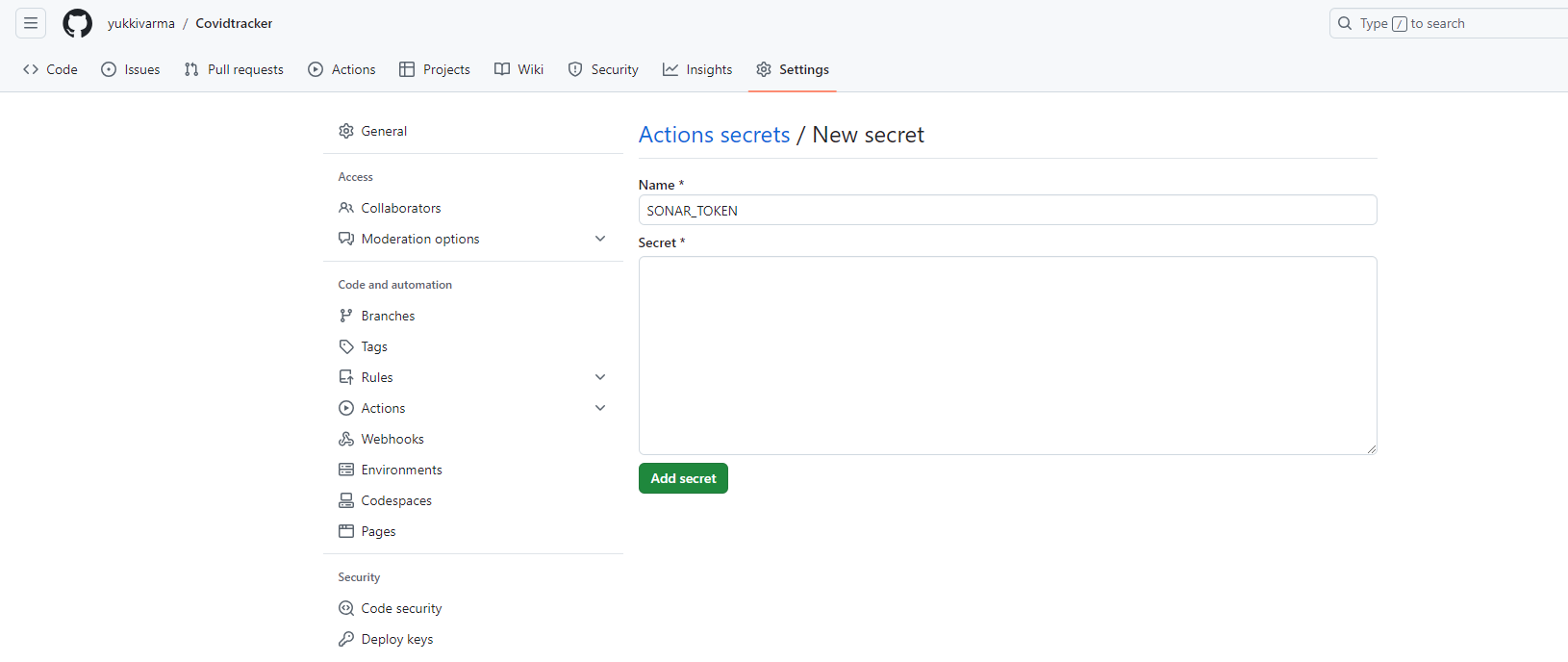


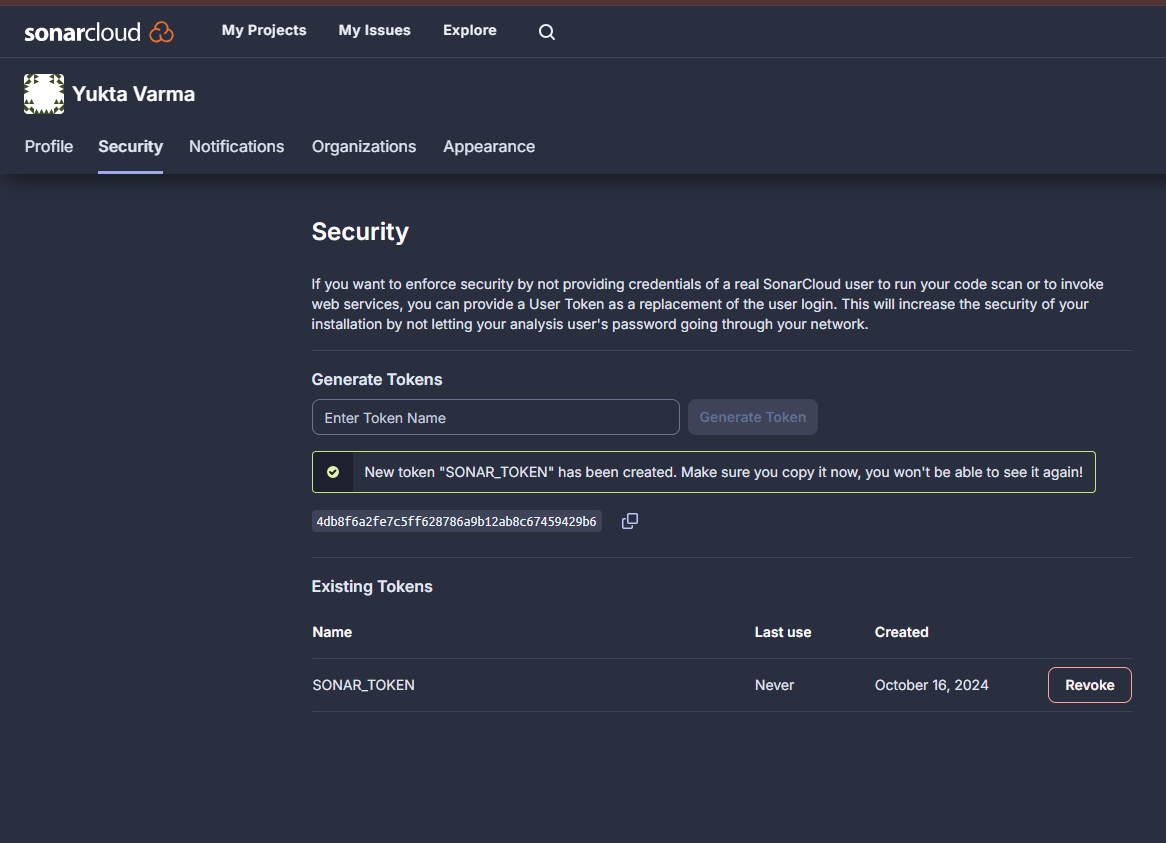


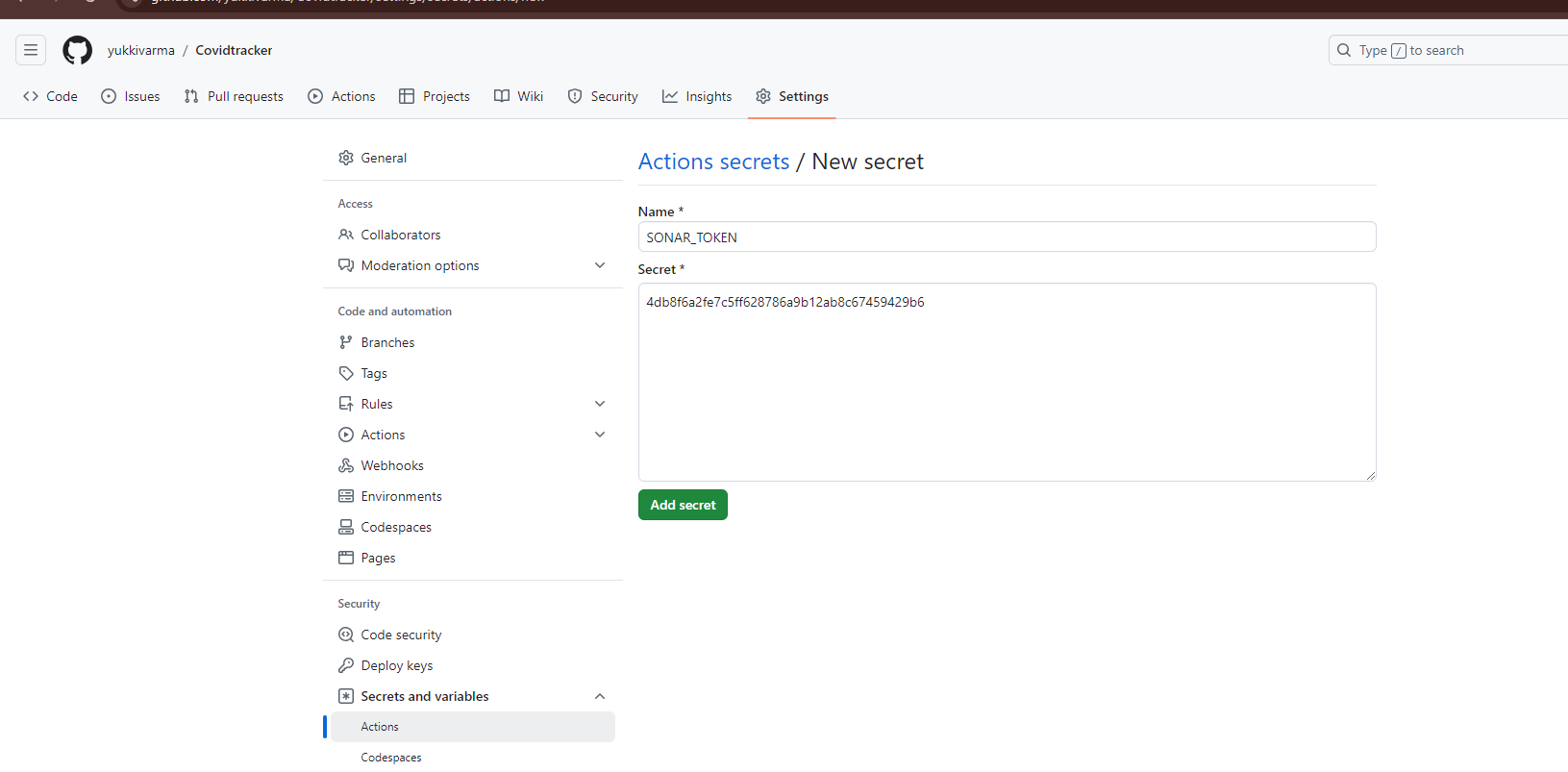


#### **4. Integrate SonarCloud with GitHub Actions**

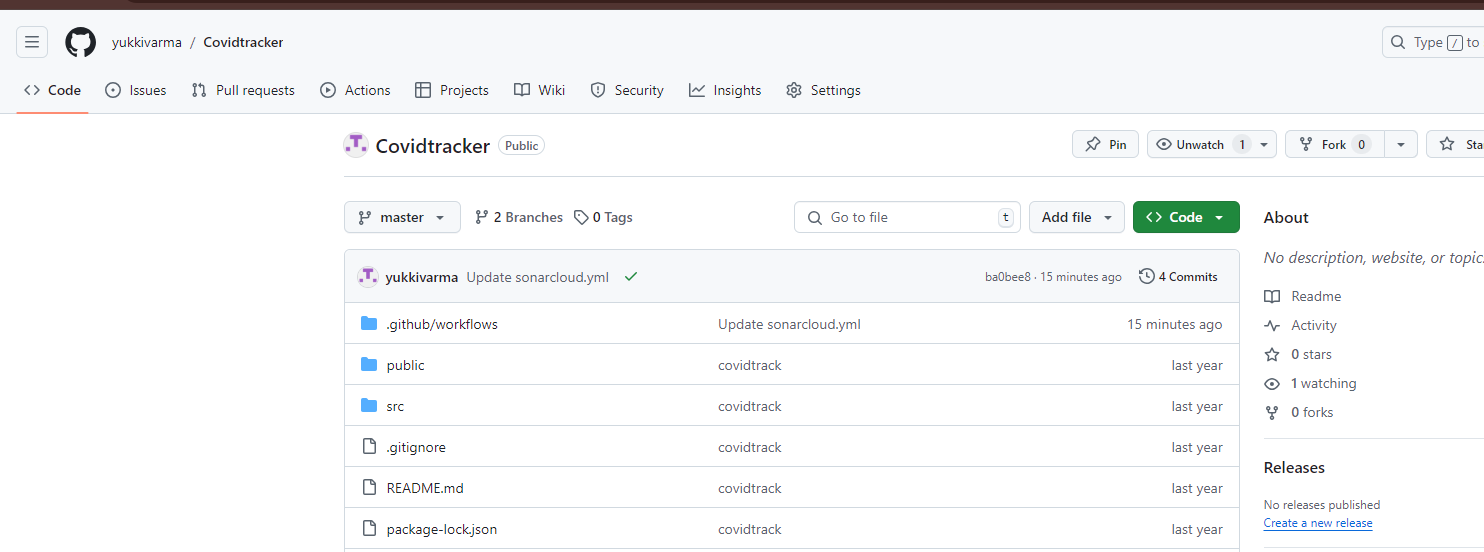
* **Navigate to Your GitHub Repository**: Open your repository on GitHub.
* **Go to Settings**:
  + Click on the "Settings" tab of your repository.
* **Add Secrets**:
  + In the left sidebar, click on **Secrets and variables** and then **Actions**.
  + Click on **New repository secret**.
    - **Name**: SONAR\_TOKEN
    - **Value**: Paste your SonarCloud token, which you can find in your SonarCloud account settings under "My Account" > "Security" > "Tokens". Create a new token if you haven't done so yet.







