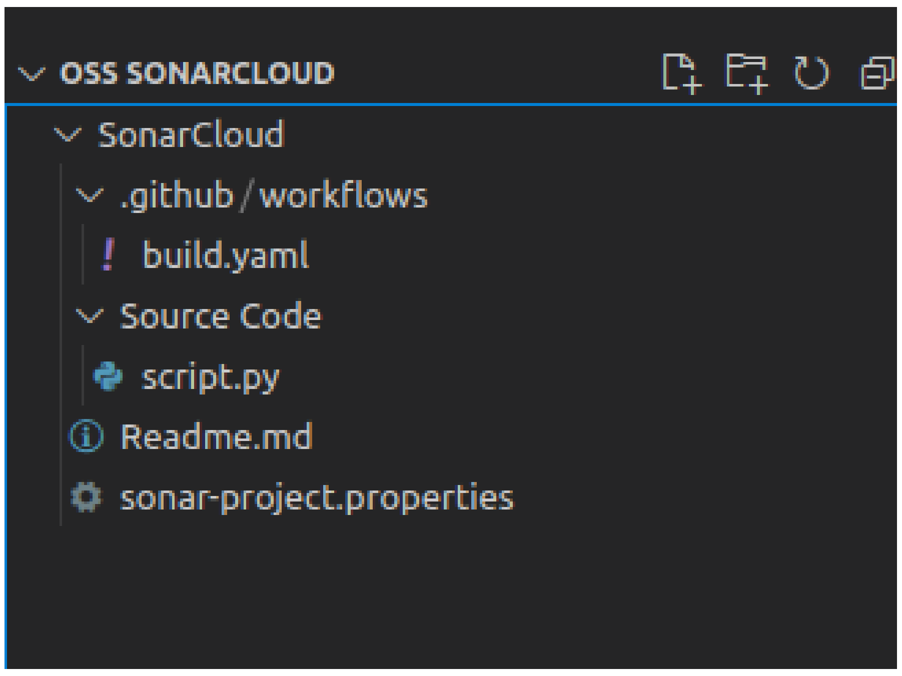
Experiment 16. SonarCloud

Experiment No: 16 Demonstrate the use/features of online Project Management tool: "SONARcloud" for managing projects. Demonstrate the one foss project for code coverage, Detect Bugs & Vulnerabilities, Review Security Hotspots, Track Code Smells & fix your Technical Debt. Show the Code Quality Metrics & History. Compare it with other **Project Management tool** (on answer sheet)

▼ Setup of Sonar Cloud

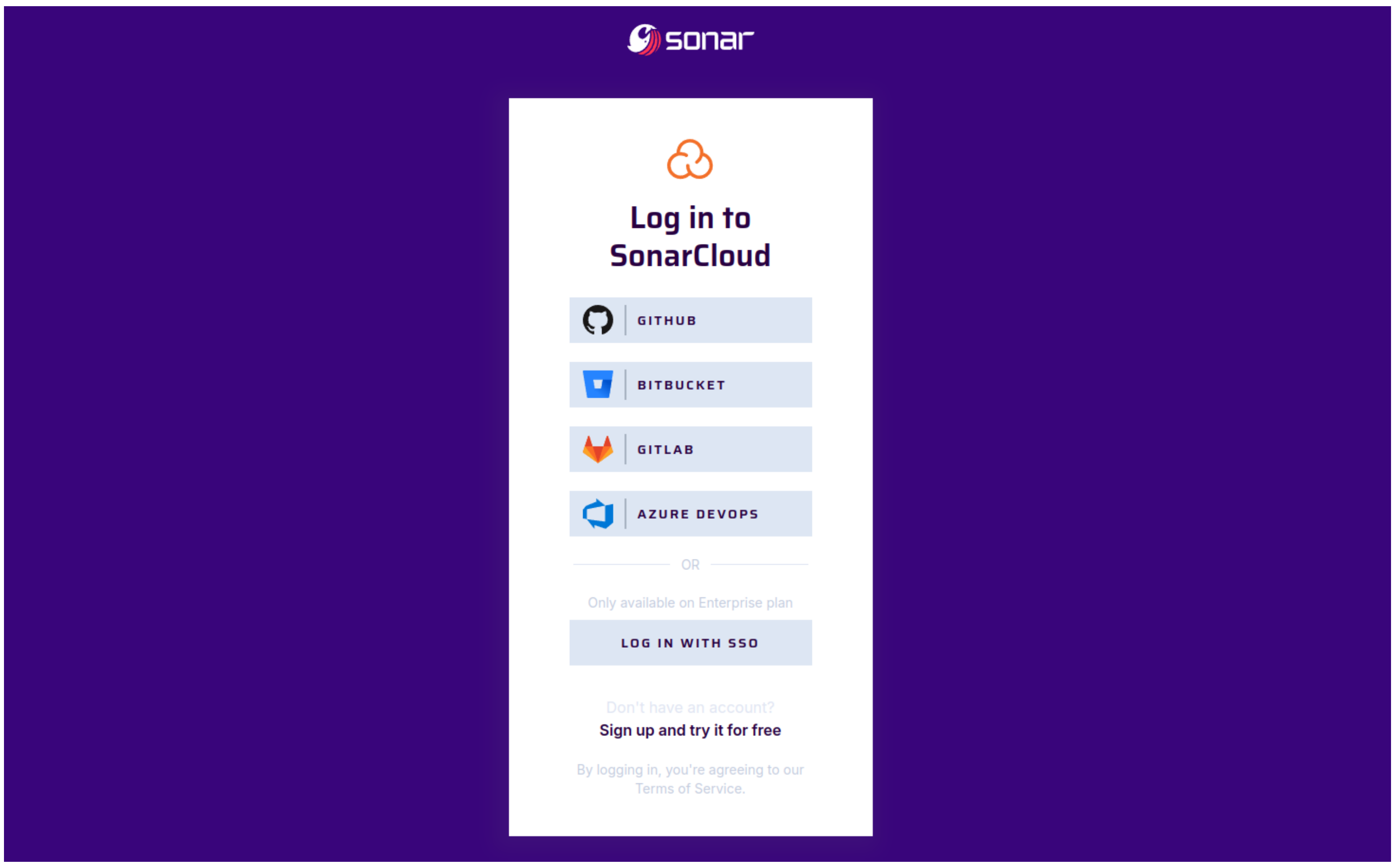
- Github repo for codes : https://github.com/Saurav1928/SonarCloud

1. create a github empty repo
2. clone the repo in VS code and creat a folder structure as follow



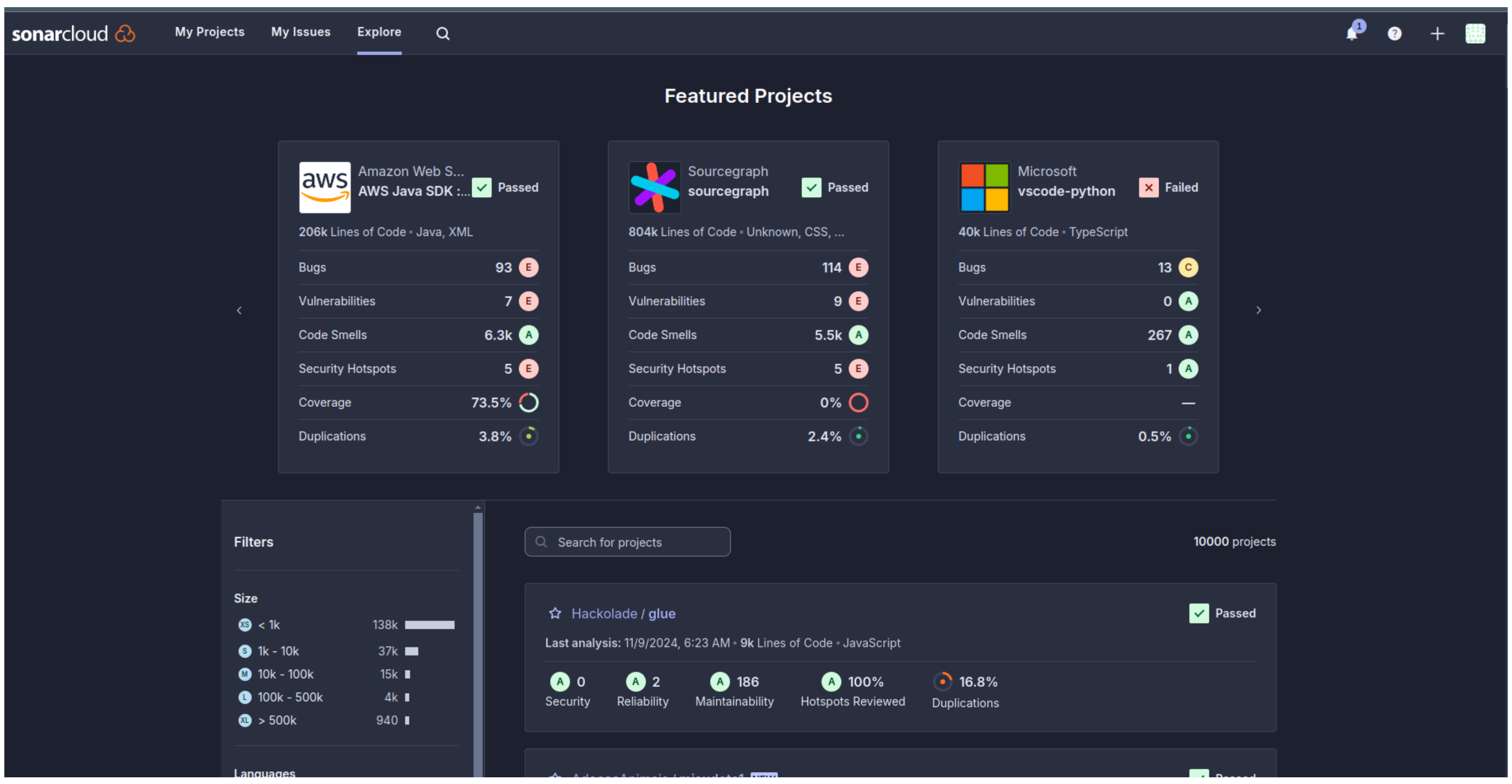
```
OSS SONARCLOUD
  SonarCloud
    .github / workflows
      build.yaml
    Source Code
      script.py
    Readme.md
    sonar-project.properties
```

3. Login to https://sonarcloud.io/login?return_to=%2Fexplore%2Fprojects



sonar
Log in to SonarCloud
GITHUB
BITBUCKET
GITLAB
AZURE DEVOPS
OR
Only available on Enterprise plan
LOG IN WITH SSO
Don't have an account?
Sign up and try it for free
By logging in, you're agreeing to our Terms of Service.

4. Click on Github
5. Hope you can see (great you have logged in successfully)



Featured Projects

Amazon Web S... AWS Java SDK :... ✓ Passed		Sourcegraph sourcegraph ✓ Passed		Microsoft vscode-python ✗ Failed	
206k Lines of Code • Java, XML		804k Lines of Code • Unknown, CSS, ...		40k Lines of Code • TypeScript	
Bugs	93 E	Bugs	114 E	Bugs	13 C
Vulnerabilities	7 E	Vulnerabilities	9 E	Vulnerabilities	0 A
Code Smells	6.3k A	Code Smells	5.5k A	Code Smells	267 A
Security Hotspots	5 E	Security Hotspots	5 E	Security Hotspots	1 A
Coverage	73.5%	Coverage	0%	Coverage	—
Duplications	3.8%	Duplications	2.4%	Duplications	0.5%

Filters

Size

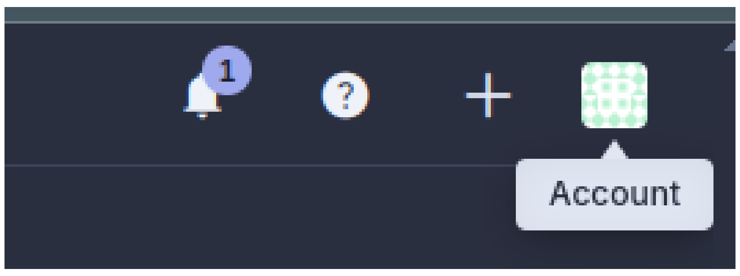
XS < 1k	138k	
S 1k - 10k	37k	
M 10k - 100k	15k	
L 100k - 500k	4k	
XL > 500k	940	

Languages

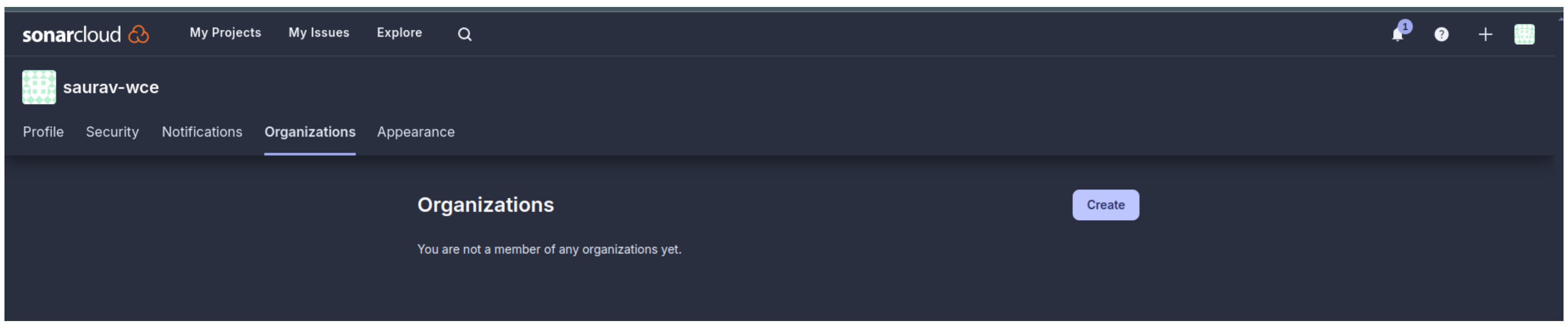
Search for projects — 10000 projects

Hackolade / glue ✓ Passed
Last analysis: 11/9/2024, 6:23 AM • 9k Lines of Code • JavaScript
A 0 Security · A 2 Reliability · A 186 Maintainability · A 100% Hotspots Reviewed · 16.8% Duplications

