SonarQube Setup Documentation (Food-Heaven Project)

Environment: Production

Application: Food-Heaven (Flutter app)

SonarQube URL: https://sonarqube.foodhvn.xyz

Setup Date: July 13, 2025

Maintainer: MD Jahidul Islam

# Overview

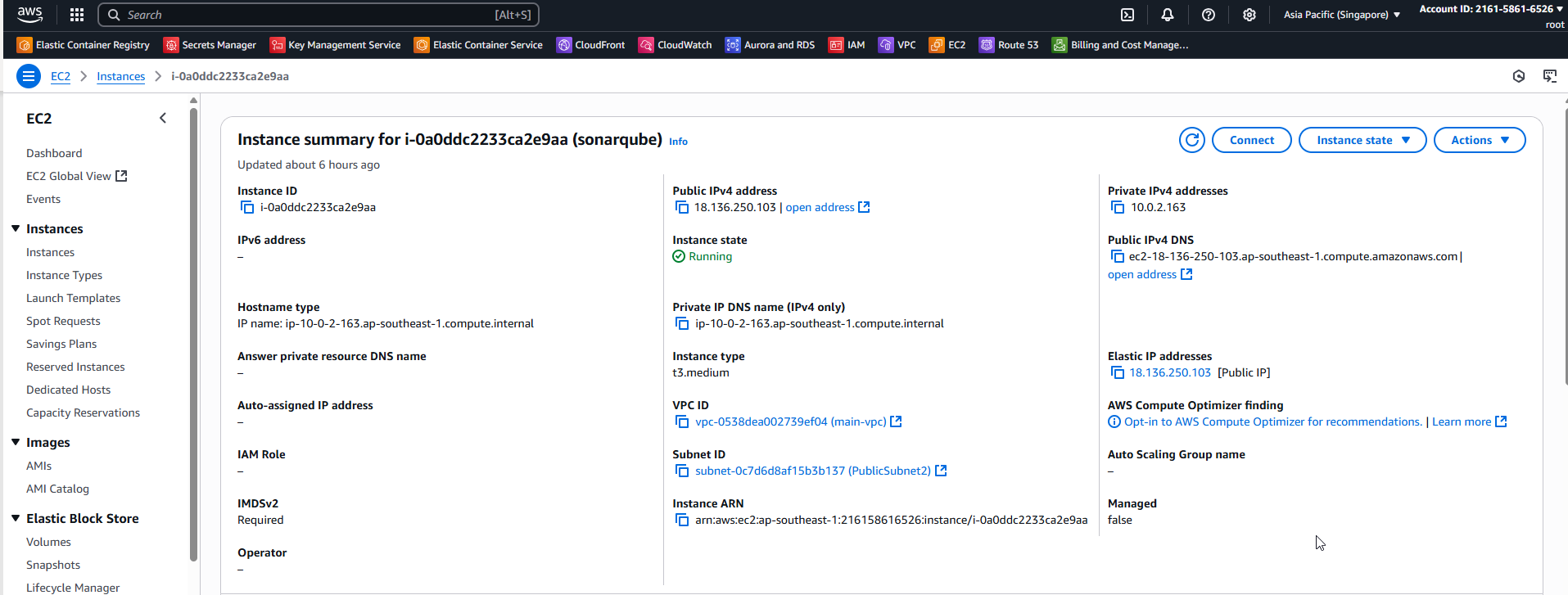
This document details the setup of a self-hosted SonarQube (Community Edition v25.7) instance on an Ubuntu 24.04 EC2 machine, used to continuously analyze and review code quality for the food-heaven application.

# Infrastructure Details

|  |  |
| --- | --- |
| Component | Value |
| Cloud Provider | AWS EC2 |
| Instance Type | t2.medium (2vCPU, 4GB+ RAM) |
| OS Version | Ubuntu 24.04 LTS |
| Hostname | sonarqube.foodhvn.xyz |
| Public IP | Elastic ip |
| Ports Allowed | 22, 80, 443, 9000 |
| IAM Role | None attached |
| Security Group | Open to HTTP/HTTPS (restricted preferred) |
| Access Control | Admin login manually created |
| SSL certification | Safe browsing |
| Refdirect http to https | Enabled |