* **Create GitHub Actions Workflow**:
  + In your local repository, create the directory structure .github/workflows/.
  + Inside the workflows directory, create a file named sonarcloud.yml.



* **Add GitHub Actions Configuration**:
  + Open sonarcloud.yml and add the following configuration:

name: SonarCloud Analysis

on:

pull\_request:

branches:

- main

push:

branches:

- main

jobs:

sonarcloud:

name: SonarCloud Scan

runs-on: ubuntu-latest

steps:

- name: Checkout code

uses: actions/checkout@v3

- name: Set up JDK 11 (if applicable, adjust for your language)

uses: actions/setup-java@v3

with:

java-version: '11'

- name: Cache SonarCloud packages

uses: actions/cache@v3

with:

path: ~/.sonar/cache

key: ${{ runner.os }}-sonar

- name: SonarCloud Scan

env:

SONAR\_TOKEN: ${{ secrets.SONAR\_TOKEN }} # Fetches the SonarCloud token from GitHub secrets

run: |

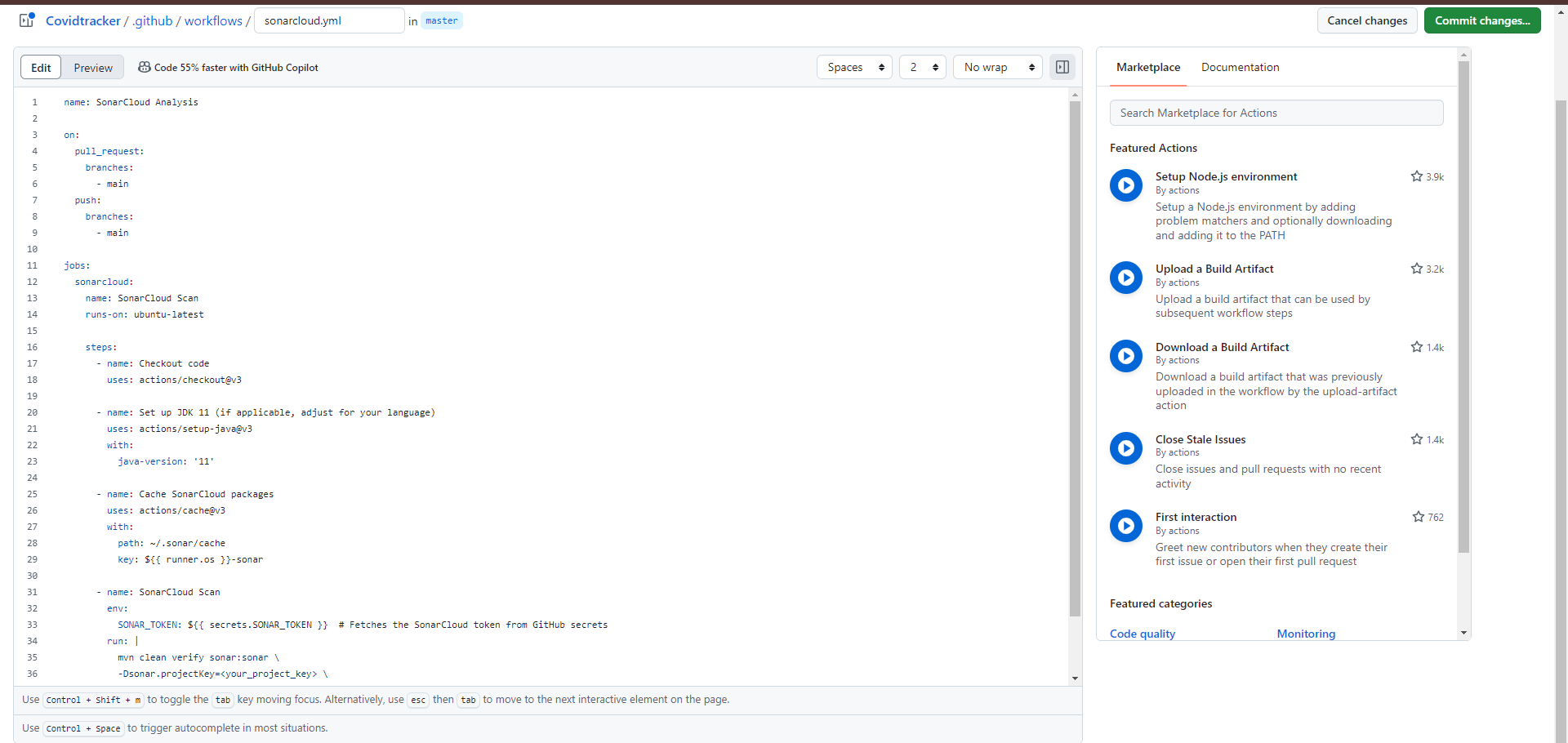
mvn clean verify sonar:sonar \

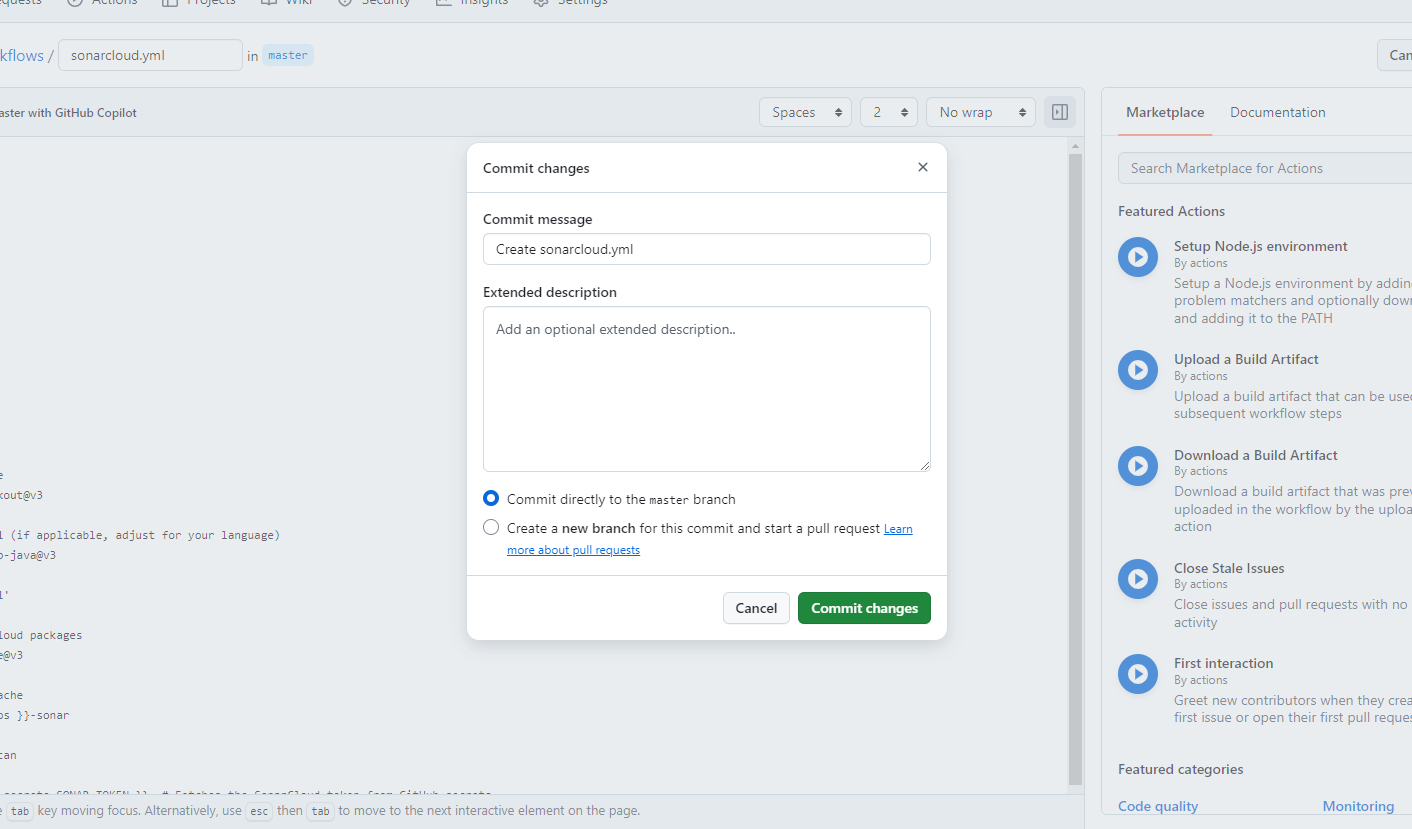
-Dsonar.projectKey= yukkivarma\_Covidtracker \