6. Click on Account



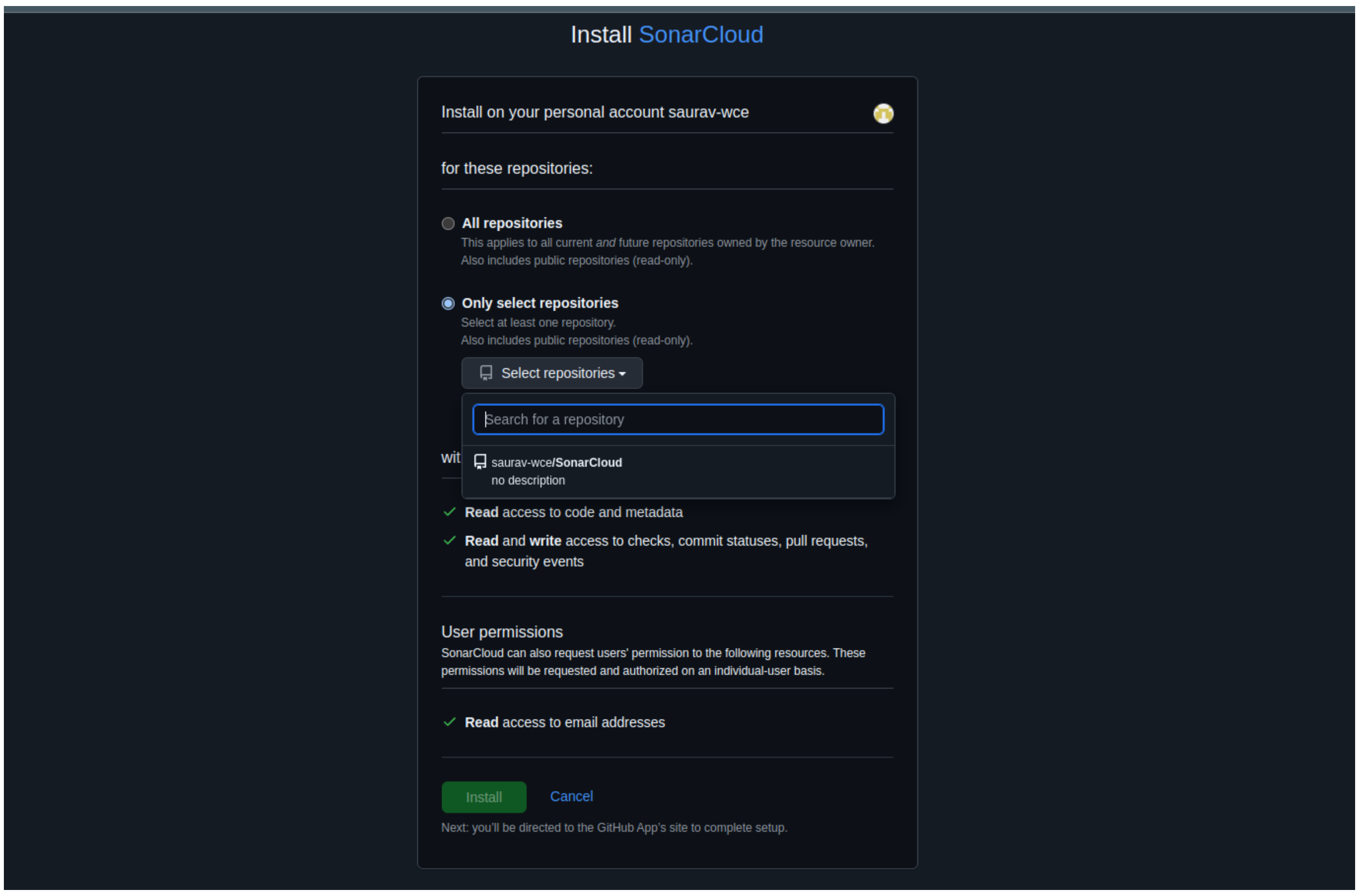
7. Click on organistion tab



Organizations

You are not a member of any organizations yet.

8. create
9. **Import from a DevOps platform**
10. choose 2nd option : Only Select repos. → select the repo which we created initially.


Install SonarCloud

Install on your personal account saurav-wce

for these repositories:

All repositories
This applies to all current *and* future repositories owned by the resource owner.
Also includes public repositories (read-only).

[✓ Only select repositories]
Select at least one repository.
Also includes public repositories (read-only).

Select repositories

Search for a repository

saurav-wce/**SonarCloud**
no description

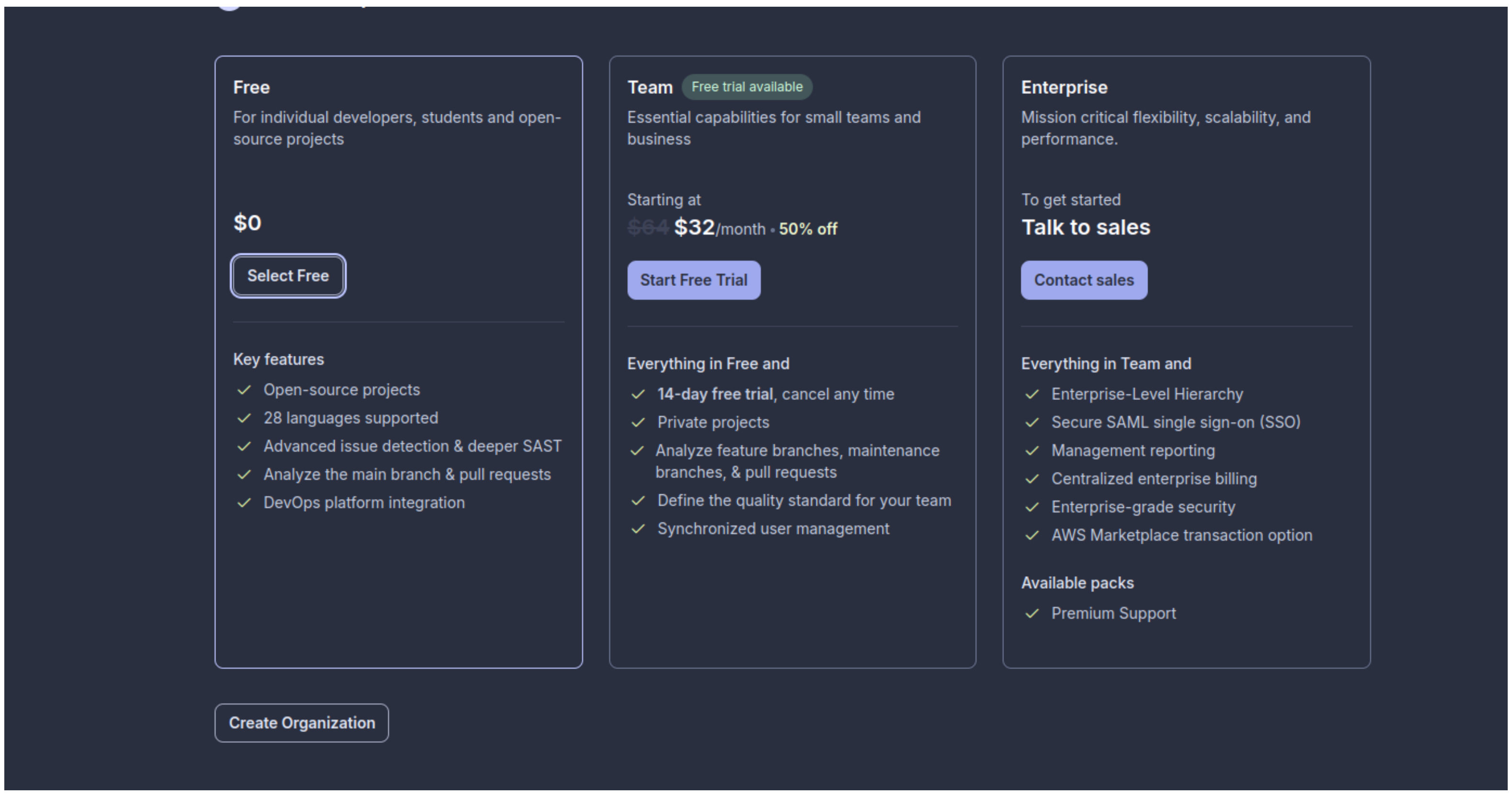
✓ **Read** access to code and metadata
✓ **Read** and **write** access to checks, commit statuses, pull requests, and security events

User permissions
SonarCloud can also request users' permission to the following resources. These permissions will be requested and authorized on an individual-user basis.

✓ **Read** access to email addresses

Next: you'll be directed to the GitHub App's site to complete setup.

11. click on Install button
12. keep the name and key as it is , select free


Free
For individual developers, students and open-source projects

$0

Key features
✓ Open-source projects
✓ 28 languages supported
✓ Advanced issue detection & deeper SAST
✓ Analyze the main branch & pull requests
✓ DevOps platform integration

Team Free trial available
Essential capabilities for small teams and business

Starting at
~~$64~~ $32/month · 50% off

Everything in Free and
✓ **14-day free trial**, cancel any time
✓ Private projects
✓ Analyze feature branches, maintenance branches, & pull requests
✓ Define the quality standard for your team
✓ Synchronized user management

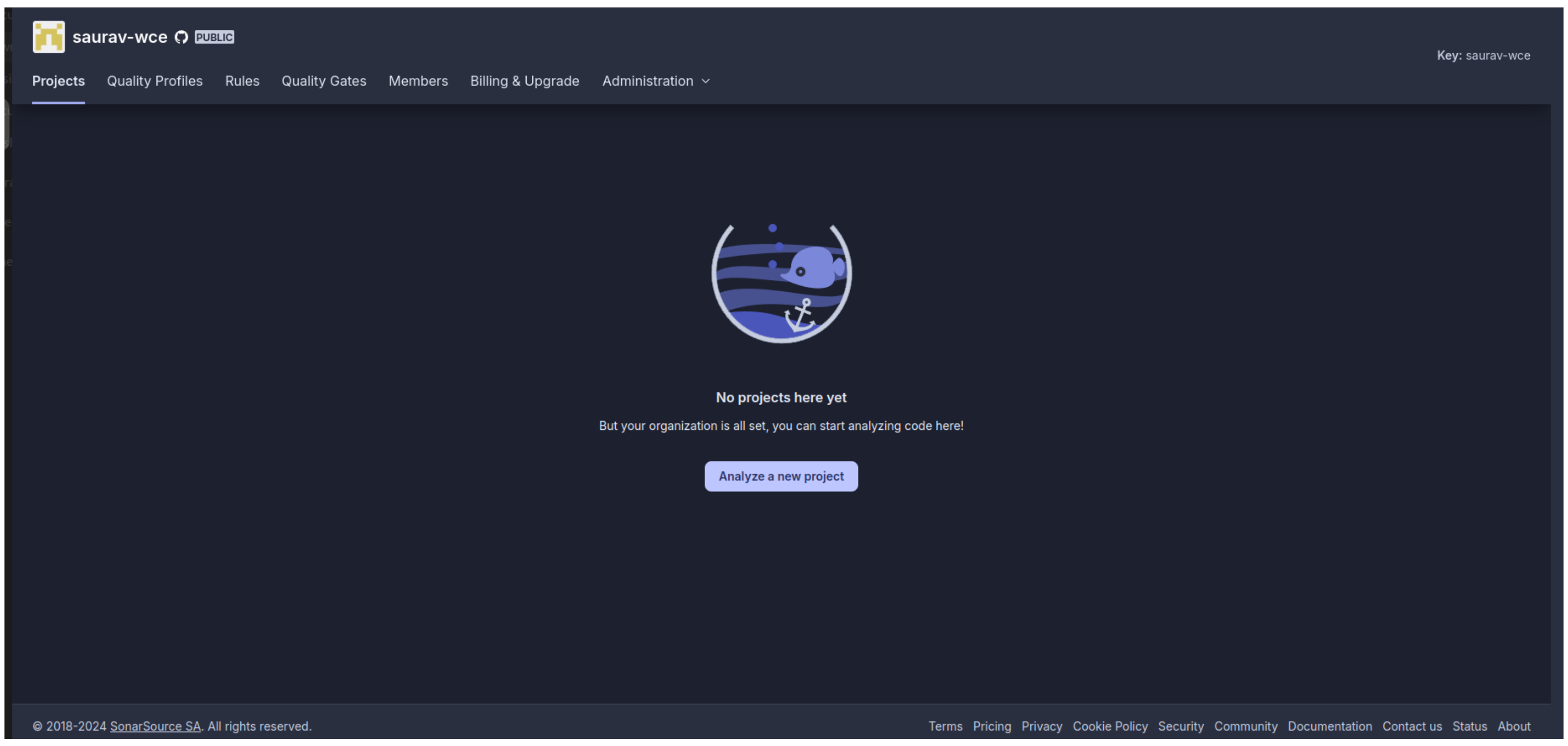
Enterprise
Mission critical flexibility, scalability, and performance.

