

MENU

+ Follow

## Orlando's Blog

# Integrating SonarQube into GitLab CI

**Orlando Thöny**Published on **Oct 20, 2020** 2 min read

How to use SonarScanner CLI.

This is an example of how you can use the SonarScanner CLI. For example, if you want to scan a PHP application. There are also alternatives: Gradle & Maven.

Create a file called `sonar-project.properties` inside of your repository root. As stated in the [SonarQube GitLab CI documentation](#).

COPY

```
# SonarQube server
# sonar.host.url & sonar.login are set by the Scanner CLI.
# See https://docs.sonarqube.org/latest/analysis/gitlab-cicd/.

# Project settings.
sonar.projectKey=my-project
sonar.projectName=My project
sonar.projectDescription=My new interesting project.
sonar.links.ci=https://gitlab.com/my-account/my-project/pipelines
sonar.links.issue=https://jira.example.com/projects/MYPROJECT

# Scan settings.
sonar.projectBaseDir=.
# Define the directories that should be scanned. Comma separated.
sonar.sources=./src,./resources,./web

sonar.test.inclusions=**/*Test.php
sonar.php.coverage.reportPaths=./coverage/lcov.info
sonar.php.file.suffixes=php
sonar.sourceEncoding=UTF-8

sonar.exclusions=,**/coverage/**

# Fail CI pipeline if Sonar fails.
sonar.qualitygate.wait=true
```

Add a SonarQube stage to your `*gitlab-ci.yml` file. I configured it to only run on the `Git master` branch. Because I'm using the SonarQube CommunityEdition — which only supports analyzing one branch per repository.\*

[COPY](#)

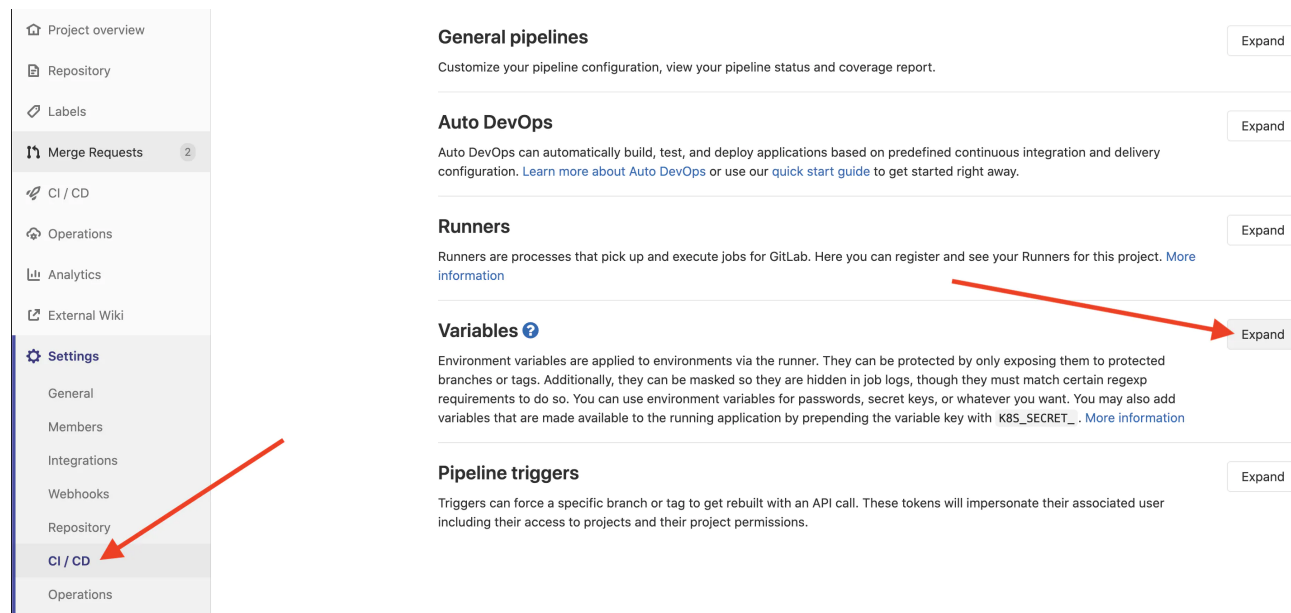
```
stages:
  - analyze

analyze:sonar:
  stage: analyze
  image:
    name: sonarsource/sonar-scanner-cli:4.5
    entrypoint: [""]
  variables:
    # Defines the location of the analysis task cache
    SONAR_USER_HOME: "${CI_PROJECT_DIR}/.sonar"
    # Shallow cloning needs to be disabled.
    # See https://docs.sonarqube.org/latest/analysis/gitlab-cicd/.
    GIT_DEPTH: 0
  cache:
    key: "${CI_JOB_NAME}"
    paths:
      - .sonar/cache
  script:
    - sonar-scanner
  rules:
    # SonarQube CommunityEdition only supports analyzing a single branch.
    # So only run on master.
    - if: '$CI_COMMIT_BRANCH == "master"'
```

```
when: on_success  
- when: never
```