-Dsonar.organization=yukkivarma \

-Dsonar.host.url=https://sonarcloud.io \

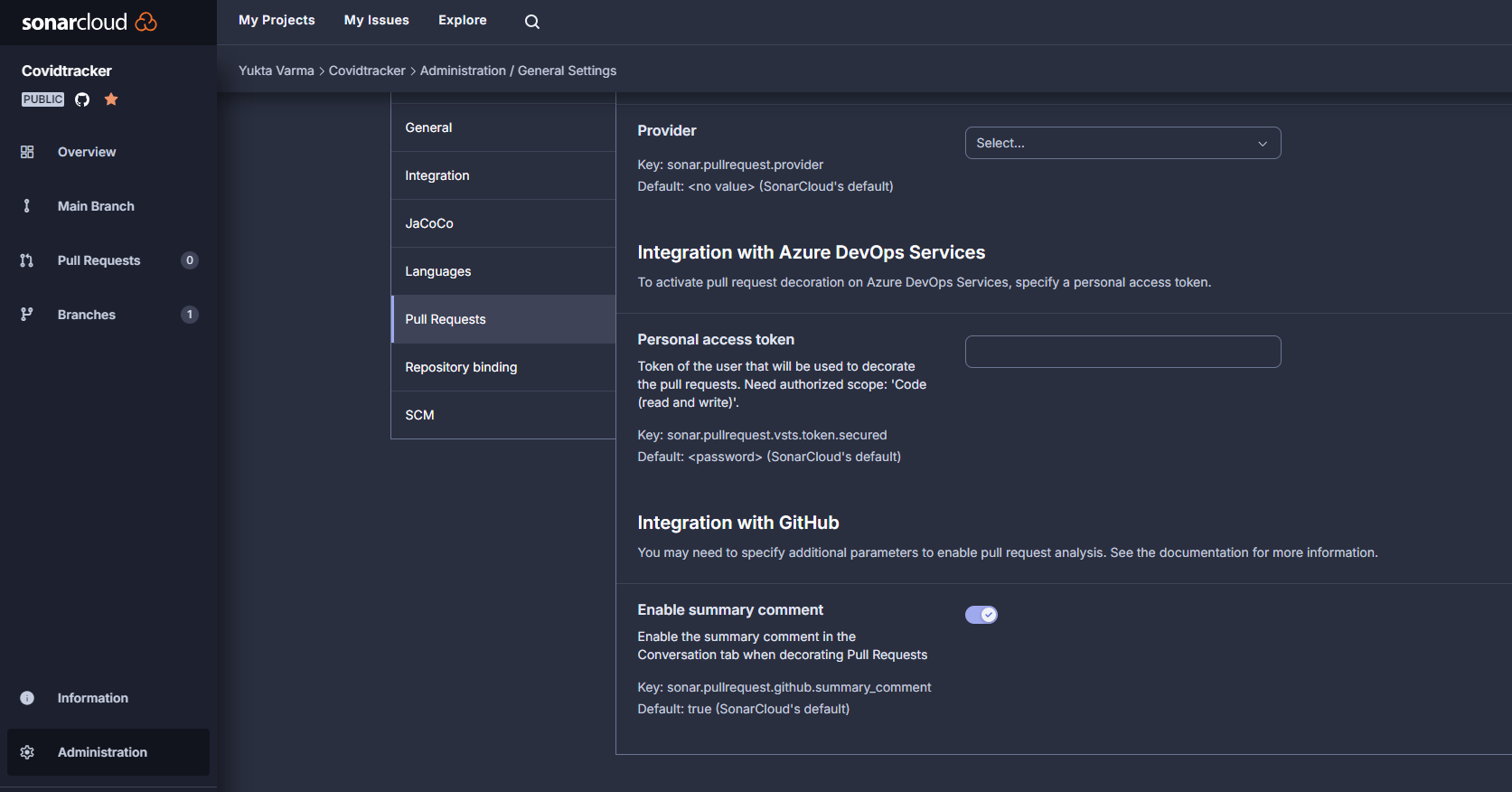
-Dsonar.login=${{ secrets.SONAR\_TOKEN }}