To get started
Talk to sales

Everything in Team and
✓ Enterprise-Level Hierarchy
✓ Secure SAML single sign-on (SSO)
✓ Management reporting
✓ Centralized enterprise billing
✓ Enterprise-grade security
✓ AWS Marketplace transaction option

Available packs
✓ Premium Support

13. create organisation (this may take 30 sec to create)
14. If you can see following window, then good to go

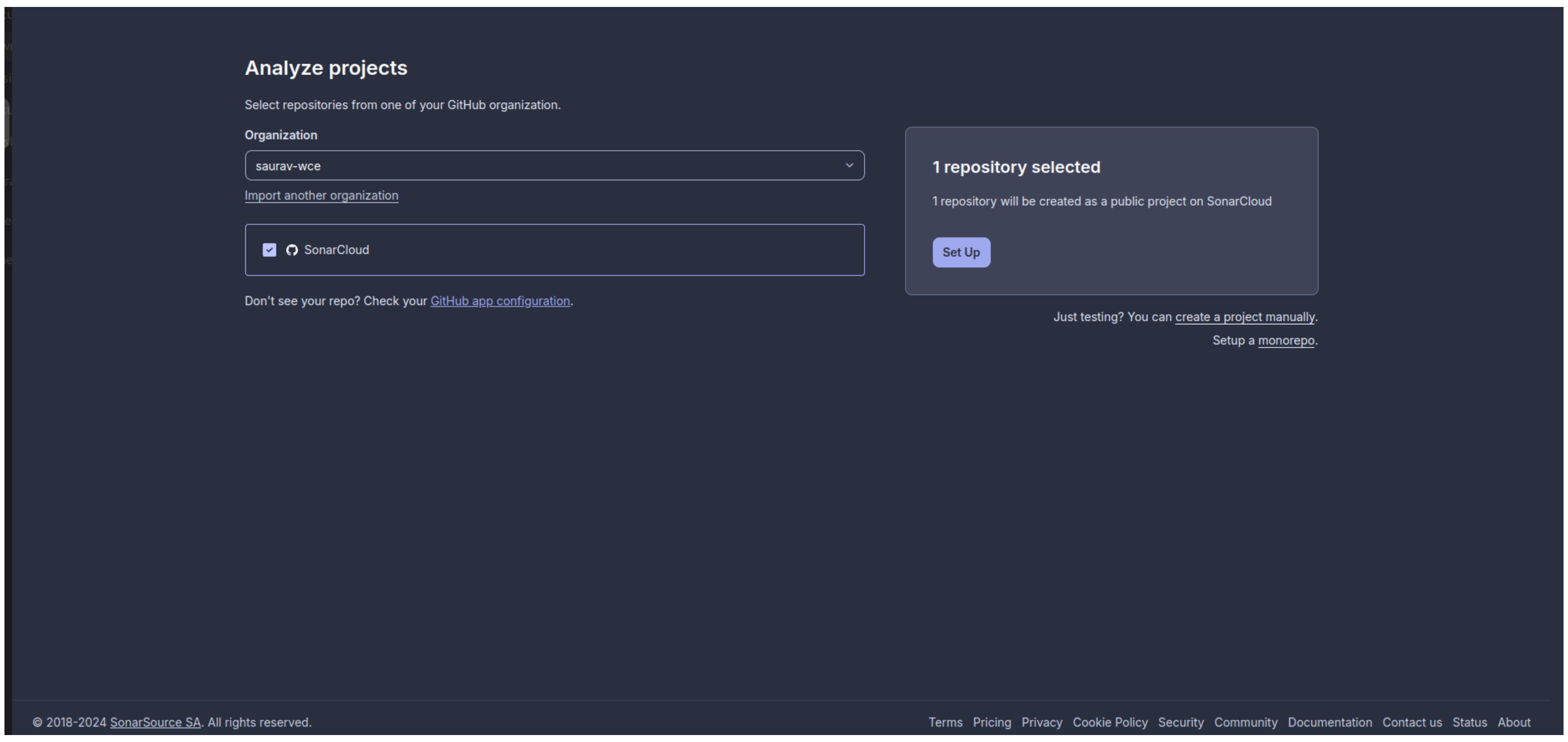


No projects here yet

But your organization is all set, you can start analyzing code here!

© 2018-2024 SonarSource SA. All rights reserved.

15. Click on Analyze a new project
16. Select the repo and click on **setup**


Set Up
Terms Pricing Privacy Cookie Policy Security Community Documentation Contact us Status About

Analyze projects

Select repositories from one of your GitHub organization.

Organization: saurav-wce

Import another organization

☑ SonarCloud

Don't see your repo? Check your GitHub app configuration.

1 repository selected

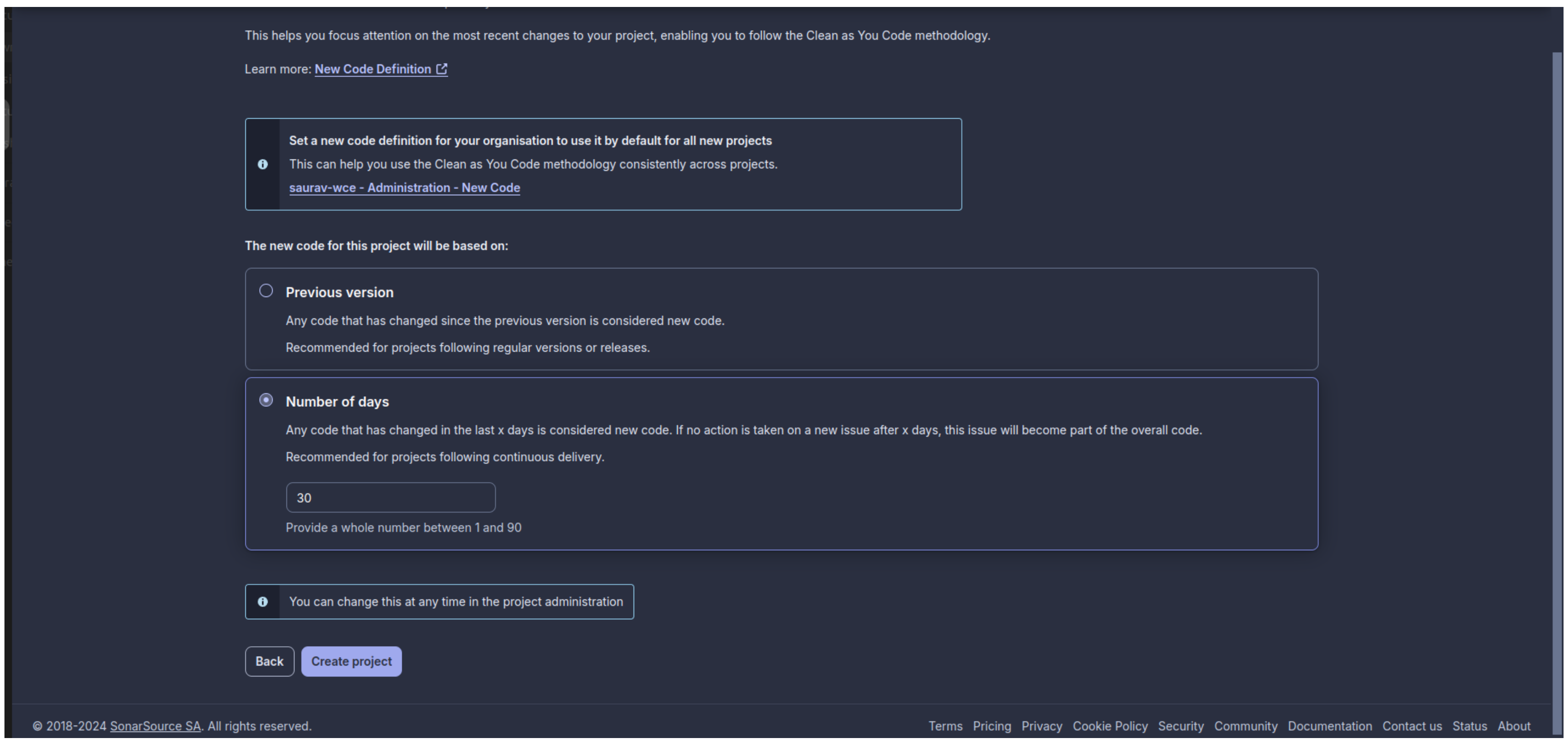
1 repository will be created as a public project on SonarCloud

Just testing? You can create a project manually.

Setup a monorepo.

© 2018-2024 SonarSource SA. All rights reserved.

17. click on 2nd option : **Number of days**


Back
Create project
Terms Pricing Privacy Cookie Policy Security Community Documentation Contact us Status About

This helps you focus attention on the most recent changes to your project, enabling you to follow the Clean as You Code methodology.

Learn more: New Code Definition

Set a new code definition for your organisation to use it by default for all new projects

This can help you use the Clean as You Code methodology consistently across projects.

saurav-wce - Administration - New Code

The new code for this project will be based on:

○ Previous version

Any code that has changed since the previous version is considered new code.

Recommended for projects following regular versions or releases.

● Number of days

Any code that has changed in the last x days is considered new code. If no action is taken on a new issue after x days, this issue will become part of the overall code.

Recommended for projects following continuous delivery.

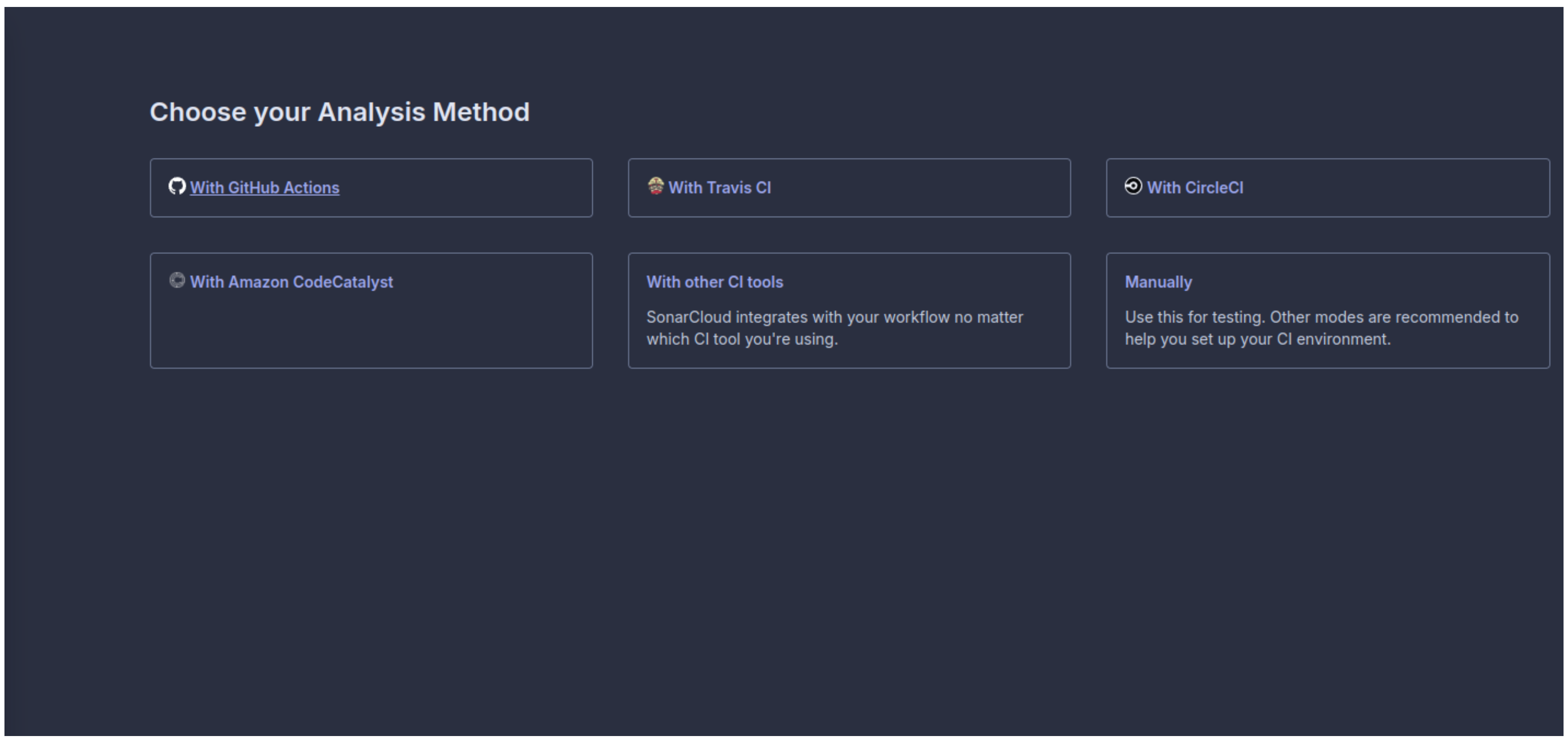
30

Provide a whole number between 1 and 90

You can change this at any time in the project administration

© 2018-2024 SonarSource SA. All rights reserved.

18. Create Project


Choose your Analysis Method

With GitHub Actions

With Travis CI

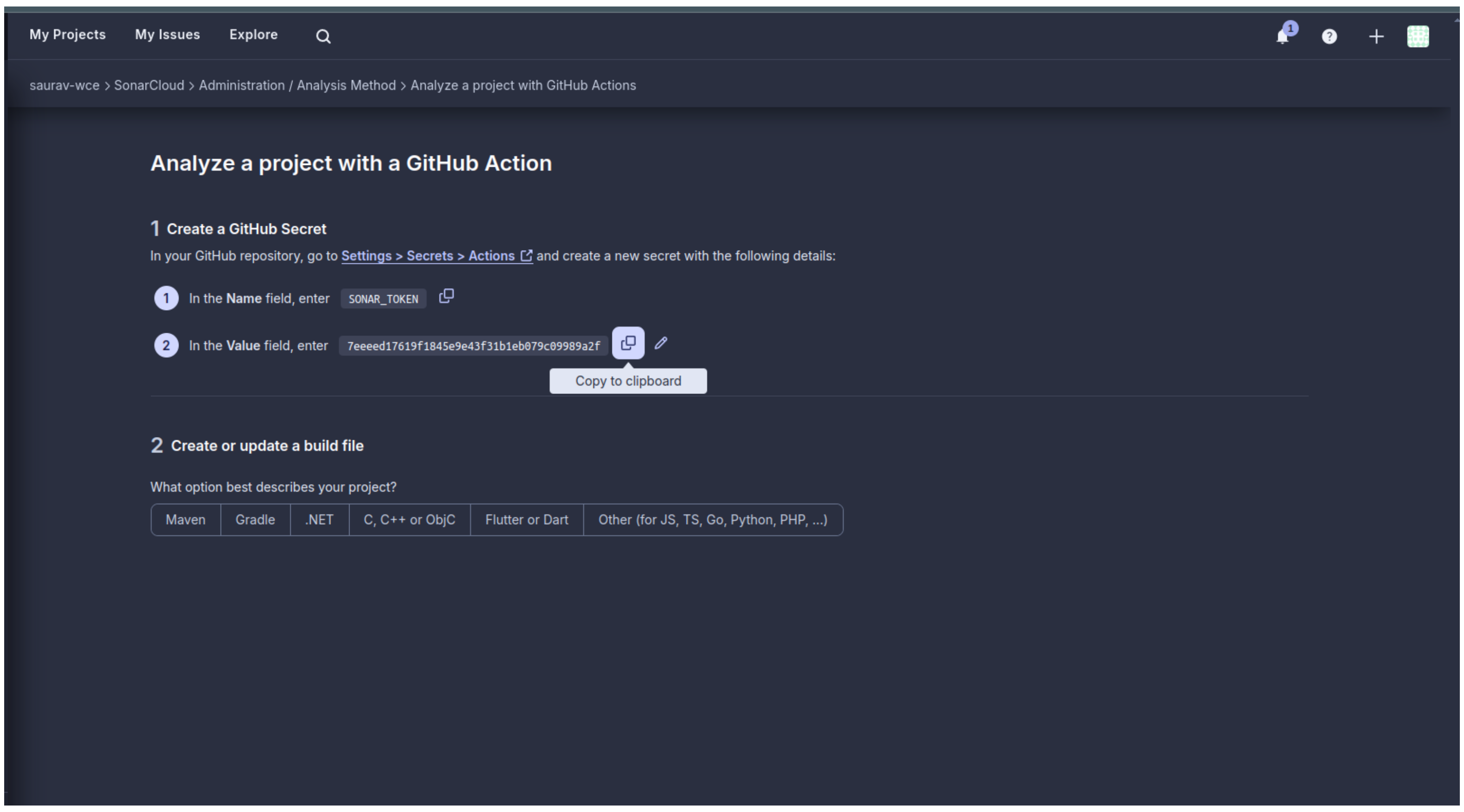
With CircleCI

With Amazon CodeCatalyst

With other CI tools
SonarCloud integrates with your workflow no matter which CI tool you're using.