Add the following variables via the GitLab CI UI. *Keep in mind not to commit any credentials to your Git repository.*

1. Go to *Settings > CI / CD*
2. Expand *Variables*



*Setting GitLab CI variables*

**Add the required Sonar variables:**

SONAR\_HOST\_URL :

### Update variable ×

**Key**

**Value**

↻

**Type** **Environment scope**

Var

All (default)

**Flags**

☒ **Protect variable** ?  
Export variable to pipelines running on protected branches and tags only.

☒ **Mask variable** ?  
Variable will be masked in job logs. Requires values to meet regular expression requirements. [More information](#)

CancelDelete variableUpdate variable

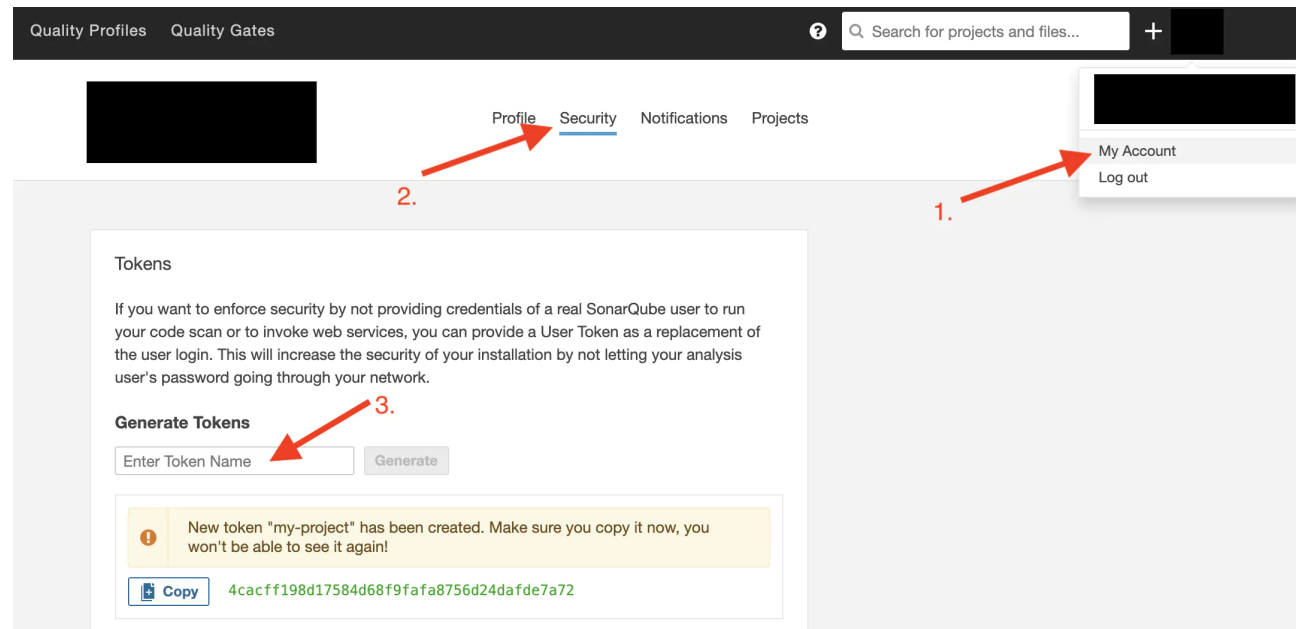
*SONAR\_HOST\_URL configuration*

SONAR\_TOKEN :