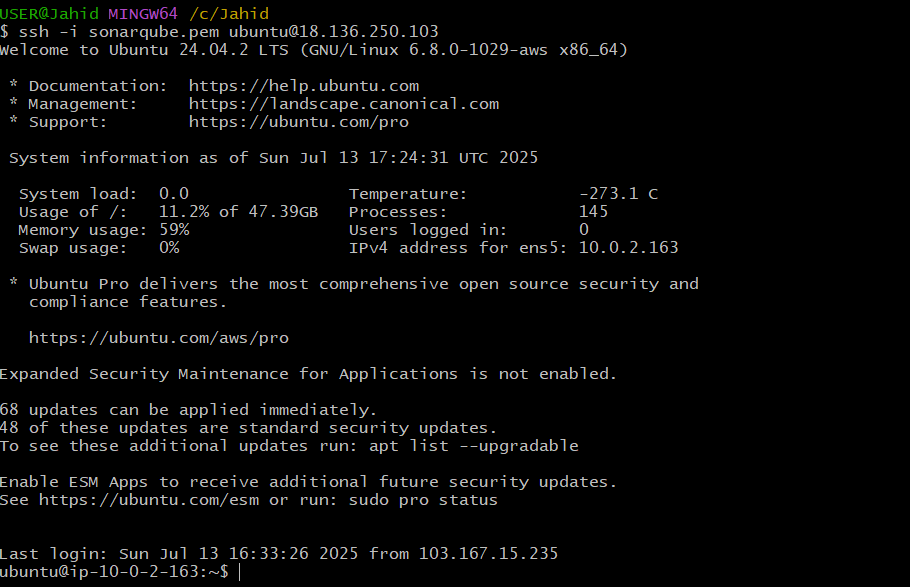
# SonarQube Setup Process

## 1. EC2 Initialization

- Launched EC2 using Ubuntu 24.04 AMI  
- Attached elastic IP and updated Route 53 DNS  
- Installed required packages:  


Connect it by git cli:

## 



**Set Up a PostgreSQL Database for SonarQube**

1. Install the PostgreSQL if it's not installed on your Ubuntu 24.04 workstation.

**sudo apt install -y postgresql-common postgresql -y**

1. Enable the PostgreSQL database server to automatically start at boot.

**sudo systemctl enable postgresql**

Output:

Synchronizing state of postgresql.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.

Executing: /usr/lib/systemd/systemd-sysv-install enable postgresql

1. Start the PostgreSQL database server.

$ sudo systemctl start postgresql

1. Log in to the PostgreSQL database server as the postgres user.

$ sudo -u postgres psql

1. Create a new sonaruser PostgreSQL role with a strong password to use with SonarQube. Replace your\_password with your desired password.

**CREATE** **ROLE** sonaruser **WITH** LOGIN **ENCRYPTED** **PASSWORD** '$&GHDE&$(9#$';

1. Create a new sonarqube database.

**CREATE** **DATABASE** sonarqube;

1. Grant the sonaruser role full privileges to the sonarqube database.

**GRANT** **ALL** **PRIVILEGES** **ON** **DATABASE** sonarqube **TO** sonaruser;

1. Switch to the sonarqube database

\c sonarqube

Output:

You are now connected to database "sonarqube" as user "postgres".

1. Grant the sonaruser role full privileges to the public schema.

**GRANT** **ALL** **PRIVILEGES** **ON** **SCHEMA** public **TO** sonaruser;

1. Exit the PostgreSQL database console.

\q

**Install Sonarqube:**

Update the server's APT package index.

$ sudo apt update

Install OpenJDK 17.

$ sudo apt install openjdk-17-jdk -y

Install Unzip to extract files from the SonarQube archive.

$ sudo apt install unzip

Verify the installed java version.

$ java -version

Your output should be similar to the one below:

openjdk version "17.0.14" 2025-01-21

OpenJDK Runtime Environment (build 17.0.14+7-Ubuntu-124.04)

OpenJDK 64-Bit Server VM (build 17.0.14+7-Ubuntu-124.04, mixed mode, sharing)

Visit the [SonarQube releases page](https://binaries.sonarsource.com/?prefix=Distribution/sonarqube/) and verify the latest version to download. For example, sonarqube-25.2.0.102705.zip.

Download the latest SonarQube archive.

$ sudo wget https://binaries.sonarsource.com/Distribution/sonarqube/sonarqube-25.2.0.102705.zip

Extract files from the downloaded archive using Unzip.

$ unzip sonarqube-25.2.0.102705.zip

Move the extracted files to a systemwide directory such as /opt.

$ sudo mv sonarqube-25.2.0.102705/ /opt/sonarqube

Create a dedicated sonarqube system user without login privileges and a home directory.

$ sudo adduser --system --no-create-home --group --disabled-login sonarqube

Grant the sonarqube user full privileges to the /opt/sonarqube directory.

$ sudo chown -R sonarqube:sonarqube /opt/sonarqube

**Install SonarScanner CLI**

[Visit the SonarScanner CLI page](https://github.com/SonarSource/sonar-scanner-cli/releases) and verify the latest version to download. For example, run the following command to download the SonarScanner CLI version 7.0.1

$ wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-7.0.1.4817-linux-x64.zip

Extract files from the archive depending on the downloaded version.

$ unzip sonar-scanner-cli-7.0.1.4817-linux-x64.zip

Move the extracted directory to /opt/sonarscanner.

$ sudo mv sonar-scanner-7.0.1.4817-linux-x64/ /opt/sonarscanner

Open the sonar-scanner.properties configuration file.

$ sudo nano /opt/sonarscanner/conf/sonar-scanner.properties

Find the following sonar.host.url directive and change the default https://mycompany.com/sonarqube value to 127.0.0.1.

...

sonar.host.url=127.0.0.1

...

Save and close the file.

The above Sonar Host directive specifies the SonarQube server URL to use while performing code scans.

Enable execute permissions on the sonar-scanner binary.

$ sudo chmod +x /opt/sonarscanner/bin/sonar-scanner

Link the sonar-scanner binary to the /usr/local/bin directory to enable it as a system-wide command.

$ sudo ln -s /opt/sonarscanner/bin/sonar-scanner /usr/local/bin/sonar-scanner

View the installed SonarScanner version.

$ sonar-scanner -v

Your output should be similar to the one below.