Manually
Use this for testing. Other modes are recommended to help you set up your CI environment.

19. click on **with github actions**
20. if you see this page then good to go, just copy the token value



saurav-wce > SonarCloud > Administration / Analysis Method > Analyze a project with GitHub Actions

Analyze a project with a GitHub Action

1 Create a GitHub Secret

In your GitHub repository, go to Settings > Secrets > Actions and create a new secret with the following details:

1 In the **Name** field, enter `SONAR_TOKEN`

2 In the **Value** field, enter `7eeeed17619f1845e9e43f31b1eb079c09989a2f`

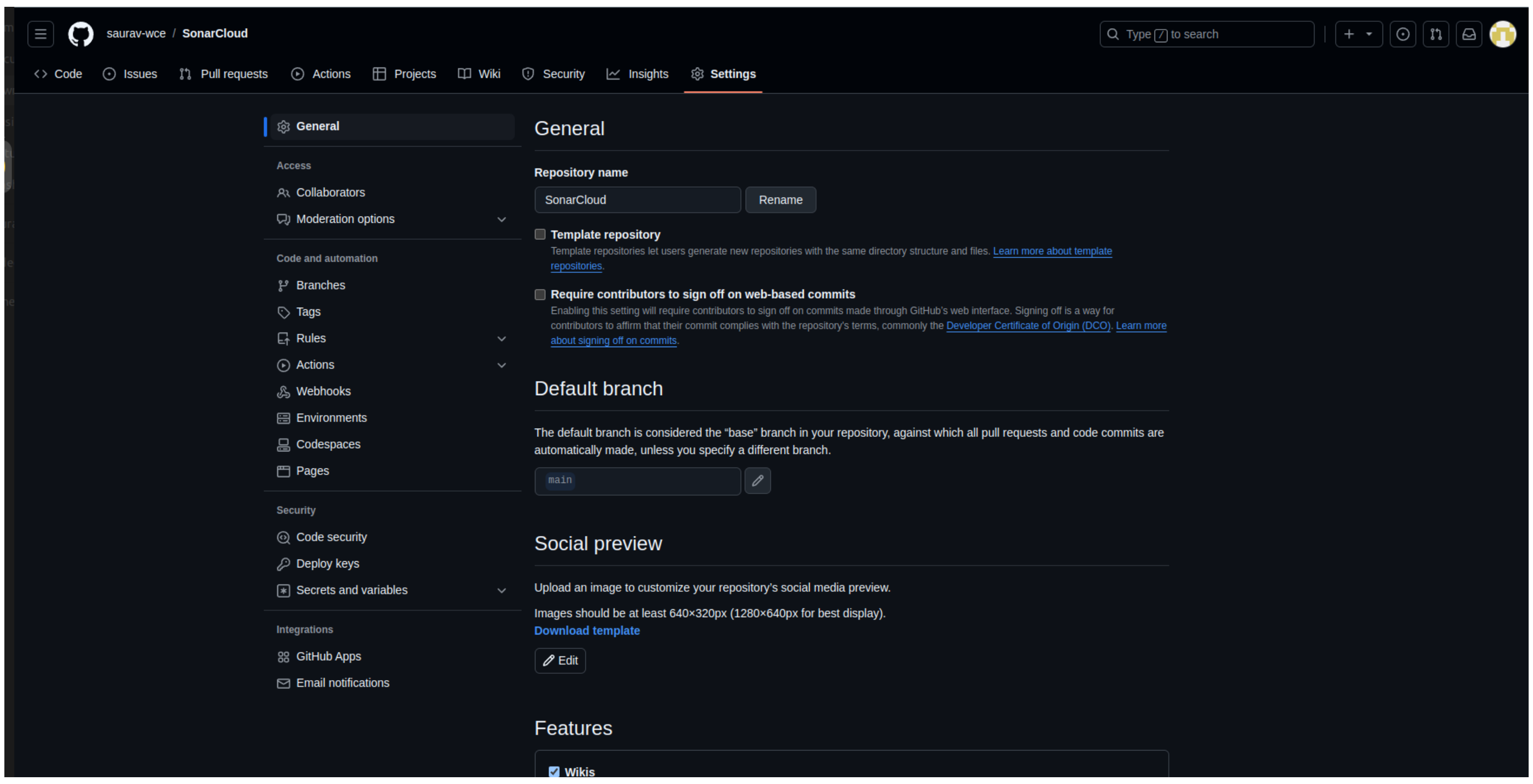
Copy to clipboard

2 Create or update a build file

What option best describes your project?

Maven | Gradle | .NET | C, C++ or ObjC | Flutter or Dart | Other (for JS, TS, Go, Python, PHP, ...)

21. Open a github repo we created in another tab,
22. Open the **settings of Repo as shown in image**



General

Repository name

SonarCloud

Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more about template repositories.

Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

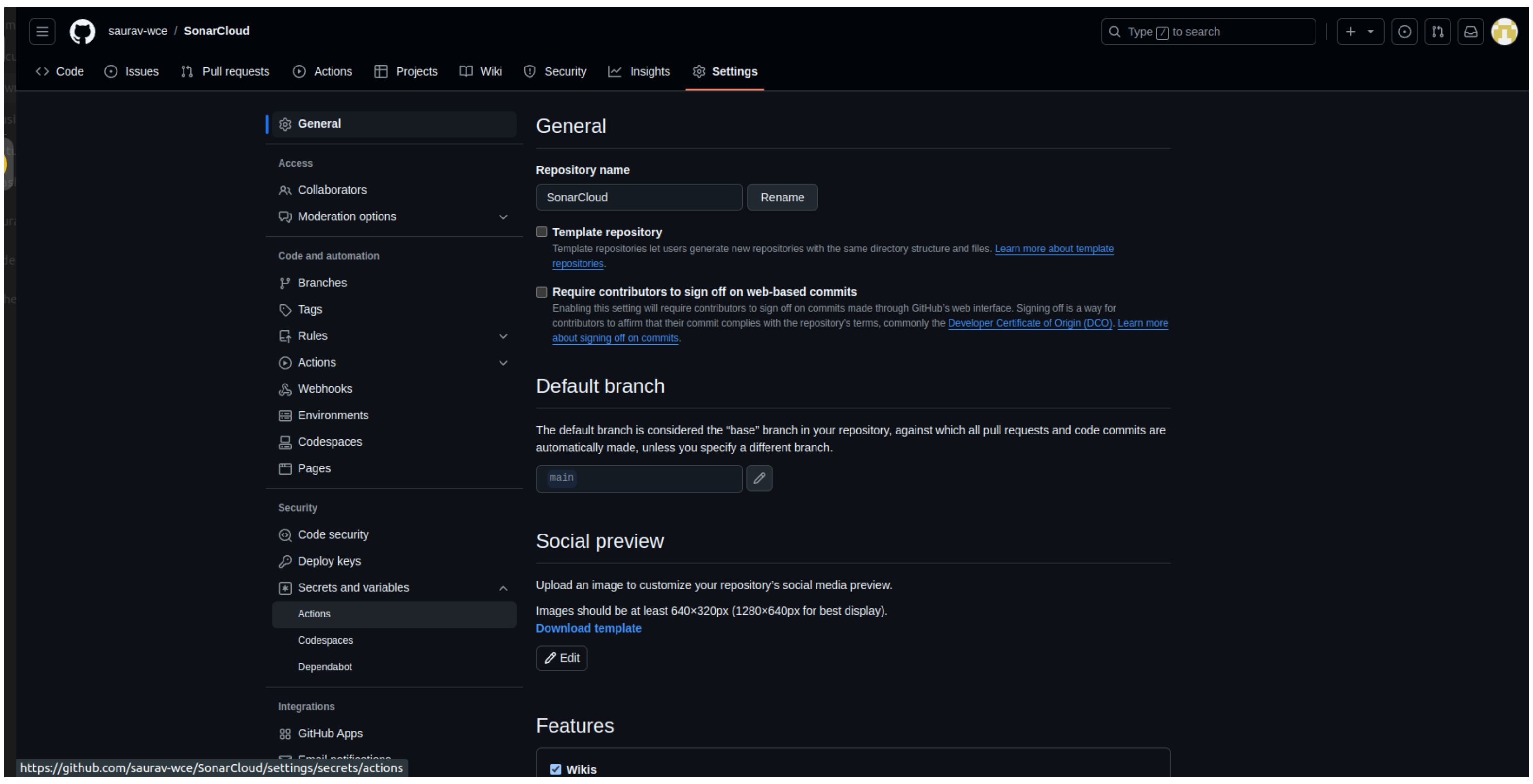
Social preview

Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Features

23. open the **(secret and variables as shown in image) (or just click on <u>this</u>)**



General

Repository name

SonarCloud

Template repository

Template repositories let users generate new repositories with the same directory structure and files. Learn more about template repositories.

Require contributors to sign off on web-based commits

Enabling this setting will require contributors to sign off on commits made through GitHub's web interface. Signing off is a way for contributors to affirm that their commit complies with the repository's terms, commonly the Developer Certificate of Origin (DCO). Learn more about signing off on commits.

Default branch

The default branch is considered the "base" branch in your repository, against which all pull requests and code commits are automatically made, unless you specify a different branch.

main

Social preview

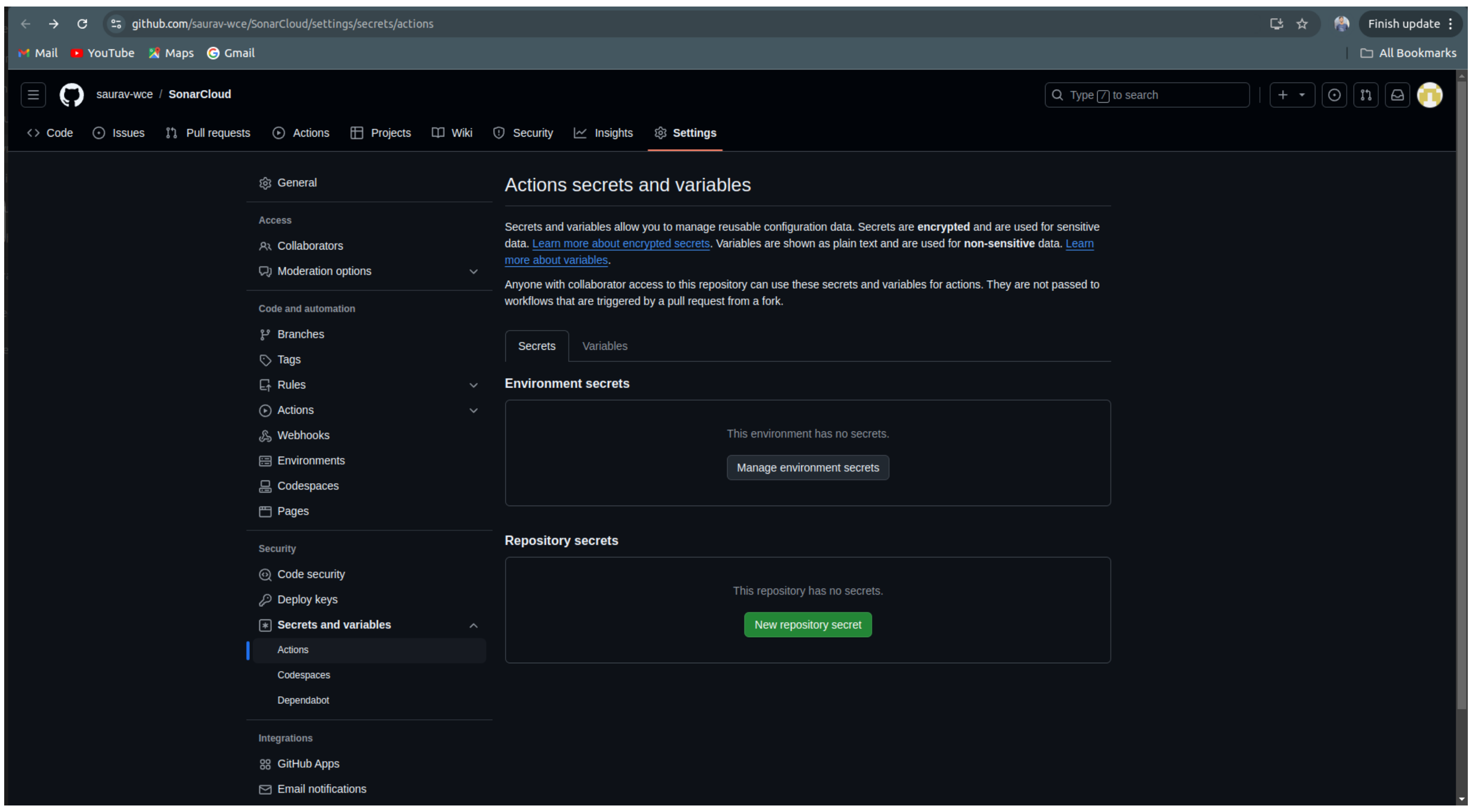
Upload an image to customize your repository's social media preview.

Images should be at least 640×320px (1280×640px for best display).

