

Copy of How to Perform a Static Analysis

Purpose

The purpose of this document is to provide resources and steps to perform or request a Static Analysis.

It is highly recommended to execute this assessment continuously while coding the application so that solving security flaws impacts only a few lines of code.

- [Purpose](#)
 - [Automatic Process](#)


Automatic Process

Create SonarQube Token

Follow these steps to create a token on SonarQube:


1. Log in to the SonarQube instance.
2. Click on the avatar with your initial in the top right corner to display a dropdown menu.
3. Select **My Account** from the dropdown menu to display your profile information.
4. Select the **Security** tab.
5. Enter the desired name on the **Enter Token Name** parameter.
6. Click **Generate** to create your token.
7. Copy your token to a safe location.

You created your SonarQube token.

 A SonarQube token is a string of characters used to authenticate a user without entering their credentials. Use the SonarQube token to run analyses or invoke web services through API requests or by adding them as the authentication method in scripts.

Set Github Secrets

Secrets are encrypted variables you create in an organization, repository, or repository environment. The secrets that you create are available to use in GitHub Actions workflows.

 You must be the repository owner to create the repository secrets.

Follow these steps to create repository secrets:

1. Login to GitHub.
2. Navigate to the main page of the repository.
3. Click **Settings** under your repository name.
4. Locate the **Security** section on the sidebar.
5. Select **Secrets and variables > Actions** to display the current Actions secrets and variables.
6. Click **New Repository Secret**.
7. Add the following secrets:

Name	Secret
------	--------

SONAR_TOKEN	<Token-created-on-the-Create-SonarQube-Token-section>
SONAR_HOST	http://18.222.178.31:9000

You created your repository secrets.

Configure SonarQube Project

A configuration file indicates SonarQube the analysis settings for your project. To set the analysis settings, create a `sonar-project.properties` file on the root folder of your repository. Include the following elements in your configuration file:

```
1 sonar.projectKey=<project-name-on-sonarqube>
2 sonar.sources=<path-to-analyze>
```

The following table describes the key values from the SonarQube project configuration file.

Key	Description	Type
<code>sonar.projectkey</code>	Unique name for your project in the SonarQube instance.	String.
<code>sonar.sources</code>	Path is relative to the <code>sonar-project.properties</code> file. Defaults to <code>.</code>	String.

Table 1. Configuration File Key Values

i The example above is a simple analysis configuration file. In case you need to add specific analysis parameters, you can see the [analysis parameters](#) documentation. Review the following documentation to create your project file if your project uses:

- [C/C++/Objective-C](#)
- [Gradle](#)
- [.NET](#)
- [Maven](#)

Create GitHub Actions Workflow

GitHub Actions is a continuous integration and continuous delivery (CI/CD) platform that allows you to automate your build, test, and deployment pipeline.


Follow these steps to automate the static analysis procedure:

1. Create a YAML file with the following route and name `.github/workflows/sonar-scan.yml` on the root folder of your repository. See the example below:

```
1 on:
2   # Trigger analysis when pushing to your main branches, and when creating a pull request.
3   push:
4     branches:
5       - main
6   pull_request:
7     types: [opened, synchronize, reopened]
8
9 name: Main Workflow
```

```
10 jobs:
11   sonarqube:
12     runs-on: ubuntu-latest
13     steps:
14     - uses: actions/checkout@v3
15       with:
16         # Disabling shallow clone is recommended for improving relevancy of reporting
17         fetch-depth: 0
18     - name: SonarQube Scan
19       uses: sonarsource/sonarqube-scan-action@master
20     env:
21       SONAR_TOKEN: ${ secrets.SONAR_TOKEN }
22       SONAR_HOST_URL: ${ secrets.SONAR_HOST }
```

You automated the static analysis procedure.

 For further information on workflows, you can see the [GitHub actions documentation](#).