........

13:33:31.946 INFO SonarScanner CLI 7.0.1.4817

13:33:31.950 INFO Java 17.0.13 Eclipse Adoptium (64-bit)

13:33:31.951 INFO Linux 6.8.0-51-generic amd64

**Configure SonarQube:**

Open the main sonar.properties Sonarqube configuration file.

$ sudo nano /opt/sonarqube/conf/sonar.properties

Add the following configurations at the end of the file. Replace sonaruser and your\_password with actual PostgreSQL database user details.

sonar.jdbc.username=sonaruser

sonar.jdbc.password=your\_password

sonar.jdbc.url=jdbc:postgresql://localhost:5432/sonarqube

sonar.web.javaAdditionalOpts=-server

sonar.web.host=0.0.0.0

sonar.web.port=9000

Save and close the file.

The above custom configuration directives enable SonarQube to access the PostgreSQL database, and listen for connections on the TCP port 9000 from all network addresses 0.0.0.0.

Open the sysctl.config configuration file to modify the system memory limits.

$ sudo nano /etc/sysctl.conf

Add the following directives at the end of the file.

vm.max\_map\_count=524288

fs.file-max=131072

Save and close the file.

Within the above configuration:

* vm.max\_map\_count=524288: Increases the number of memory maps Elasticsearch can use, allowing it to handle large datasets.
* fs.file-max=131072: Increases the maximum number of files Elasticsearch can open, allowing it to run efficiently.

SonarQube uses Elasticsearch to store indices in a memory-mapped file system. Adjusting the system limits for virtual memory mapping and file handling ensures better stability and performance for SonarQube.

Create a new /etc/security/limits.d/99-sonarqube.conf file to create a resource limits configuration for SonarQube.

$ sudo nano /etc/security/limits.d/99-sonarqube.conf

Add the following directives to increase the file descriptor and process limits for SonarQube.

sonarqube - nofile 131072

sonarqube - nproc 8192

Save and close the file.

Within the configuration:

* nofile=131072: Increases the number of open file descriptors, allowing SonarQube to handle large workloads.
* nproc=8192: Raises the process limit to prevent failures under high concurrency.

Allow network connections to the SonarQube port 9000.

$ sudo ufw allow **9000**/tcp

Run the following command to install UFW and allow SSH connections if it's unavailable.

$ sudo apt install ufw -y && sudo ufw allow **22**/tcp

Reload UFW to apply the firewall configurations.

$ sudo ufw reload

View the UFW status and verify that below are the only active firewall rules.

$ sudo ufw status

Your output should be similar to the one below:

Status: active

To Action From

-- ------ ----

22/tcp ALLOW Anywhere

9000/tcp ALLOW Anywhere

22/tcp (v6) ALLOW Anywhere (v6)

9000/tcp (v6) ALLOW Anywhere (v6)

**Set Up SonarQube as a System Service**

Create a new sonarqube.service file.

$ sudo nano /etc/systemd/system/sonarqube.service

Add the following configurations to the file.

**[Unit]**

Description=SonarQube service

After=syslog.target network.target

**[Service]**

Type=forking

ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start

ExecStop=/opt/sonarqube/bin/linux-x86-64/sonar.sh stop

User=sonarqube

Group=sonarqube

PermissionsStartOnly=true

Restart=always

StandardOutput=syslog

LimitNOFILE=131072

LimitNPROC=8192

TimeoutStartSec=5

SuccessExitStatus=143

**[Install]**

WantedBy=multi-user.target

Save and close the file.

The above configuration creates a new SonarQube system service to monitor and manage the application processes.

Reload systemd to apply the service changes.

$ sudo systemctl daemon-reload

Enable SonarQube to start at boot.

$ sudo systemctl enable sonarqube

Start the SonarQube service.

$ sudo systemctl start sonarqube

View the SonarQube service status and verify that it's running.

$ sudo systemctl status sonarqube

Your output should be similar to the one below:

● sonarqube.service - SonarQube service

Loaded: loaded (/etc/systemd/system/sonarqube.service; enabled; preset: enabled)