Features

https://github.com/saurav-wce/SonarCloud/settings/secrets/actions

24. click on new repo secret



Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

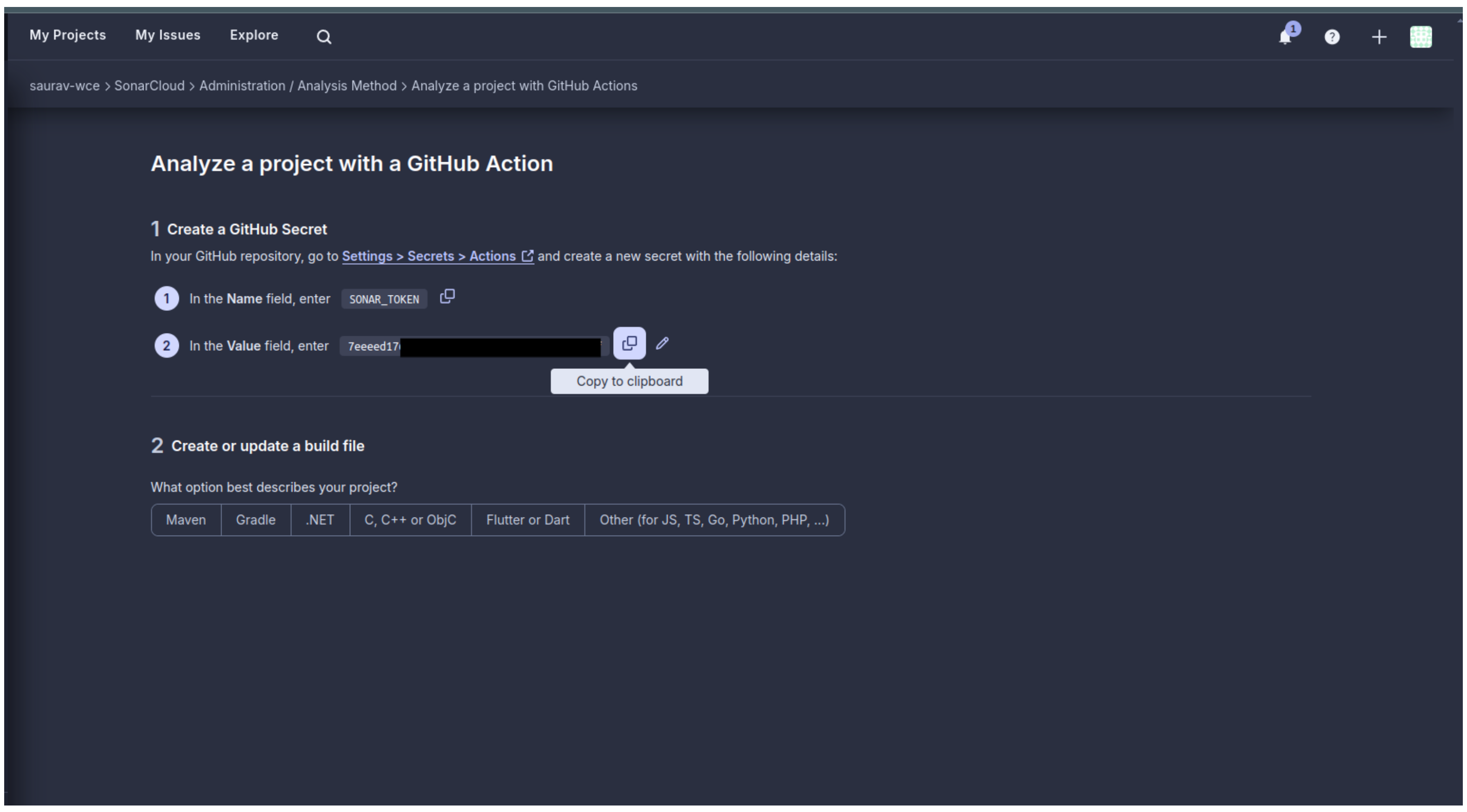
Environment secrets

This environment has no secrets.

Repository secrets

This repository has no secrets.

25. use Name as **SONAR_TOKEN**
26. paste the token value we copied there in step 20


My Projects
My Issues
Explore
saurav-wce > SonarCloud > Administration / Analysis Method > Analyze a project with GitHub Actions

Analyze a project with a GitHub Action

1 Create a GitHub Secret

In your GitHub repository, go to Settings > Secrets > Actions and create a new secret with the following details:

1 In the **Name** field, enter SONAR_TOKEN

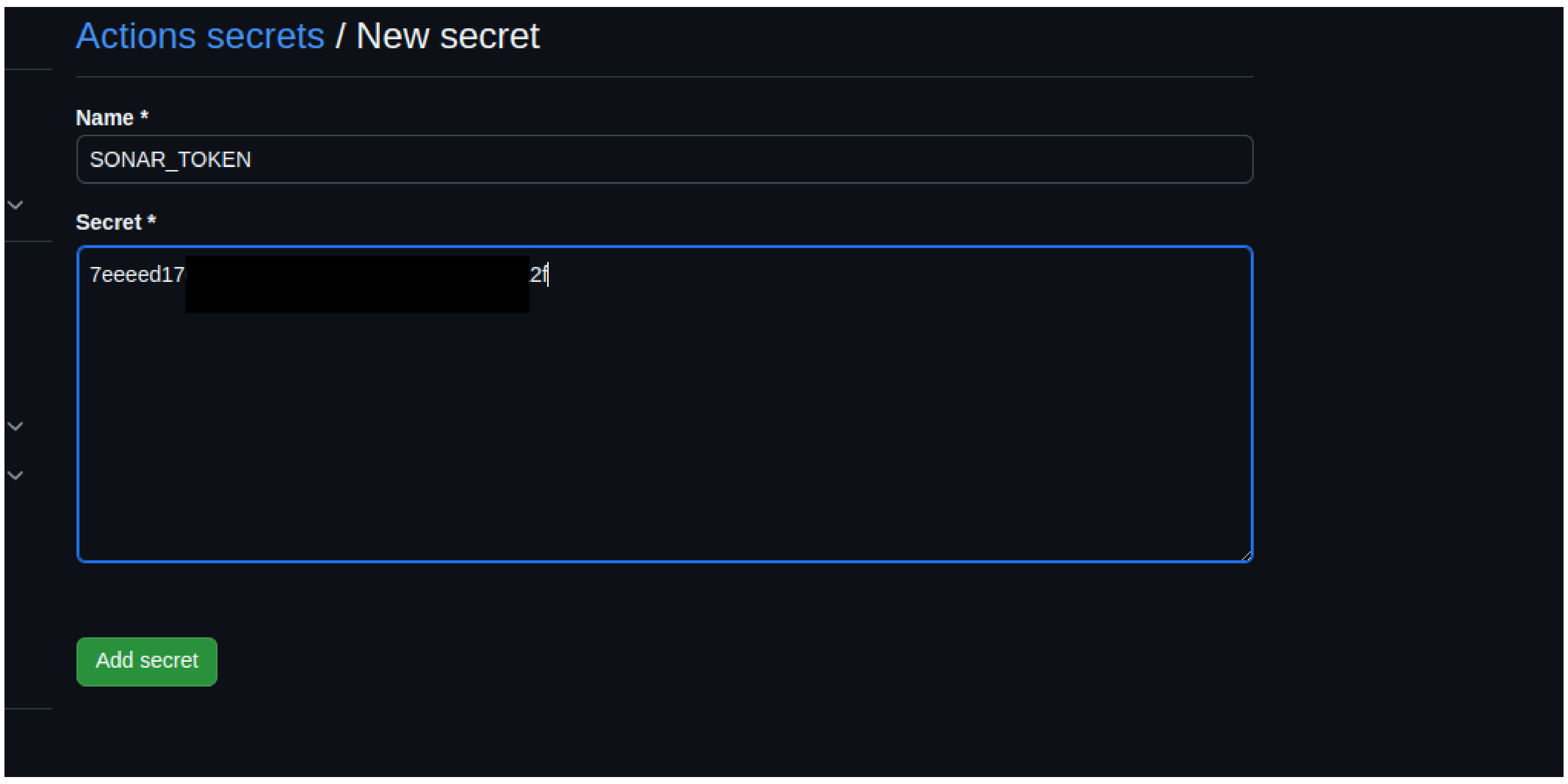
2 In the **Value** field, enter 7eeeed17[redacted]

2 Create or update a build file

What option best describes your project?

Maven
Gradle
.NET
C, C++ or ObjC
Flutter or Dart
Other (for JS, TS, Go, Python, PHP, ...)

27. click on add secret



Actions secrets / New secret

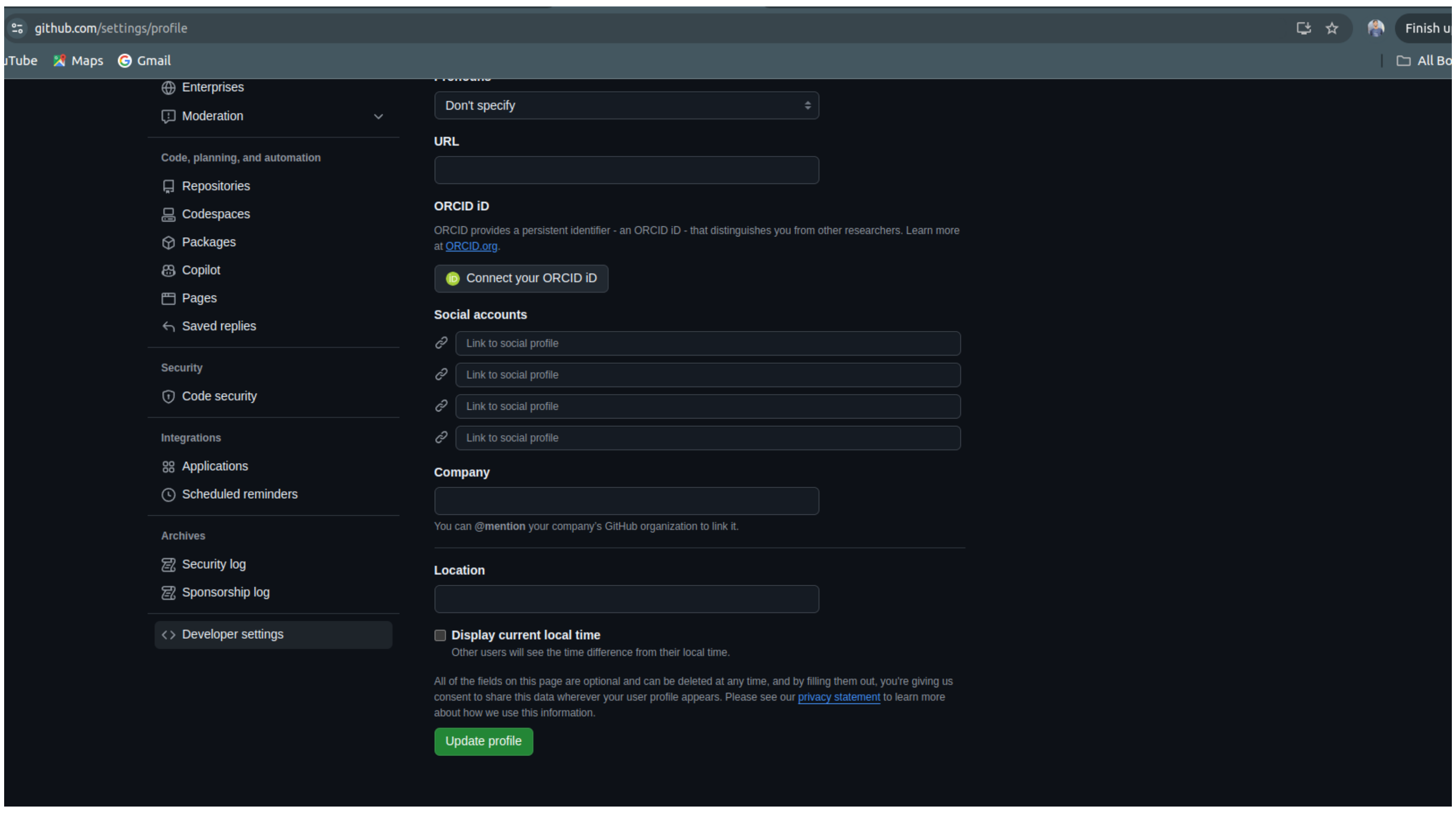
Name *

SONAR_TOKEN

Secret *

7eeeed17[redacted]2f

28. We need to add one more token for that
29. open the **Github Profile settings** by clicking : https://github.com/settings/profile


github.com/settings/profile

Enterprises
Moderation

Code, planning, and automation
Repositories
Codespaces
Packages
Copilot
Pages
Saved replies

Security
Code security

Integrations
Applications
Scheduled reminders

Archives
Security log
Sponsorship log

Developer settings

Don't specify

URL

ORCID iD

ORCID provides a persistent identifier - an ORCID iD - that distinguishes you from other researchers. Learn more at ORCID.org.

Connect your ORCID iD

Social accounts

Link to social profile
Link to social profile
Link to social profile
Link to social profile

Company

You can @mention your company's GitHub organization to link it.

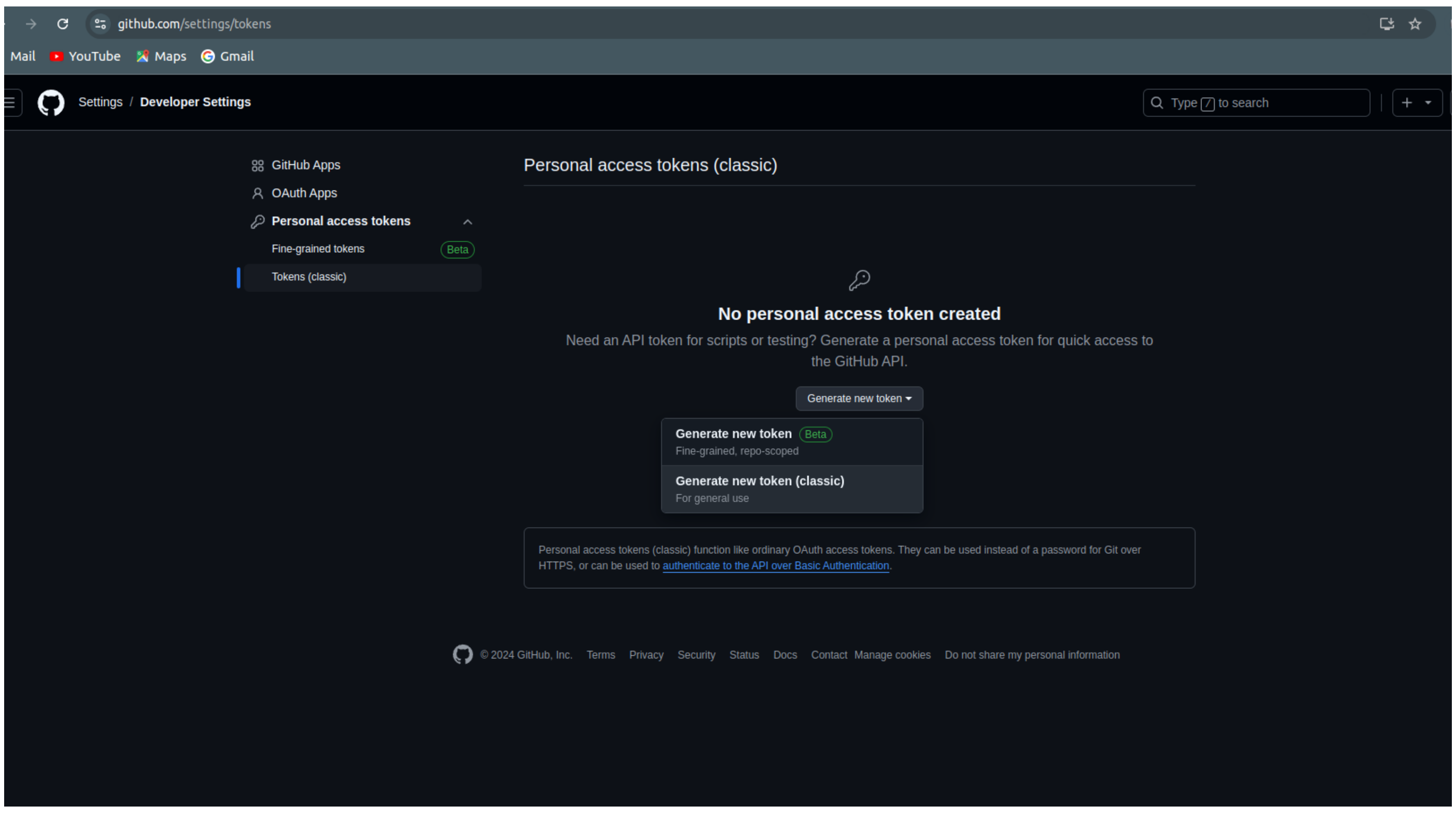
Location

Display current local time
Other users will see the time difference from their local time.

