

SonarQube

É uma ferramenta que consegue identificar as alterações que queremos fazer no nosso código (**pull requests**) e quando identifica ela analisa todo o código e verifica se não tem erros, faz testes, e também pode verificar se há algum tipo de gambiarra no código, verifica se tem as melhores práticas aplicadas, verifica se tem questões de vulnerabilidade de segurança, etc.

Para configurar a gente baixa o **sonar** como arquivo zip e coloca na pasta do **windows** e depois configura como **variável de ambiente path** e depois a gente roda o **docker** com o comando **docker run -d --name sonarqube -p 9000:9000 sonarqube:[Versão desejada]**

Depois disso iremos entrar no <http://localhost:9000> para fazer login (de primeiro acesso o usuário e senha ficam como 'admin') no **SonarQube** e criar um projeto e depois executar os comandos que ele passará de acordo e logo em seguida irá automaticamente atualizar a página e mostrar o resultado quando for concluído a 'inspeção' do **Sonar Scanner**.

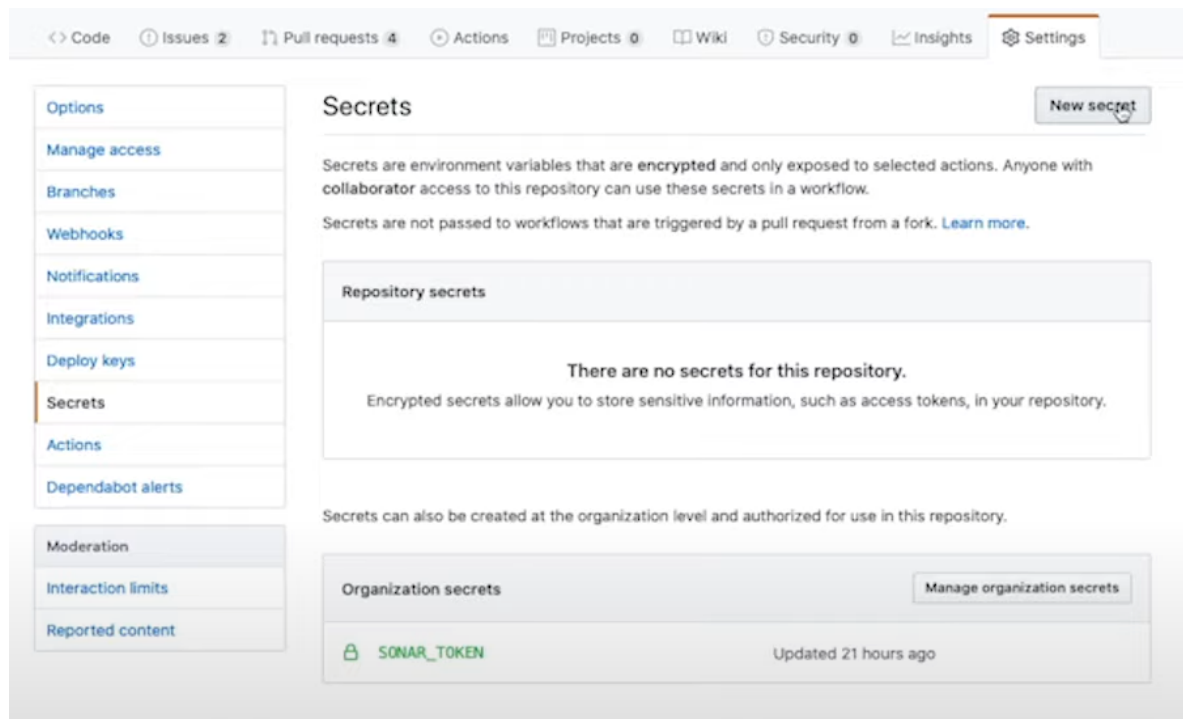
Para rodar em projetos do **GitHub** a gente cria um arquivo **.yaml** com os comandos para usar **actions** do **GitHub** para rodar o OS desejado, baixar a linguagem, checar o código, criar o **.env** para projetos que tem variáveis de ambiente, para executar os teste unitários também e para funcionar o **Sonar Scanner**(para isso vamos usar 2 variáveis de ambiente que 1 é o **GitHub Token** que é um **Secret Token** do **GitHub** que todo mundo já tem e já é embutida outra é o **Sonar Token** que precisamos gerar no **SonarQube** e criar um **Secret** no nosso repositório do **GitHub**).