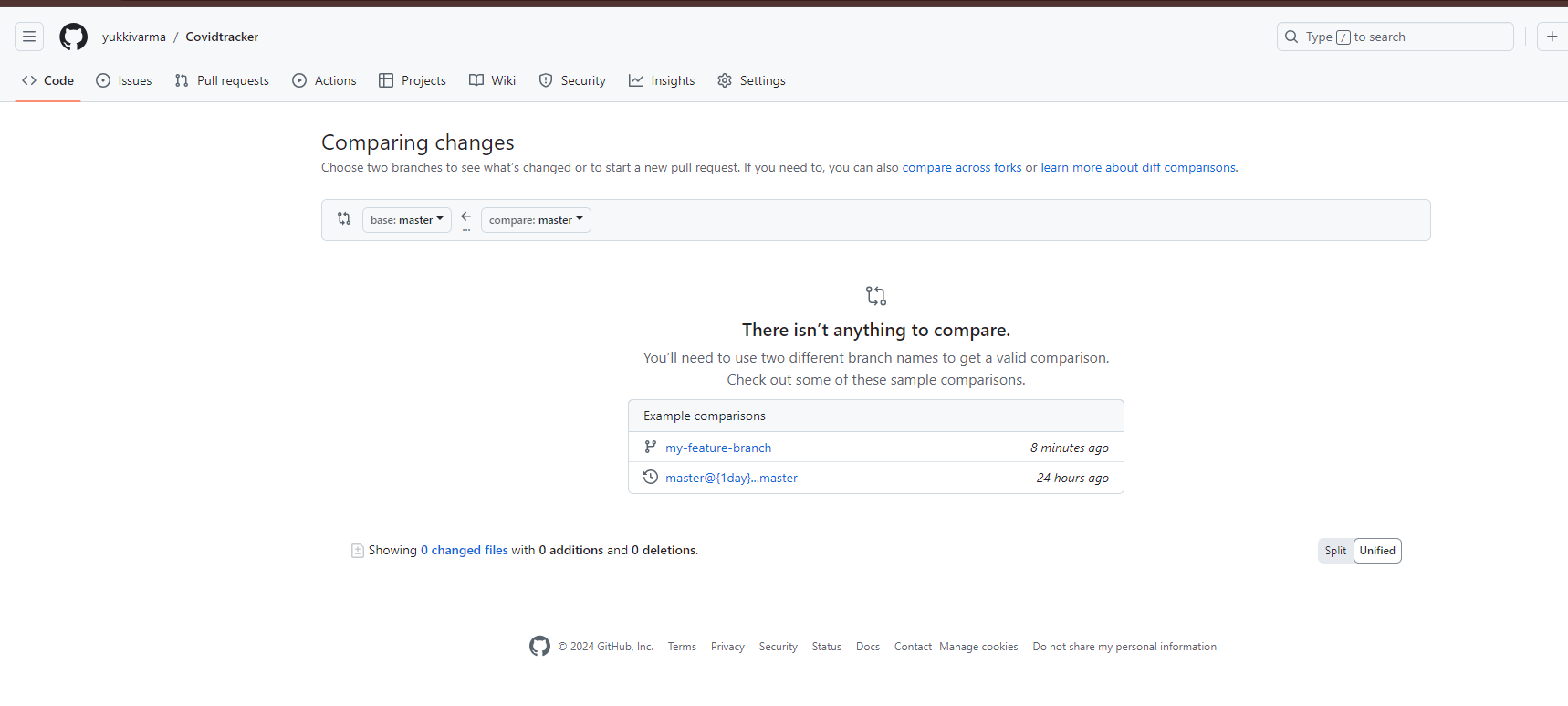
### Step 5: Configure Pull Request Decoration in SonarCloud

1. **Navigate to SonarCloud**:
   * Go to [SonarCloud](https://sonarcloud.io).
2. **Select Your Project named CovidTracker**:
   * From the SonarCloud dashboard, locate and click on the project you added from GitHub.
3. **Configure Pull Request Decoration**:
   * In the project overview, look for the **Administration** tab. Click on it.
   * In the left sidebar, click on **Pull Requests**.
   * You will see options related to Pull Request Decoration. Make sure the **Pull Request Decoration** feature is enabled. If it's already enabled, you should see a confirmation message or indicator.
   * Check for any additional settings that might be relevant to your integration with GitHub and ensure they are correctly set.
4. **Integration with GitHub**:
   * SonarCloud should automatically be configured to integrate with GitHub if you've set it up following the previous steps. If you encounter issues, you may need to revisit the integration settings to ensure everything is connected properly.
5. **Save Changes**:
   * If you made any changes or enabled Pull Request Decoration, make sure to click the **Save** button to apply your settings.