First off, we need a token. To get one, log into you Sonar instance and create a new one:

1. Go to *My Account*

2. Click the *Security* tab
3. Enter a token name, and click *Generate*
4. Copy the generated token




## Add variable ×

**Key**

SONAR\_TOKEN

**Value**


1bf73d72bfd34485257bab2579g27036gf4893645 

**Type** **Environment scope**


Var

All (default)

**Flags**

☒ **Protect variable** 

Export variable to pipelines running on protected branches and tags only.

☒ **Mask variable** 

Variable will be masked in job logs. Requires values to meet regular expression requirements. [More information](#)

Cancel

Add variable

*SONAR\_TOKEN configuration*

Now your project will show up in SonarQube after the first GitLab CI pipeline run.

#gitlab


#continuous-integration






Like


READ NEXT




Orlando Thöny




HashiCorp  
**Terraform**




"redis" Module State




google\_redis\_instance.my\_redis\_instance




random\_password.my\_redis\_pw



"databases" Module State



module.postgresql



google\_sql\_database\_instance.postgresql

Before

<https://thoeny.dev/integrating-sonarqube-into-gitlab-ci>

8/10





## How to move Terraform root module state inside other root module

I ran into the problem that I wanted to make an existing root module (I'll call this source) a child...



## Orlando Thöny

```

10 Pulling docker image docker:latest ...
11 Using worker image sha256:33647722d6479f9c3d8d8e6a3d4b6e22d8b8b69fe for docker:latest with digest docker:sha256:8a2c323d46f32279a3b6a89a342c796f823796733d468d2c26e12e6dc
12 Inspecting container
13 Running as runner:72889763 project:32046473 concurrent:8 via runner:72889763 src:3d2788862_62b747d...
14 Getting source from git repository
15 $ echo "HELLO WORLD" >hello.txt
16 Pushing changes with git depth set to 0...
17 Initialized empty Git repository in /build/dockerlands/empty/empty-web/.git/
18 Created fresh repository
19 Checking out 3d43d0e as master...
20 Updating Git submodule state
21 Executing "run_script" stage of the job script
22 Pulling docker image sha256:33647722d6479f9c3d8d8e6a3d4b6e22d8b8b69fe for docker:latest with digest docker:sha256:8a2c323d46f32279a3b6a89a342c796f823796733d468d2c26e12e6dc
23 $ docker run --rm hello-world:latest
24 Unable to find image 'hello-world:latest' locally
25 latest: Pulling from library/hello-world
26 3d43d0e127a20: Pulling fs layer
27 3d43d0e127a20: Verifying Checksum
28 3d43d0e127a20: Download complete
29 3d43d0e127a20: Pull complete
30 Digest: sha256:3d43d0e127a20179f15a16a7933d468d2c26e12e6dc33d468d2c26e12e6dc
31 Status: Downloaded newer image for hello-world:latest
32 hello from Docker!
33 This message shows that your installation appears to be working correctly.
34 To generate this message, Docker took the following steps:
35 1. The Docker client contacted the Docker daemon.
36 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
37    (amd64)
38 3. The Docker daemon created a new container from that image which runs the
39    executable that produces the output you are currently seeing.
40 4. The Docker daemon streamed that output to the Docker client, which sent it
41    to your terminal.
42 To try something more ambitious, you can run an Ubuntu container with:
43 $ docker run -ti ubuntu bash
44 Share images, automate workflows, and more with a new Docker ID!
45 https://cloud.docker.com
46 For more examples and ideas, visit:

```

## GitLab CI: Docker in Docker (DinD) on gitlab.com Shared Runners

How to run DinD jobs on gitlab.com Shared Runners Gitlab provides shared runners for the repositorie...



## Orlando Thöny

## GitLab CI: Security Scan using KICS

Scan Kubernetes, Helm, Terraform, Docker, Ansible & AWS CloudFormation Code for security vulnerabili...

No Comments Yet



Add a comment

---

© 2021 Orlando's Blog   Privacy   Terms

Proudly part of