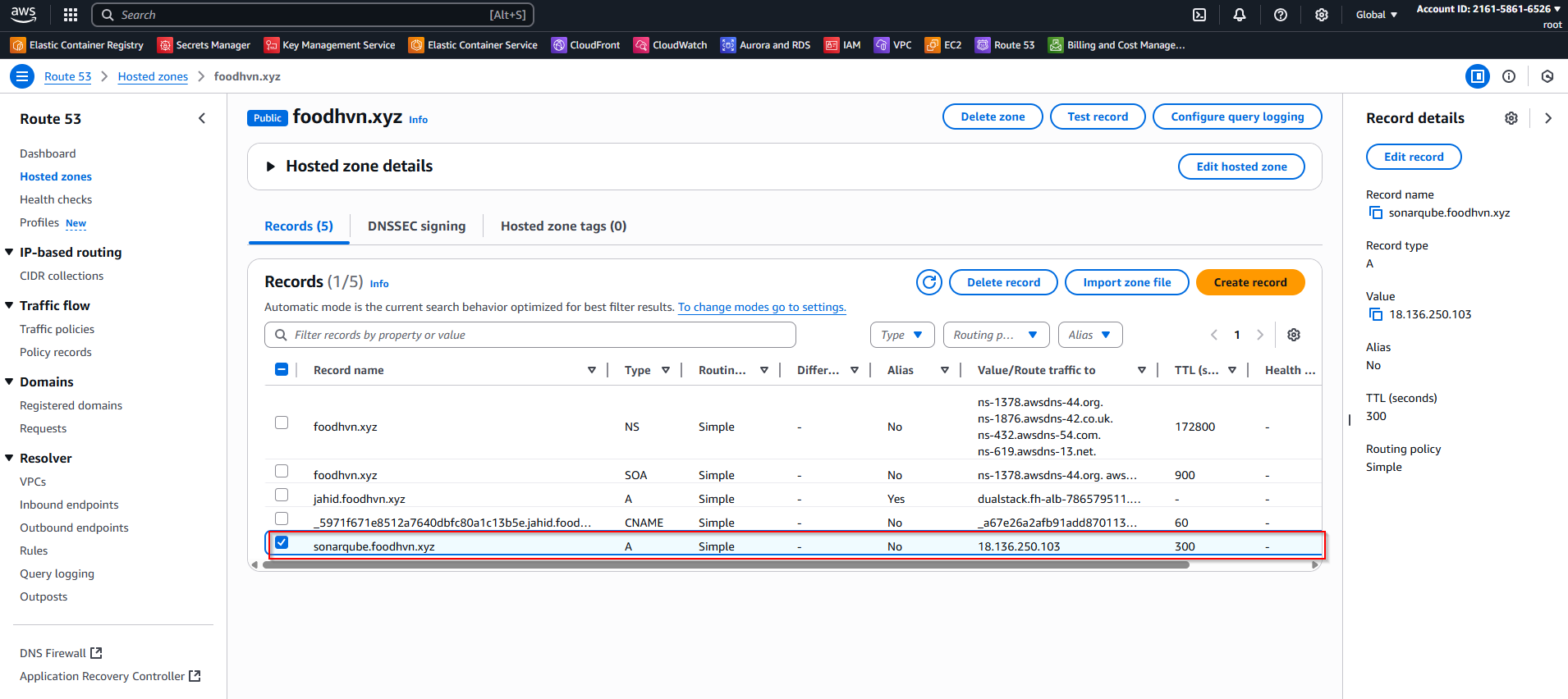
Active: active (running) since Thu 2024-12-26 14:12:47 WAT; 2h 54min ago

Process: 1085 ExecStart=/opt/sonarqube/bin/linux-x86-64/sonar.sh start (code=exited, status=0/SUCCESS)

Main PID: 1108 (java)

Restart the server to apply the SonarQube installation changes.

$ sudo reboot now

**Create a record in R53:  
**

**Install reverse proxy :**Install Nginx

$ sudo apt update

$ sudo apt install nginx -y

Install Certbot

$ sudo apt install certbot python3-certbot-nginx -y

Configure Nginx Reverse Proxy

$ sudo vi /etc/nginx/sites-available/sonarqube

Paste this (replace sonarqube.foodhvn.xyz):

$ server {

listen 80;

server\_name sonarqube.foodhvn.xyz;

location / {

proxy\_pass http://localhost:9000;

proxy\_set\_header Host $host;

proxy\_set\_header X-Real-IP $remote\_addr;

proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

proxy\_set\_header X-Forwarded-Proto $scheme;

}

}

Save and close

Enable the config:

$ sudo ln -s /etc/nginx/sites-available/sonarqube /etc/nginx/sites-enabled/

$ sudo nginx -t

$ sudo systemctl reload nginx

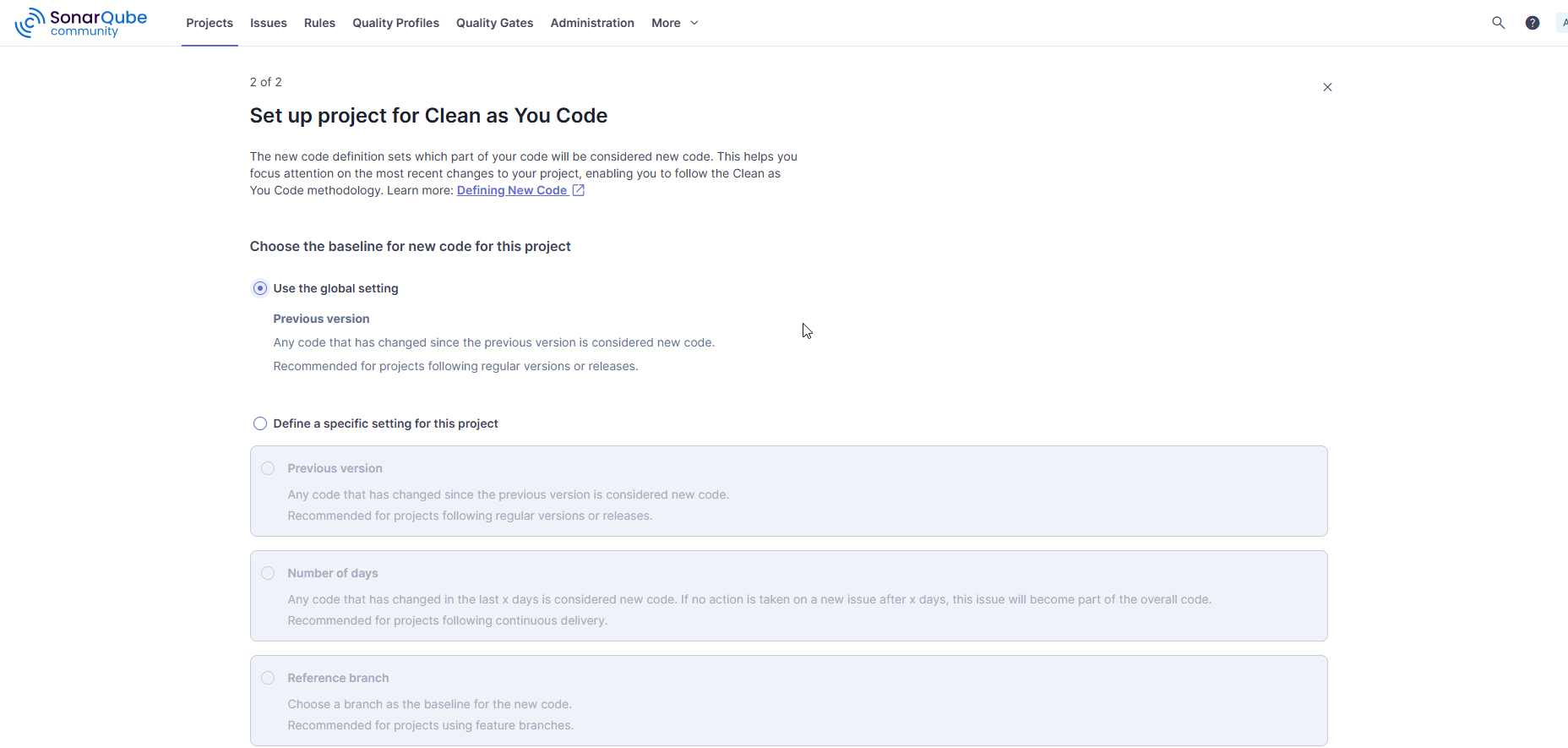
Obtain SSL Certificate

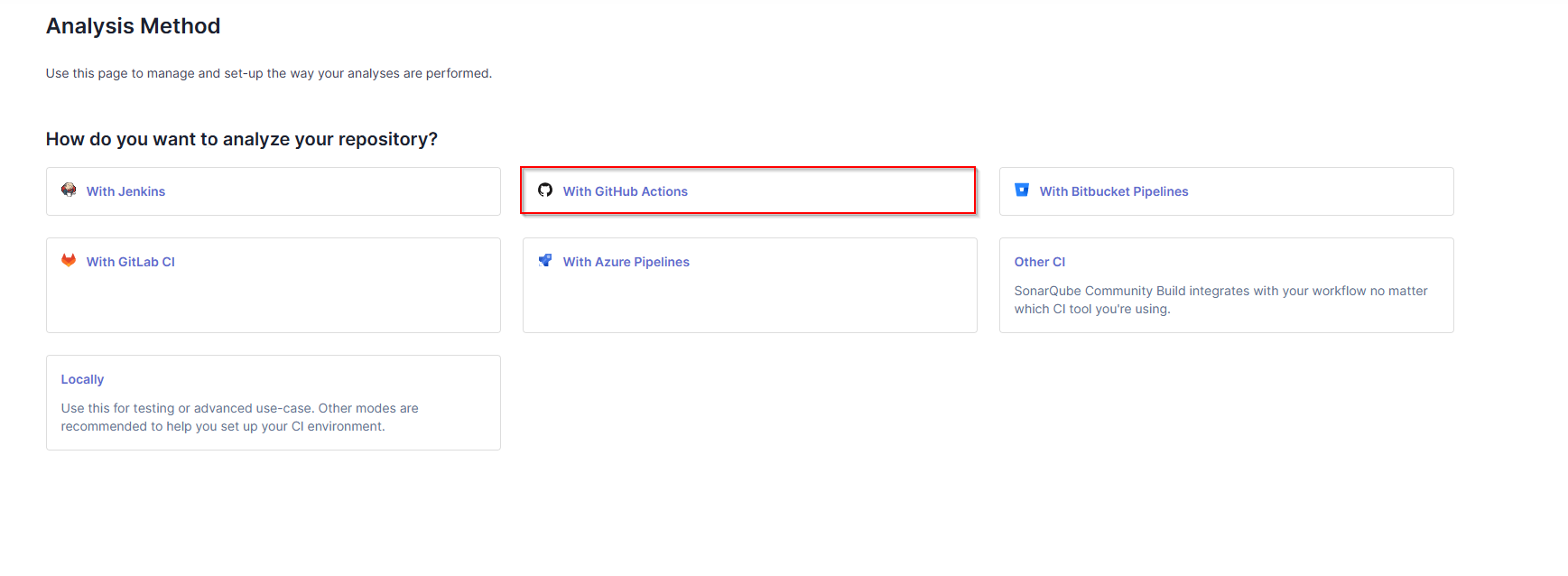
$ sudo certbot --nginx -d sonarqube.foodhvn.xyz