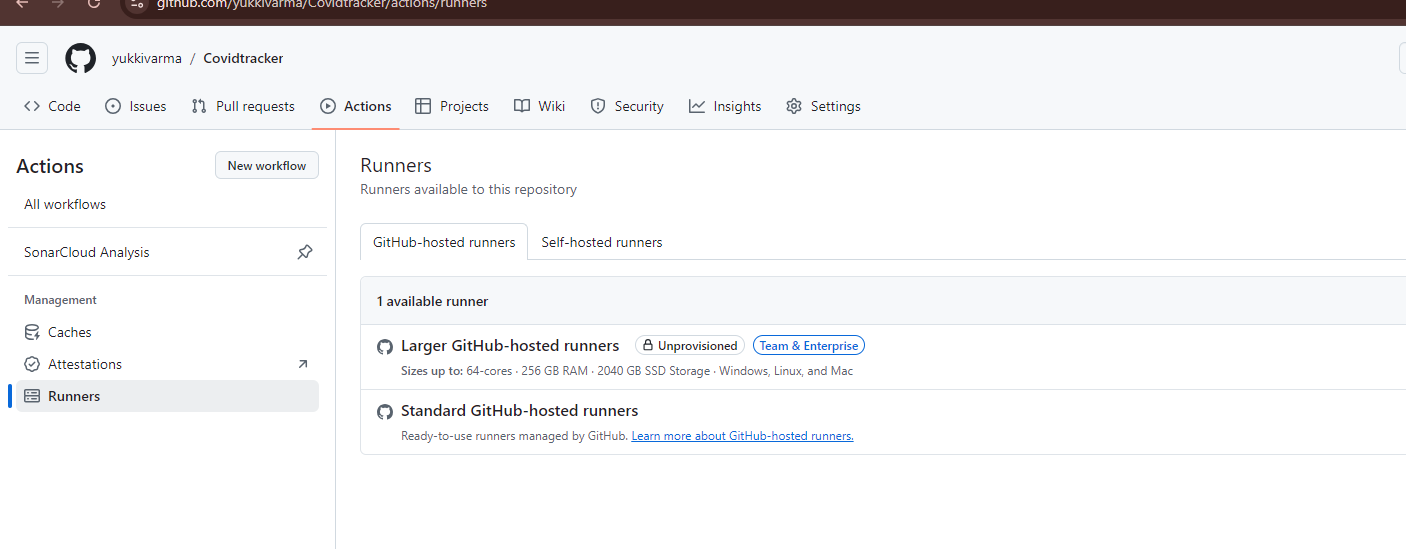


**Step 6: Trigger the Analysis via Pull Request**

1. **Create a Pull Request**:
   * Go back to your GitHub repository where you set up the workflow.
   * Navigate to the **Pull Requests** tab.
   * Click on the **New pull request** button.
   * Select the branch you want to merge into the main branch and create the pull request. You can include a descriptive title and comment that provides context for the changes.



1. **GitHub Actions Triggers**:
   * Once you create the pull request, GitHub Actions will automatically trigger the workflow you set up earlier (sonarcloud.yml).
   * You can monitor the progress of the workflow by clicking on the **Actions** tab in your GitHub repository. You should see the workflow running for your pull request



1. **View SonarCloud Feedback in the PR**:
   * After the SonarCloud analysis completes, it will post comments directly on your pull request.
   * Check the **Conversation** tab of your pull request. You will see SonarCloud comments that provide insights into:
     + **Code Quality**: Ratings and metrics about your code.
     + **Code Smells**: Suggestions for improving code maintainability.
     + **Bugs**: Any issues found in the code that could lead to runtime errors.
     + **Vulnerabilities**: Security issues that need to be addressed.
     + **Code Coverage**: Information about the percentage of code covered by tests.
2. **Review the Feedback**:
   * Review the comments and feedback provided by SonarCloud. You can address any issues highlighted and make additional commits to your branch.
   * After addressing the feedback, push the changes to the same branch. This will trigger the SonarCloud analysis again, and you’ll see updated feedback in the pull request.
3. **Merge the Pull Request**:
   * Once you have addressed all comments and are satisfied with the code quality, you can proceed to merge the pull request into the main branch.