EXEMPLO DE .YAML

32 lines (28 sloc) | 769 Bytes

```
1  on: [push, pull_request]
2  name: Test
3  jobs:
4    test:
5      strategy:
6        matrix:
7          go-version: [1.14.x]
8          platform: [ubuntu-latest]
9      runs-on: ${{ matrix.platform }}
10     steps:
11       - name: Install Go
12         uses: actions/setup-go@v1
13         with:
14           go-version: ${{ matrix.go-version }}
15
16       - name: Checkout code
17         uses: actions/checkout@v2
18
19       - name: Creating .env
20         uses: canastro/copy-file-action@master
21         with:
22           source: ".env.example"
23           target: ".env"
24
25       - name: Test
26         run: go test ./...
27
28       - name: SonarCloud Scan
29         uses: sonarsource/sonarcloud-github-action@master
30         env:
31           GITHUB_TOKEN: ${{ secrets.GITHUB_TOKEN }}
32           SONAR_TOKEN: ${{ secrets.SONAR_TOKEN }}
```

AONDE CONFIGURAR A SECRET TOKEN DO SONAR.



Depois disso quando executarmos um **pull request** ou **push** irá rodar os **testes unitários** e o **SonarQube**, vemos o progresso na aba **Actions** e o resultado em **pull requests**.

Tem como adicionar o selo de qualidade do Sonar no painel na opção **Get project badges**

Project Information

Project Information

Description

CRUD-Spring

No tags

Reliability

Lines of Code

191 XS

Security

Quality Gate used

(Default) Sonar way

Quality Profiles used

(Java) Sonar way

(XML) Sonar way

Security Review

External Links

Project's Website

Sources

Maintainability

Project Key

crud-spring-key

Copy

0

ated Blocks

Get project badges

Set notifications

E também adicionar notificações para determinadas ações em [Set notification](#).