All of the fields on this page are optional and can be deleted at any time, and by filling them out, you're giving us consent to share this data wherever your user profile appears. Please see our privacy statement to learn more about how we use this information.

Update profile

30. Click on Developer Settings : or use Link
31. Now click on Personal Access token ⇒ Tokens (classic) ⇒ Generate New Toekn (classic) ⇒


github.com/settings/tokens

Settings / Developer Settings

GitHub Apps
OAuth Apps
Personal access tokens
Fine-grained tokens Beta
Tokens (classic)

Personal access tokens (classic)

No personal access token created

Need an API token for scripts or testing? Generate a personal access token for quick access to the GitHub API.

Generate new token ▾

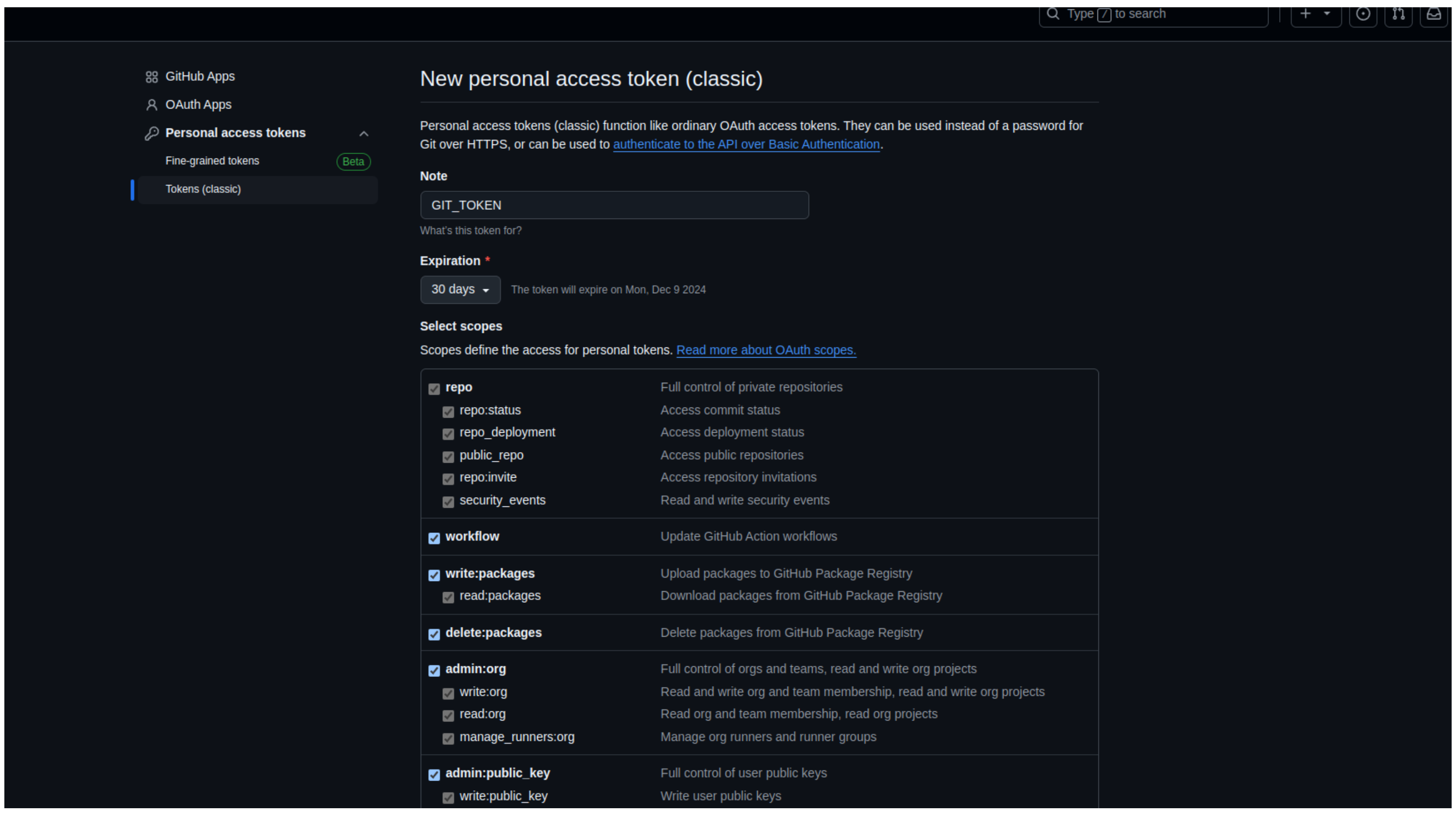
Generate new token Beta
Fine-grained, repo-scoped

Generate new token (classic)
For general use

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

© 2024 GitHub, Inc. Terms Privacy Security Status Docs Contact Manage cookies Do not share my personal information

- add a name as GIT_TOKEN



New personal access token (classic)

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

Note

GIT_TOKEN

What's this token for?

Expiration *

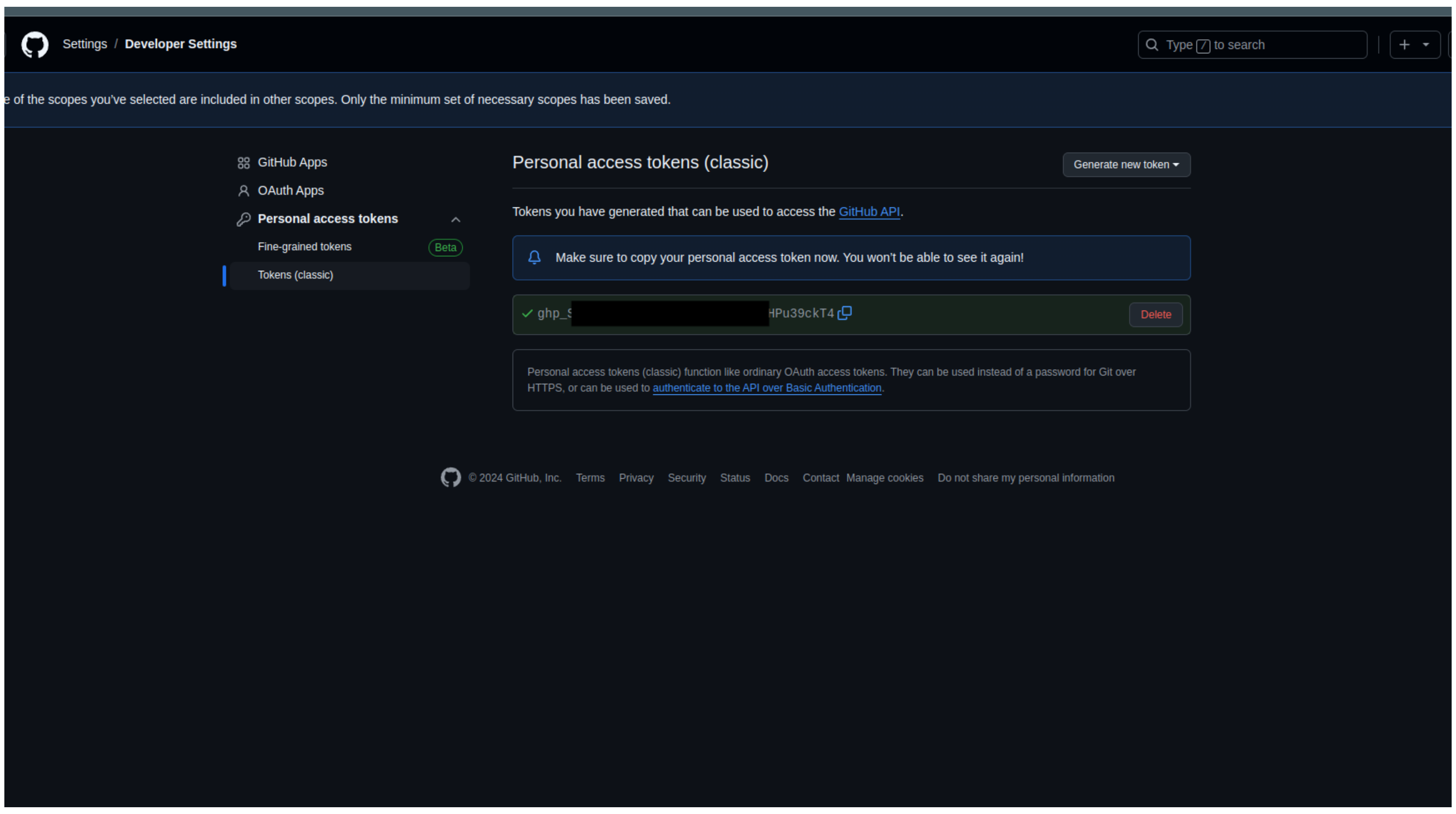
30 days — The token will expire on Mon, Dec 9 2024

Select scopes

Scopes define the access for personal tokens. Read more about OAuth scopes.

✓ **repo**	Full control of private repositories
✓ repo:status	Access commit status
✓ repo_deployment	Access deployment status
✓ public_repo	Access public repositories
✓ repo:invite	Access repository invitations
✓ security_events	Read and write security events
✓ **workflow**	Update GitHub Action workflows
✓ **write:packages**	Upload packages to GitHub Package Registry
✓ read:packages	Download packages from GitHub Package Registry
✓ **delete:packages**	Delete packages from GitHub Package Registry
✓ **admin:org**	Full control of orgs and teams, read and write org projects
✓ write:org	Read and write org and team membership, read and write org projects
✓ read:org	Read org and team membership, read org projects
✓ manage_runners:org	Manage org runners and runner groups
✓ **admin:public_key**	Full control of user public keys
✓ write:public_key	Write user public keys

- tick the all checkboxes
- click on Genereate Token



e of the scopes you've selected are included in other scopes. Only the minimum set of necessary scopes has been saved.

Personal access tokens (classic)

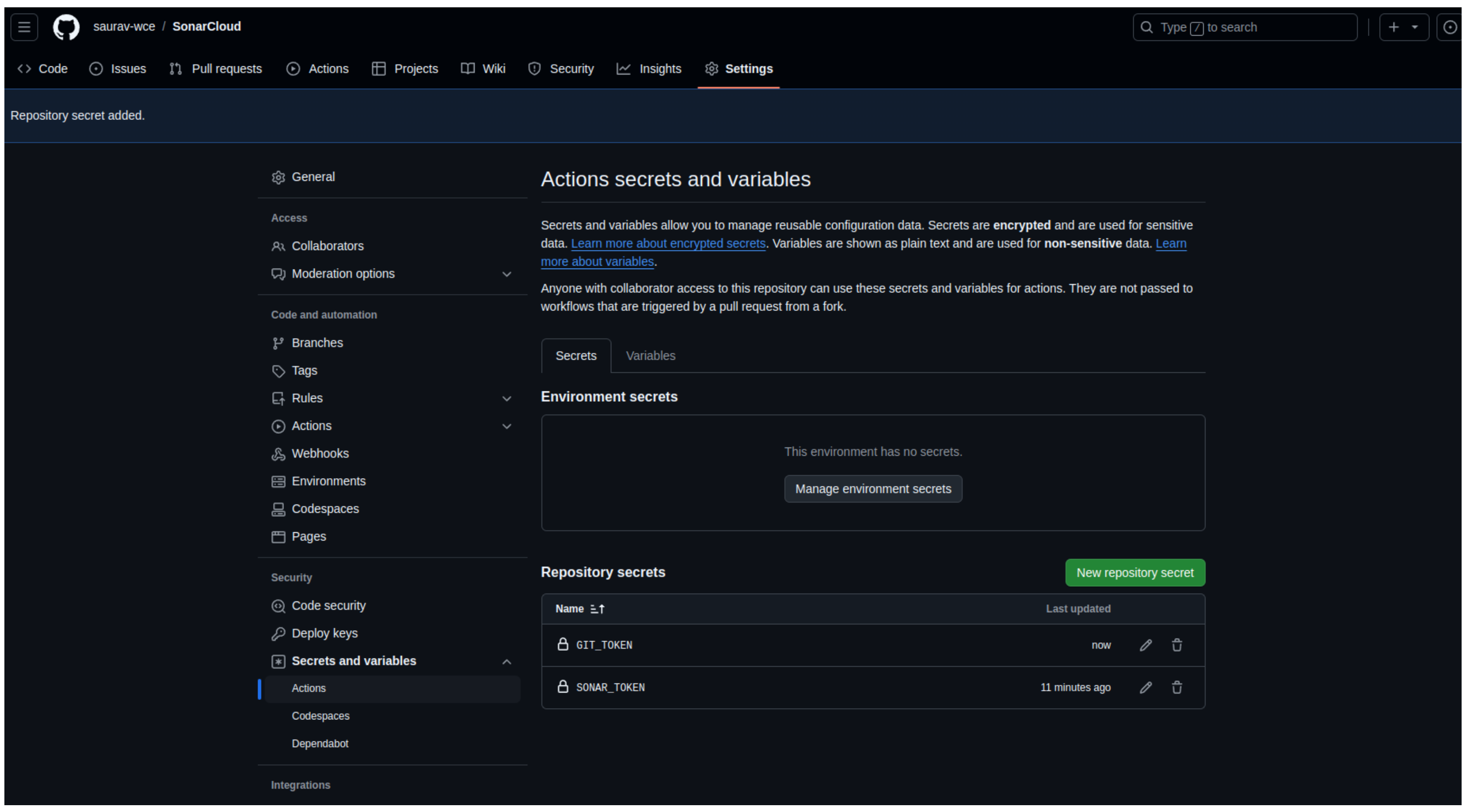
Tokens you have generated that can be used to access the GitHub API.