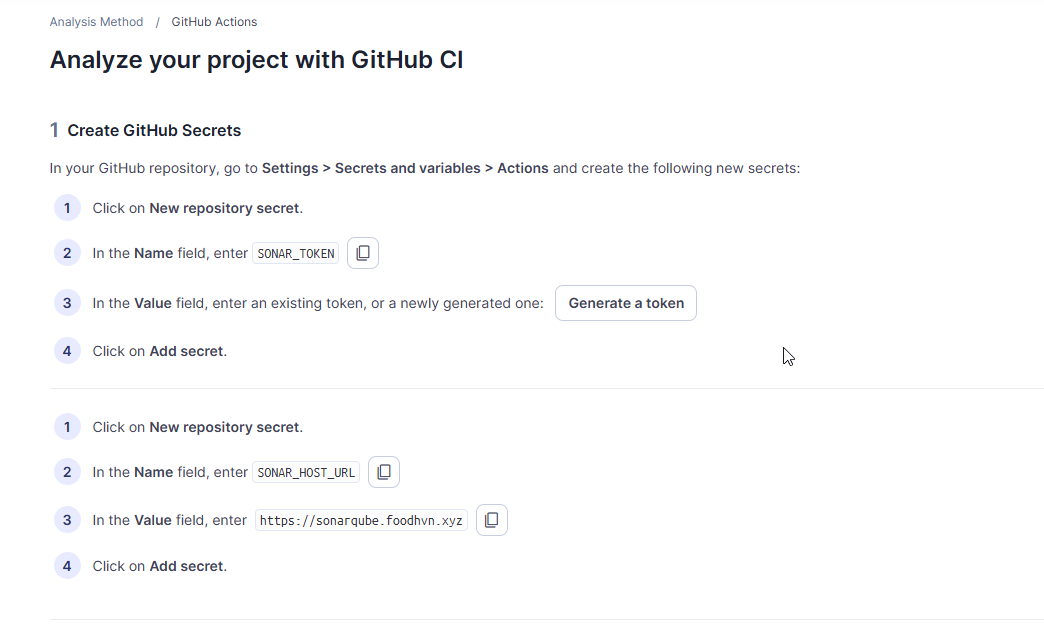
Here is the final Output

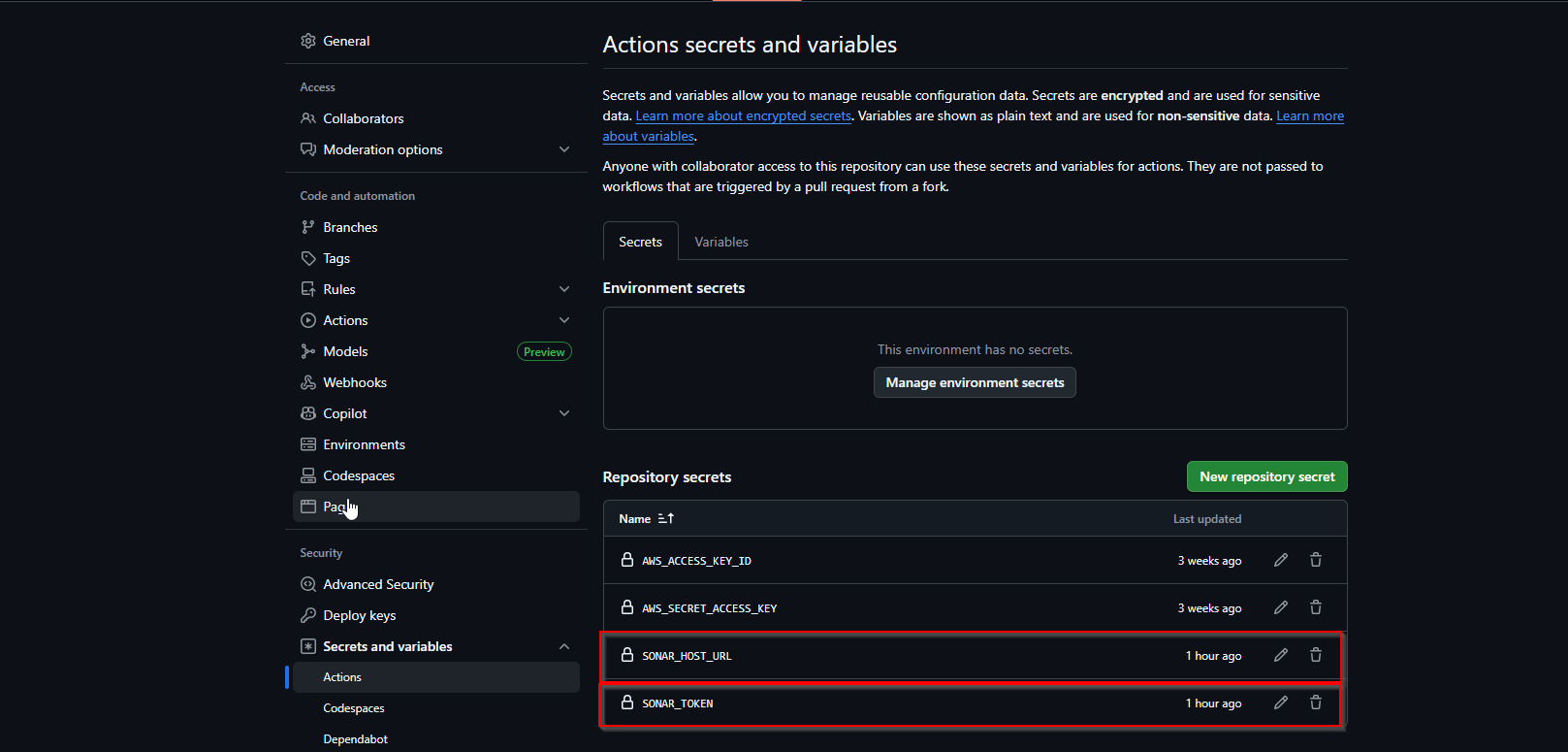
## 

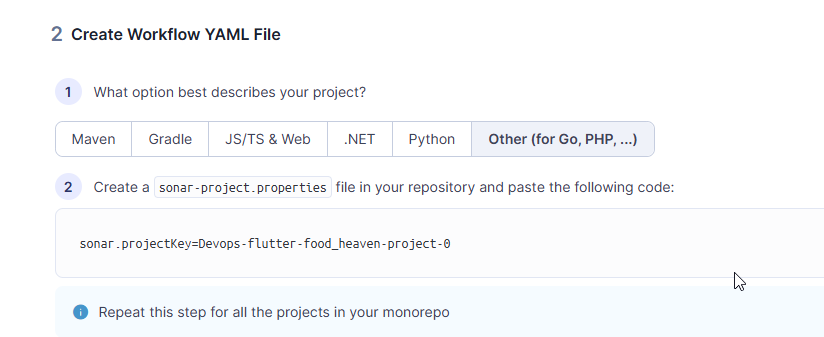