Project Information

Project Information

Description

CRUD-Spring

No tags

Reliability

Lines of Code

191 XS

Security

Quality Gate used

(Default) Sonar way

Quality Profiles used

(Java) Sonar way

(XML) Sonar way

Security Review

External Links

Project's Website

Sources

Maintainability

Project Key

crud-spring-key

Copy

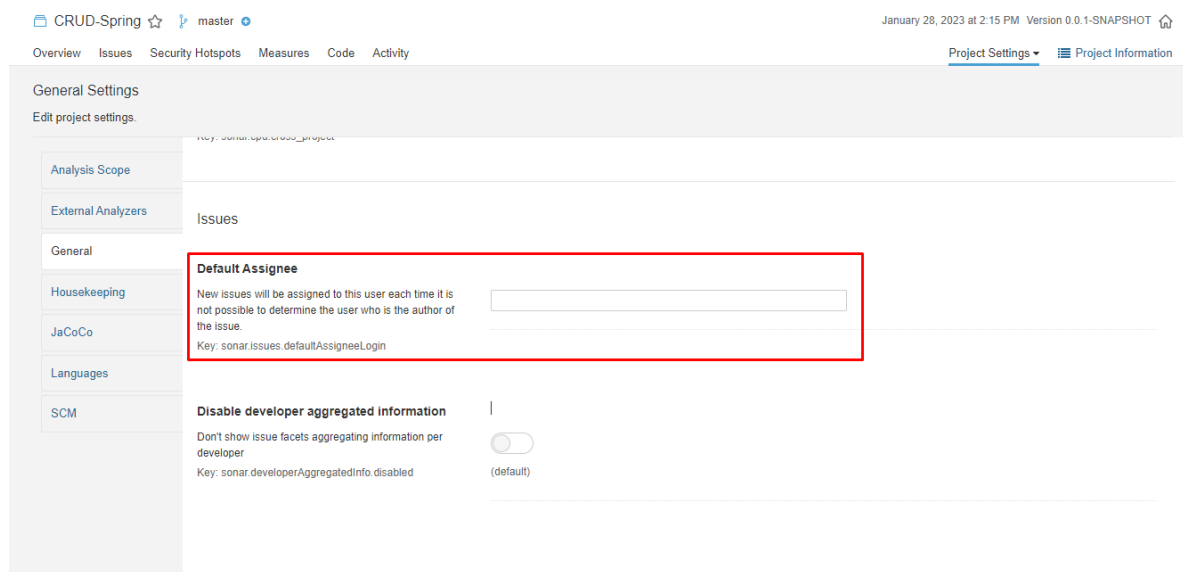
0

ated Blocks

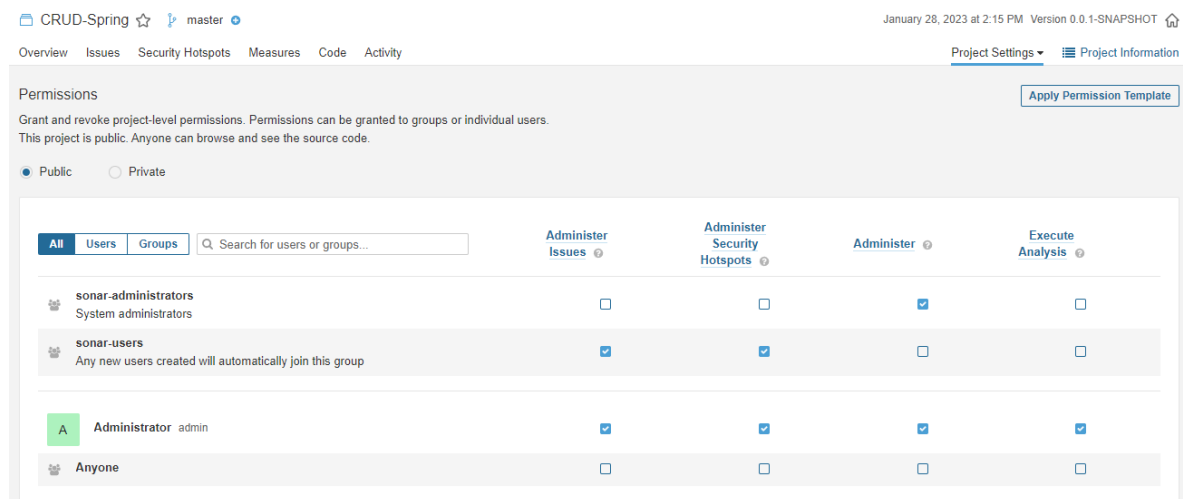
Get project badges

Set notifications

Também podemos configurar um usuário que receberá a culpa para quando não for identificado o usuário que ocasionou um problema de fato.



Temos também como dar permissões para equipes ou pessoas específicas.



Temos como criar [WebHooks](#) para notificar serviços externos quando uma análise de projeto é feita, uma solicitação [HTTP POST](#) incluindo uma carga [JSON](#) é enviada para cada um dos [URLs](#) fornecidos.



Em ALM integrations configuramos aonde o nosso resultado do Sonar pode ser exibido além do painel do mesmo, também temos como ativar a autenticação por outras ferramentas (github, azure, etc).

Administration

Configuration ▾ Security ▾ Projects ▾ System Marketplace

General Settings

Edit global settings for this SonarQube instance.

ALM Integrations

Analysis Scope

External Analyzers

General

Housekeeping

JaCoCo

Languages

New Code

SCM

Security

Technical Debt

Integration configurations

ALM integrations allow SonarQube to interact with your ALM. This enables things like authentication, or providing analysis details and a Quality Gate to your Pull Requests directly in your ALM provider's interface.

GitHub
 Bitbucket
 Azure DevOps
 GitLab

Create your first GitHub configuration to start analyzing your repositories on SonarQube.

[Create configuration](#)

GitHub Authentication

In order to enable authentication on GitHub.com or GitHub Enterprise:

- SonarQube must be publicly accessible through HTTPS only
- The property 'sonar.core.serverBaseUrl' must be set to this public HTTPS URL
- In your GitHub profile, you need to create a Developer Application for which the 'Authorization callback URL' must be set to `'<value_of_sonar_core_serverBaseUrl_property>/oauth2/callback'`.

Enabled

Enable GitHub users to login. Value is ignored if client ID and secret are not defined.

Key: sonar.auth.github.enabled

☒
(default)

Preços

100,000 - \$150 per year/instance

250,000 - \$1,200 per year/instance

500,000 - \$2,400 per year/instance

1 Million - \$4,000 per year/instance

2 Million - \$8,000 per year/instance

5 Million - \$23,000 per year/instance

10 Million - \$48,000 per year/instance

20 Million - \$65,000 per year/instance

Encryption:

Agora no Sonar tem como criptografar as propriedades com uma chave secreta.