Make sure to copy your personal access token now. You won't be able to see it again!

✓ ghp_[redacted]HPu39ckT4

Personal access tokens (classic) function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to authenticate to the API over Basic Authentication.

- Copy the token and go to Link add the secret.



Repository secret added.

Actions secrets and variables

Secrets and variables allow you to manage reusable configuration data. Secrets are **encrypted** and are used for sensitive data. Learn more about encrypted secrets. Variables are shown as plain text and are used for **non-sensitive** data. Learn more about variables.

Anyone with collaborator access to this repository can use these secrets and variables for actions. They are not passed to workflows that are triggered by a pull request from a fork.

Environment secrets

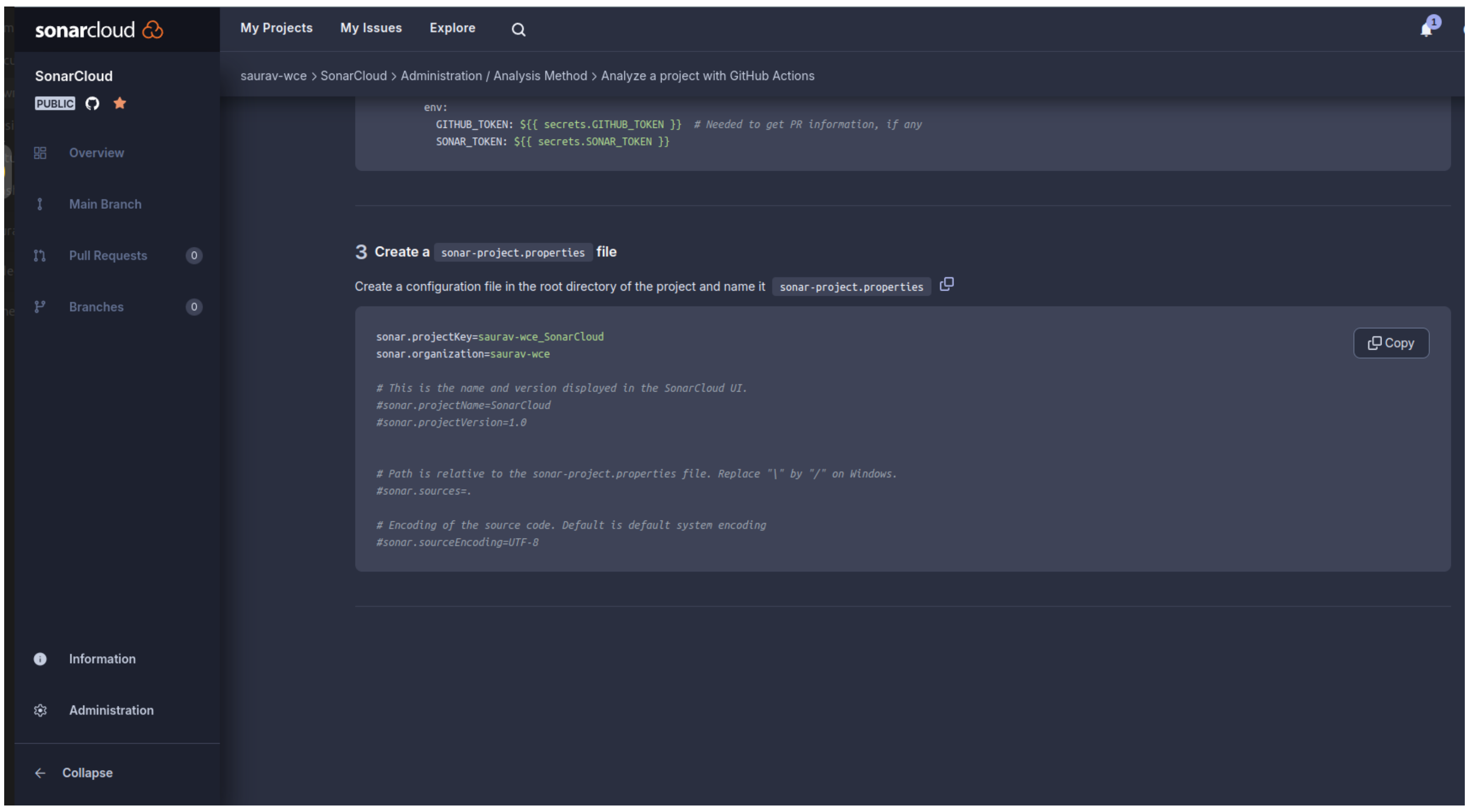
This environment has no secrets.

Repository secrets

Name	Last updated
GIT_TOKEN	now
SONAR_TOKEN	11 minutes ago

- Hope you can see the 2 tokens as above

32. click on **Other (for JS, TS, Go, Python, PHP, ...)**
33. copy the 2 lines from sonar-project.properies



saurav-wce > SonarCloud > Administration / Analysis Method > Analyze a project with GitHub Actions

```
env:
  GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}  # Needed to get PR information, if any
  SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

3 Create a `sonar-project.properties` file

Create a configuration file in the root directory of the project and name it `sonar-project.properties`

```
sonar.projectKey=saurav-wce_SonarCloud
sonar.organization=saurav-wce

# This is the name and version displayed in the SonarCloud UI.
#sonar.projectName=SonarCloud
#sonar.projectVersion=1.0


# Path is relative to the sonar-project.properties file. Replace "\" by "/" on Windows.
#sonar.sources=.

# Encoding of the source code. Default is default system encoding
#sonar.sourceEncoding=UTF-8
```

34. Go to **vs code** and copy paste the codes
    a. File path : `.github/workflows/build.yaml`

```
name: SonarCloud Analysis

on:
  push:
    branches:
      - main
  pull_request:
    types: [opened, synchronize, reopened]
    branches:
      - main

jobs:
  sonarcloud:
    name: SonarCloud
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
        with:
          fetch-depth: 0  # This already gets full hist

      - name: SonarCloud Scan
        uses: SonarSource/sonarcloud-github-action@mast
        env:
          GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
          SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
        with:
          args: >
            -Dsonar.scm.provider=git
            -Dsonar.pullrequest.provider=github
            -Dsonar.pullrequest.github.repository=${{ g
            -Dsonar.pullrequest.key=${{ github.event.pu
            -Dsonar.pullrequest.branch=${{ github.head_
            -Dsonar.pullrequest.base=${{ github.base_re
```

b. File path : `Source Code/script.py`

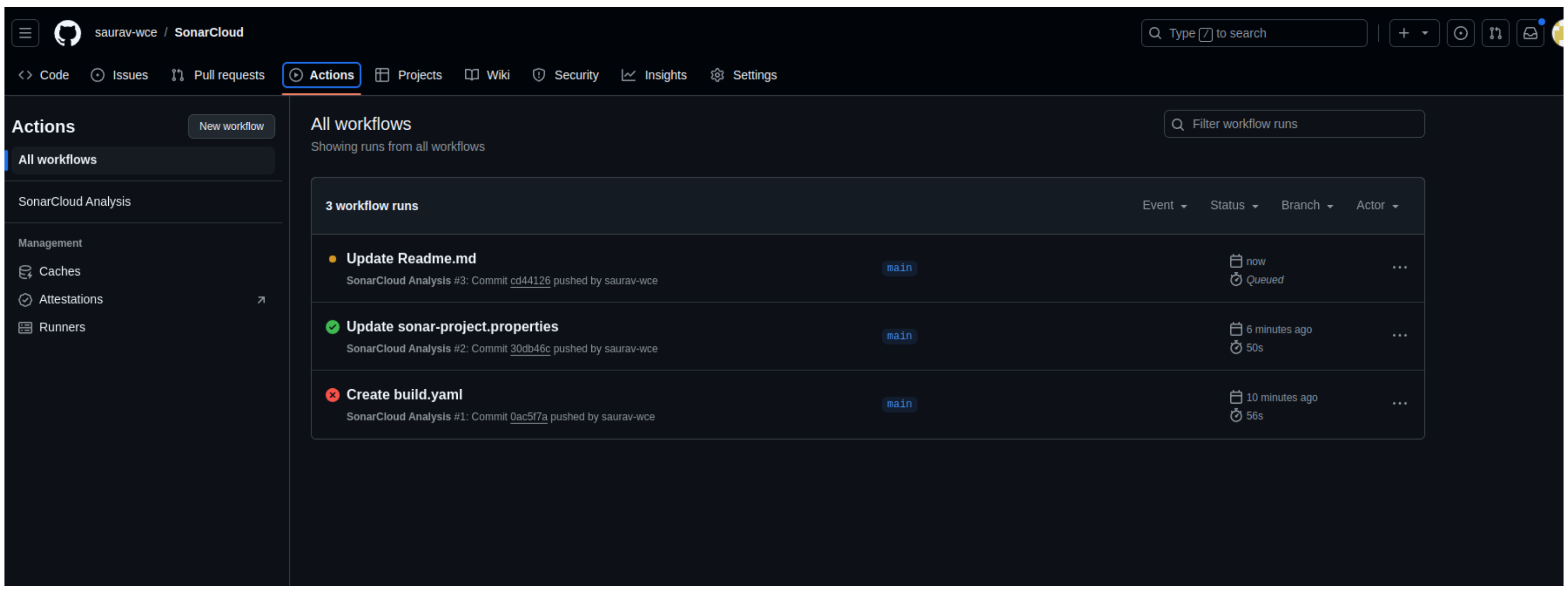
```
import os
import requests
import sys

TOKEN= str(sys.argv[1])
OWNER= str(sys.argv[2])
REPO= str(sys.argv[3])
workflowname= str(sys.argv[4])
parameter1= str(sys.argv[5])
parameter2 = str(sys.argv[6])

print( "the toke value is")
def trigger_workflow(workflowname,parameter1,parameter2

      headers = {
        "Accept": "application/vnd.github.v3+json",
        "Authorization": f"token {TOKEN}",
      }

      data = {
        "event_type": workflowname,
        "client_payload": {
          'parameter1': parameter1,
          'parameter2': parameter2
        }
      }

      responsevalue=requests.post(f"https://api.github.
      print(responsevalue.content)

trigger_workflow(workflowname,parameter1,parameter2)
```

c. File path : `sonar-project.properties` (refer 33 , and change the first 2 lines of following from step 33)

```
# Required metadata change these 2 lines as per your ,
# you an get it there
sonar.projectKey=saurav-wce_SonarCloud
sonar.organization=saurav-wce