## 

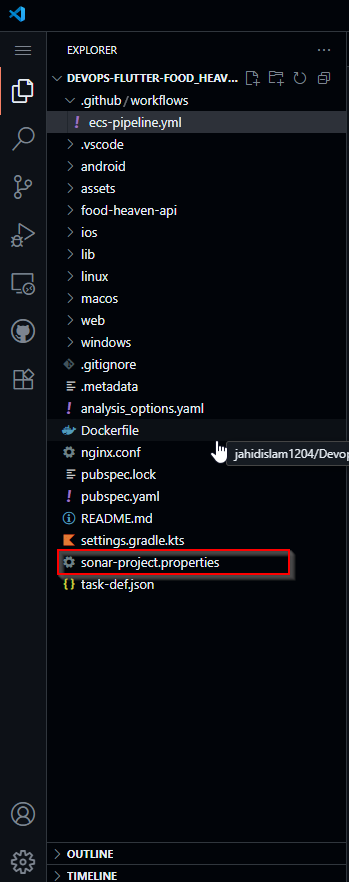






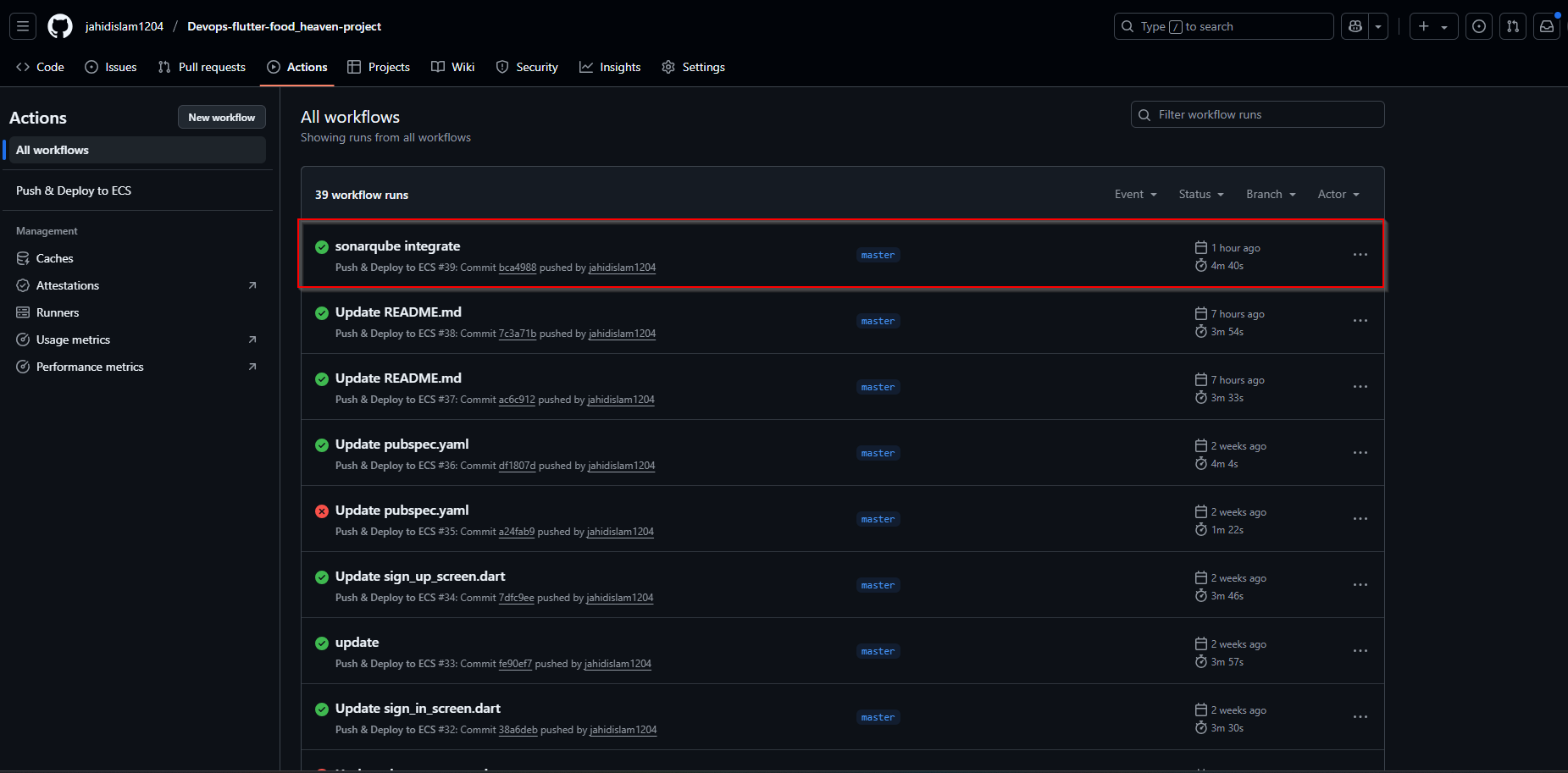


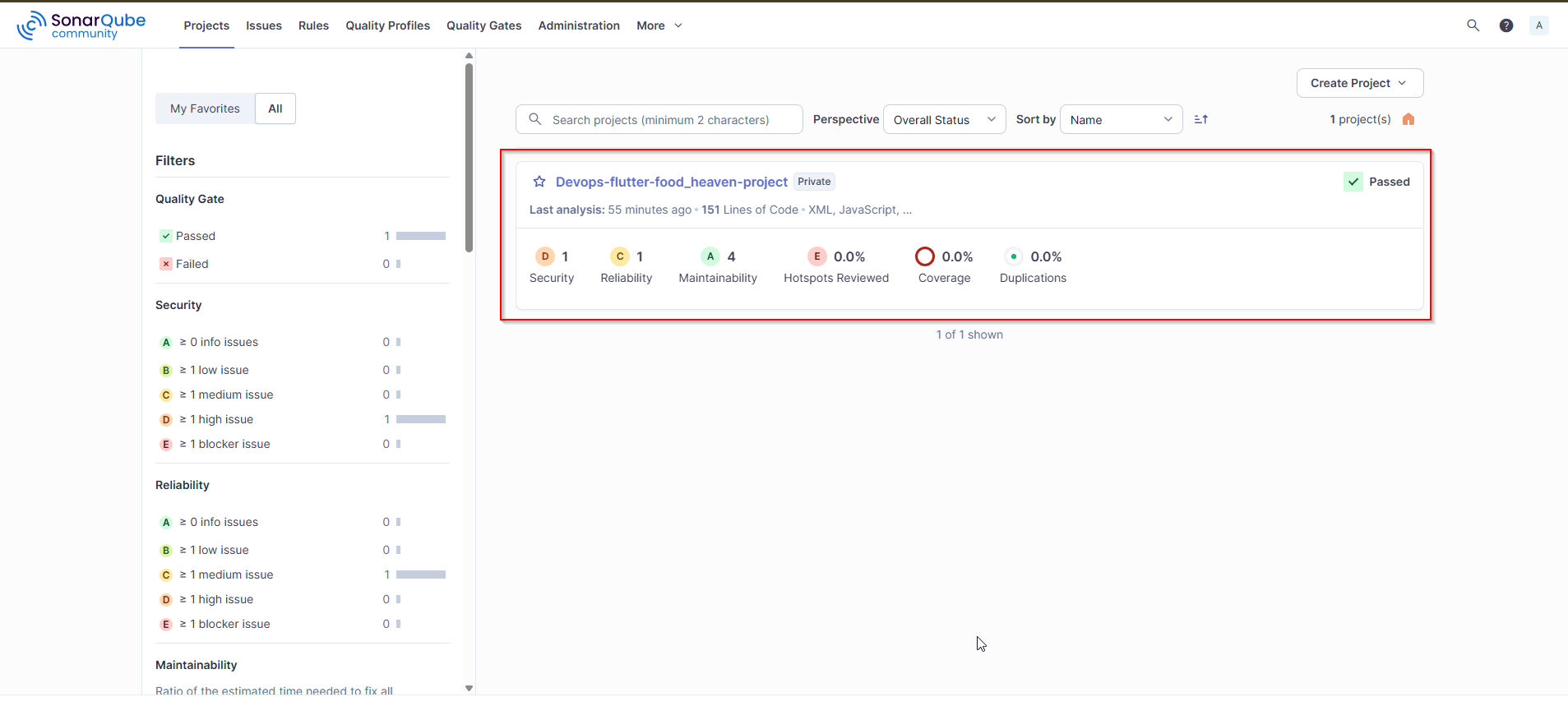






## 





# Notes & To-Do

- Restrict public access — consider VPN or IP allowlisting.  
- Coverage and test integration pending.  
- Backup SonarQube config and DB regularly.  
- Create read-only SonarQube users for team members.