Secret Key

How To Use

- Store the secret key in the file `~/sonar/sonar-secret.txt` of the server. This file can be relocated by defining the property `sonar.secretKeyPath` in `conf/sonar.properties`
- Restrict access to this file by making it readable and by owner only
- Restart the server if the property `sonar.secretKeyPath` has been set or changed.
- For each property that you want to encrypt, generate the encrypted value and replace the original value wherever it is stored (configuration files, command lines).

System:

Podemos verificar tudo sobre o sistema Sonar, inclusive memória sendo usada, processador, baixar logs por lá, etc.

The screenshot shows the SonarQube Administration interface. The top navigation bar includes 'sonarqube' logo and tabs for 'Projects', 'Issues', 'Rules', 'Quality Profiles', 'Quality Gates', and 'Administration'. The 'Administration' tab is active, and the 'System' sub-tab is selected. Below the navigation bar, there's a 'System Info' section with a warning message: 'This server ID is valid only for the embedded database, which should be considered disposable. Consider configuring an external database for long-term use prior to requesting your license.' The 'Server ID' is redacted, and the 'Version' is '9.9.0.65466'. There are buttons for 'Download Logs' and 'Download System Info'. Below this, a table lists system components: 'System' (Status is up), 'Web', 'Compute Engine', and 'Search Engine'. At the bottom, there's a 'Logs level: INFO' section with a dropdown menu for 'Download Logs' showing options: 'Main Process', 'Compute Engine', 'Search Engine', 'Web Server', and 'Access Logs'.

sonarqube Projects Issues Rules Quality Profiles Quality Gates Administration

Administration

Configuration Security Projects System Marketplace

System Info

Logs level: INFO Download Logs Download System Info

This server ID is valid only for the embedded database, which should be considered disposable. Consider configuring an external database for long-term use prior to requesting your license.

Server ID [REDACTED] Copy ID information

Version 9.9.0.65466

System	Status is up
Web	
Compute Engine	
Search Engine	

Logs level: INFO Download Logs Download System Info

- Main Process
- Compute Engine
- Search Engine
- Web Server
- Access Logs

Marketplace:

Podemos agora baixar plugins para o Sonar através do Marketplace, isso pode ser muito útil em projetos que tem peculiaridades.

Explicação de como instalar plugins pela própria documentação do Sonar:

<https://docs.sonarqube.org/latest/setup-and-upgrade/install-a-plugin/>

sonarqube
Projects
Issues
Rules
Quality Profiles
Quality Gates
Administration
?
Search for projects...
A

Administration

Configuration
Security
Projects
System
Marketplace

[Request a free trial](#)
[Request a free trial](#)
[Request a free trial](#)

Plugins

Plugins available in the Marketplace are not provided or supported by SonarSource. Please reach out directly to their maintainers for support.

Installation of plugins

Plugins are not provided by SonarSource and are therefore installed at your own risk. SonarSource disclaims all liability for installing and using such plugins. You can install plugins directly from the list below after you acknowledge the risk.

[I understand the risk](#)

All
Installed
Updates Only

Search by features, tags, or categories...

1C (BSL) Community Plugin LANGUAGES	1.11.0 13 new rules, bug fixes, support new reporters ...	Homepage Issue Tracker Licensed under GNU LGPL v3 Developed by 1c-syntax
AEM Rules for SonarQube EXTERNAL ANALYZERS	1.6 SonarQube 8.9 LTS compatibility release due to underlying Java plugin API changes ...	Homepage Issue Tracker Licensed under The Apache Software Li... Developed by Wunderman Thompson Technology
Ansible Lint EXTERNAL ANALYZERS	2.5.1 Support for SonarQube 9.2 ...	Homepage Issue Tracker Licensed under Apache License, Versio...
Apigee EXTERNAL ANALYZERS	3.0.2 Support for SQ 9.7+ ...	Homepage Issue Tracker Licensed under Apache License, Versio... Developed by Crédit Mutuel Arkéa
Azure Active Directory (AAD) Authentication Plug-in for SonarQube INTEGRATION	1.3.2 Updates commons-text to fix CVE-2022-42889. ...	Homepage Issue Tracker Licensed under The MIT License (MIT) Developed by ALM DevOps Rangers
CVS INTEGRATION	1.1.1 Fix classnotfound error ...	Homepage Issue Tracker

Hosts and locations:

Para um desempenho ideal, o servidor SonarQube e o banco de dados devem ser instalados em hosts separados, e o host do servidor deve ser dedicado. O servidor e os hosts do banco de dados devem estar localizados na mesma rede. Todos os hosts devem estar sincronizados com o tempo.