# Project information
sonar.projectName=sonar_repo
sonar.projectVersion=1.0

# Source settings
sonar.sources=.
sonar.sourceEncoding=UTF-8

# SCM settings
sonar.scm.provider=git
sonar.scm.forceReloadAll=true

# Exclude files that shouldn't be analyzed
sonar.exclusions=**/node_modules/**,**/*.spec.ts,.githu
```

35. Commit the changes.

36. If you can see the yellow dot as

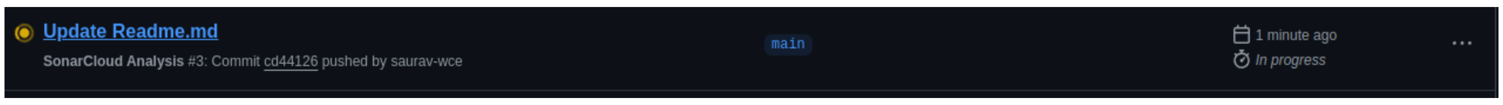


All workflows
Showing runs from all workflows

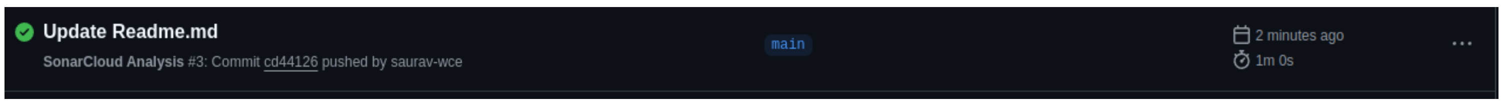
3 workflow runs

Update Readme.md SonarCloud Analysis #3: Commit cd44126 pushed by saurav-wce	main	now Queued
Update sonar-project.properties SonarCloud Analysis #2: Commit 30db46c pushed by saurav-wce	main	6 minutes ago 50s
Create build.yaml SonarCloud Analysis #1: Commit 0ac5f7a pushed by saurav-wce	main	10 minutes ago 56s

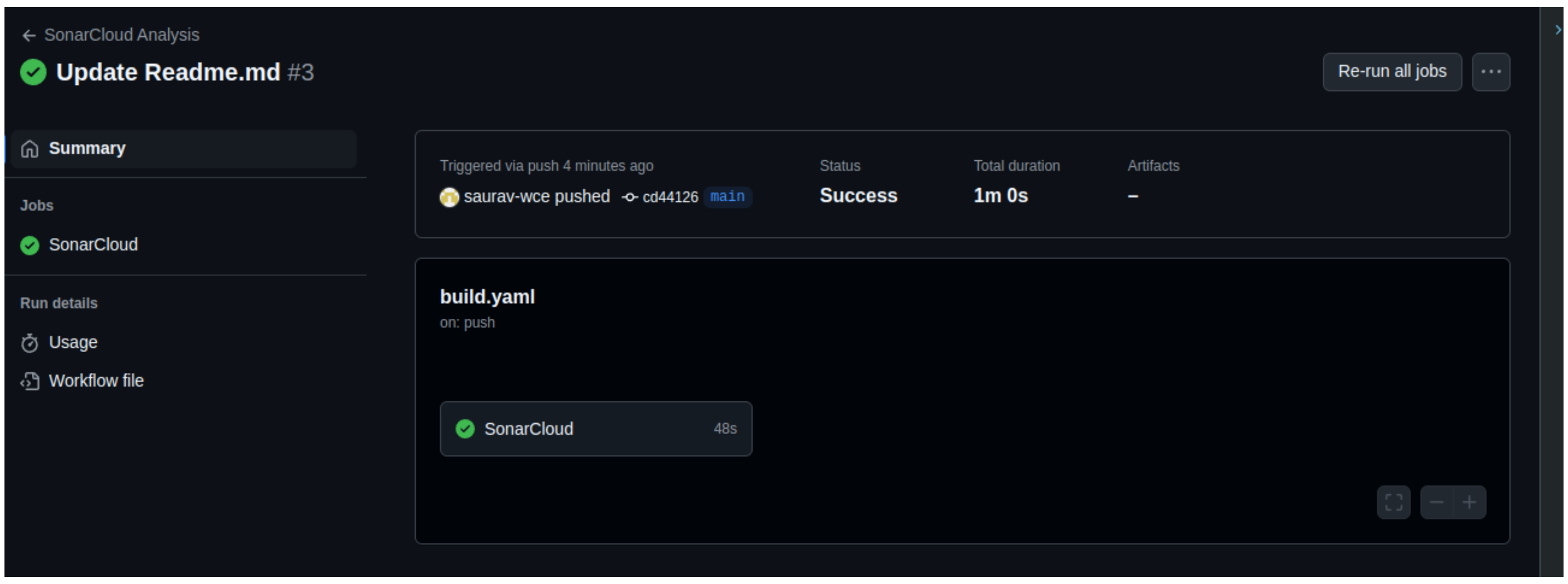
1. If you can see the following yellow dot , then great you are done


Update Readme.md
SonarCloud Analysis #3: Commit cd44126 pushed by saurav-wce
main
1 minute ago
In progress

37. Final step
    a. after a minute you may see ,


Update Readme.md
SonarCloud Analysis #3: Commit cd44126 pushed by saurav-wce
main
2 minutes ago
1m 0s

38. click on update readme.md (in your case this may be different) , commit msg)
39. If you see the following green tick then you are done, click on the **SonarCloud to see the execution**



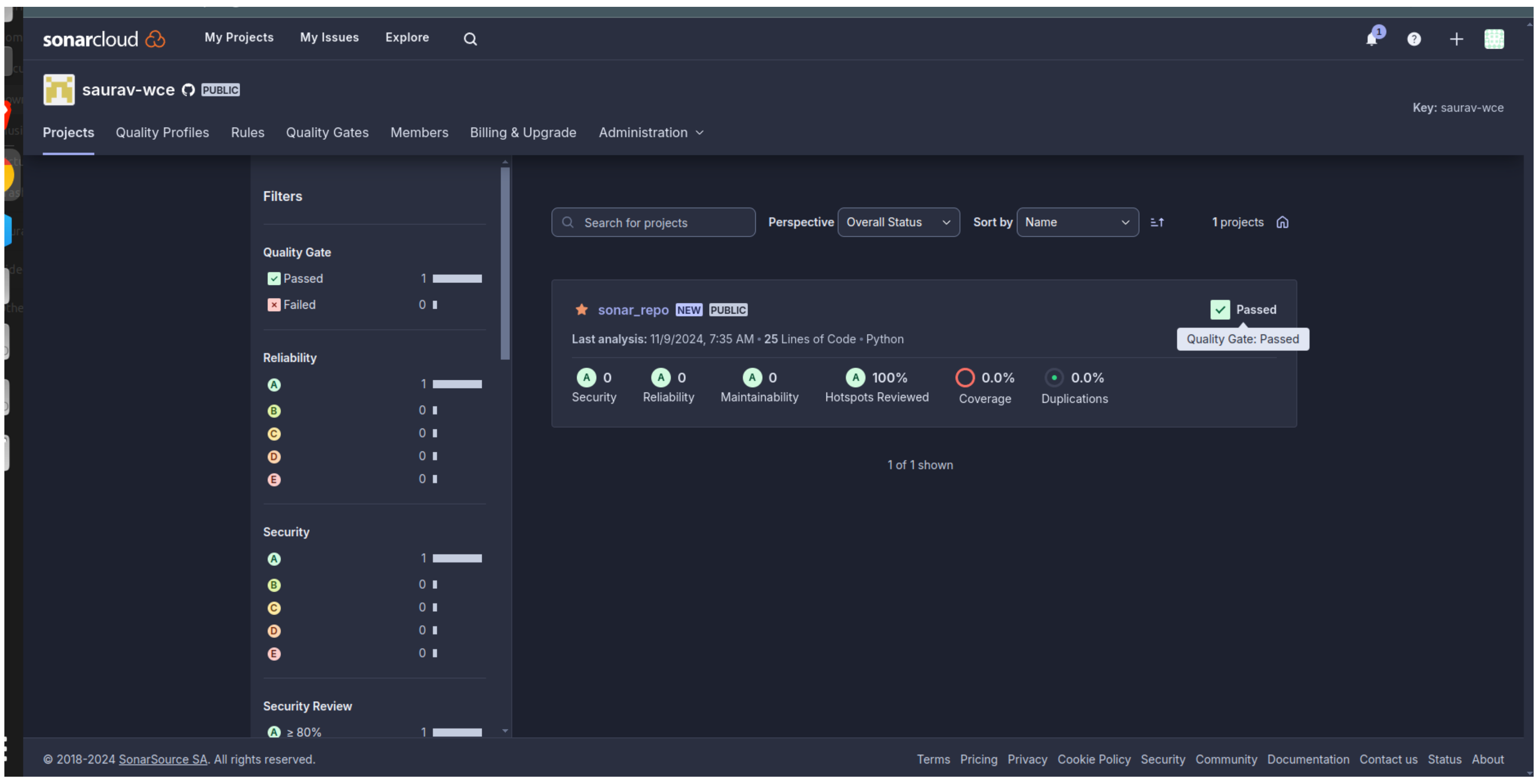
Update Readme.md #3

Triggered via push 4 minutes ago	Status	Total duration	Artifacts
saurav-wce pushed cd44126 main	Success	1m 0s	–

build.yaml
on: push

SonarCloud 48s

40. click on https://sonarcloud.io/projects



Perspective: Overall Status   Sort by: Name

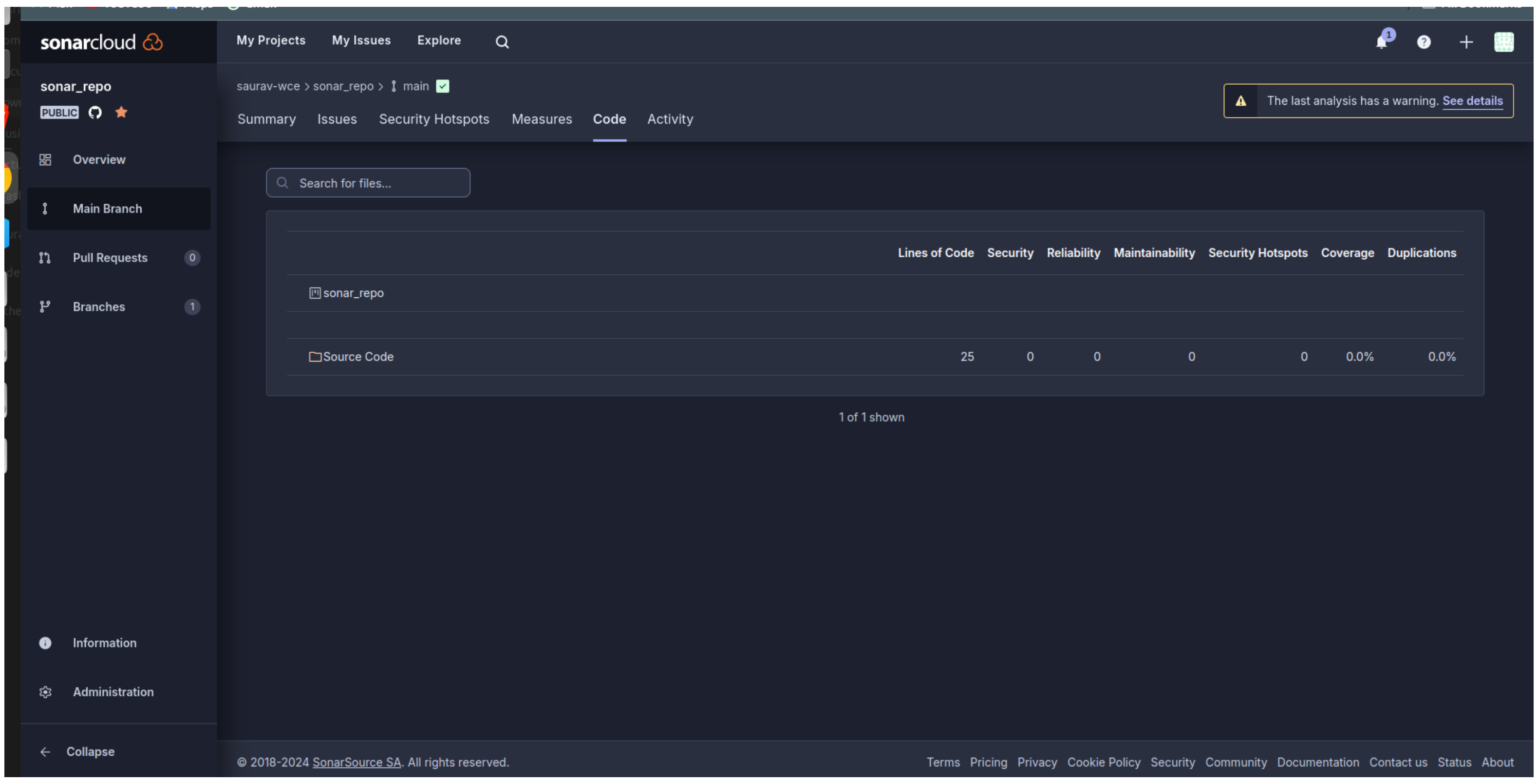
Quality Gate	
Passed	1
Failed	0
Reliability	
A	1
B	0
C	0
D	0
E	0
Security	
A	1
B	0
C	0
D	0
E	0
Security Review	
A ≥ 80%	1

sonar_repo NEW PUBLIC — Passed — Quality Gate: Passed

Last analysis: 11/9/2024, 7:35 AM · 25 Lines of Code · Python

Security	Reliability	Maintainability	Hotspots Reviewed	Coverage	Duplications
A 0	A 0	A 0	A 100%	0.0%	0.0%

41. clcik on the project created (sonar_repo)
42. Hope you can see it



The last analysis has a warning. See details

	Lines of Code	Security	Reliability	Maintainability	Security Hotspots	Coverage	Duplications
sonar_repo							
Source Code	25	0	0	0	0	0.0%	0.0%

Done With setup

▼ Theory to write on paper

Here's a more detailed comparison between SONARcloud and popular project management tools (like **JIRA**, **GitHub**, and **Bitbucket**) to understand how each handles code quality, testing, and project management:

1. Code Coverage

SONARcloud	GitHub	Bitbucket	JIRA
Measures code coverage directly and provides a percentage that's	Can display code coverage through CI/CD integrations (e.g., GitHub	Supports code coverage display when used with	Not built-in for code coverage. Integrates with other tools like

updated with each code analysis.	Actions, third-party apps).	CI/CD services like Bitbucket Pipelines.	SonarQube for reporting.
Provides a detailed report showing which parts of the code are untested.	Coverage reports are basic unless plugins (e.g., Codecov) are used for detailed insights.	Basic coverage reporting; requires external tools for in-depth analysis.	Requires integration for any code coverage insights.

2. Detect Bugs & Vulnerabilities

SONARcloud	GitHub	Bitbucket	JIRA
Automates bug and vulnerability detection with a static code analysis engine; lists critical, major, and minor issues.	Code scanning possible with GitHub Security and third-party apps.	Limited direct support; relies on integrations with security tools (e.g., Snyk) for bug detection.	Primarily a project management tool. Can be combined with SonarQube for insights.
Provides suggestions and recommendations on how to fix detected issues.	Requires setup of security scanning tools or GitHub Actions for similar functionality.	Requires external integrations for automated suggestions or bug fixes.	No direct feature; often linked with other tools for tracking issues.

3. Security Hotspots

SONARcloud	GitHub	Bitbucket	JIRA
Identifies and flags potential security risks (e.g., sensitive data exposure, injections) and categorizes them as “security hotspots.”	Security analysis with GitHub Advanced Security (paid feature) or plugins.	Limited; needs integration with third-party security tools to detect hotspots.	Security tracking usually managed by linking with SonarQube or similar tools.
Provides a review process for developers to verify or remediate flagged hotspots.	Offers minimal review options; requires setup for security alerts.	Limited unless integrated with external security plugins.	Requires dedicated tools for effective security management.

4. Code Smells & Technical Debt

SONARcloud	GitHub	Bitbucket	JIRA
Detects "code smells" and calculates technical debt, helping	Basic code review with PRs, but no built-in code smell detection. Needs	Requires integration with tools like Code Climate for	Tracks tasks related to technical debt when linked

developers maintain cleaner code.	plugins like Code Climate for similar features.	detecting code smells.	with other tools; no direct support.
Technical debt calculation provides a clear estimate of time required to improve code quality.	Technical debt is usually tracked manually or through comments on pull requests.	Technical debt tracking relies on manual input or third-party apps.	Can manage tasks related to tech debt but doesn't provide automated insights.

5. Code Quality Metrics & Historical Tracking

SONARcloud	**GitHub**	**Bitbucket**	**JIRA**
Provides detailed metrics, including maintainability, reliability, security ratings, and duplicated code.	Tracks basic code metrics; can display advanced metrics with integrations.	Basic tracking of PR status; uses plugins for detailed code metrics.	Tracks project status and tasks rather than code quality directly.
Tracks historical changes in code quality, allowing teams to monitor improvements or regressions.	Limited built-in tracking of historical code quality; relies on version history and CI setup.	Limited historical tracking; requires setup of additional tools.	Primarily tracks project timeline, backlog, and task progress, not code quality.

6. Summary of Differences

Feature	**SONARcloud**	**GitHub**	**Bitbucket**	**JIRA**
Code Quality Focus	Specialized in code quality and security	General; primarily version control and CI integration	General; also primarily version control and CI	Project/task management only
Integration Needs	Minimal for code quality; does not need extra plugins	Requires plugins or GitHub Actions for quality metrics	Requires plugins or external tools for quality metrics	Needs plugins or integrations for code quality insights
Code Quality Automation	Automated with direct focus on bugs, vulnerabilities, and tech debt	Basic PR and security scanning with CI support	Basic; relies on integrations with Code Climate or similar tools	No direct code quality; tracks project management tasks

Conclusion

SONARcloud is specialized for code quality management, focusing heavily on detecting and preventing issues in code, with built-in analysis and reporting on code coverage, bugs, vulnerabilities, and technical debt. GitHub and Bitbucket provide broader developer collaboration features but need plugins for advanced code quality metrics. JIRA, as a project management tool, doesn’t focus on code analysis directly but works well for task and progress tracking when integrated with code quality tools like SONARcloud or SonarQube.