**SonarQube Integration**

**1. Prerequisites**

Make sure you have the following installed on your system:

* **Docker** and **Docker Compose**
* **GitLab Runner** (Shell or Docker executor)
* **GitLab CE/EE instance** (local or remote)
* **Access to DockerHub** (for image pushing)

**2. Install SonarQube using Docker**

Run the following command to deploy SonarQube locally:

docker run -d \

--name sonarqube \

--restart always \

-p 9000:9000 \

sonarqube:lts-community

This will:

* Pull the Long-Term Support (LTS) SonarQube Community Edition image
* Expose it on [**http://localhost:9000**](http://localhost:9000)
* Auto-start on system reboot

**3. Verify SonarQube**

Check container status:

docker ps

If it’s running, open in your browser:

http://localhost:9000

Default login credentials:

Username: admin

Password: admin

After login → set a **new password**.

**4. Create a Project and Token**

1. Log into **SonarQube Dashboard**
2. Go to: **Projects → Manually Create Project**
3. Give a name (e.g. node-todo-cicd)
4. Choose **“Use existing CI”** → select **GitLab**
5. Generate a **token**:
   * Navigate to **My Account → Security**
   * Click **Generate Token** → Copy it
6. Save:
   * **SONAR\_HOST\_URL:** http://<your-server-ip>:9000
   * **SONAR\_TOKEN:** <your-generated-token>

**5. Register GitLab CI/CD Variables**

In your GitLab project:

* Go to **Settings → CI/CD → Variables**
* Add:

| **Key** | **Value** | **Masked** | **Description** |
| --- | --- | --- | --- |
| SONAR\_HOST\_URL | http://<your-server-ip>:9000 | ✅ | SonarQube Server URL |
| SONAR\_TOKEN | <generated-token> | ✅ | Sonar Token |
| DOCKERHUB\_USER | <your-dockerhub-username> | ✅ | For Docker Push |
| DOCKERHUB\_PASS | <your-dockerhub-password> | ✅ | For Docker Push |

**6. Install SonarScanner (optional for local use)**

If you want to test scanning manually:

apt-get update -y

apt-get install unzip -y

# Download latest scanner (replace version if needed)

wget https://binaries.sonarsource.com/Distribution/sonar-scanner-cli/sonar-scanner-cli-6.1.0.4477-linux.zip

unzip sonar-scanner-cli-6.1.0.4477-linux.zip

mv sonar-scanner-6.1.0.4477-linux /opt/sonar-scanner

# Add to PATH

export PATH=$PATH:/opt/sonar-scanner/bin

# Verify

sonar-scanner --version

**7. GitLab CI/CD Pipeline (.gitlab-ci.yml)**

Here’s your final **production-ready pipeline**:

stages:

  - build

  - sonar\_scan

  - push\_to\_dockerhub

  - deploy

build\_job:

  stage: build

  script:

    - docker build -t node-app:latest .

sonarqube-check:

  stage: sonar\_scan

  image:

    name: sonarsource/sonar-scanner-cli:latest

    entrypoint: [""]

  variables:

    SONAR\_USER\_HOME: "${CI\_PROJECT\_DIR}/.sonar"  # Defines the location of the analysis task cache

    GIT\_DEPTH: "0"  # Tells git to fetch all the branches of the project, required by the analysis task

  cache:

    key: "${CI\_JOB\_NAME}"

    paths:

      - .sonar/cache

  script:

    - sonar-scanner

  allow\_failure: true

  only:

    - master

push\_job:

  stage: push\_to\_dockerhub

  script:

    - docker login -u $DOCKERHUB\_USER -p $DOCKERHUB\_PASS

    - echo "hi 2"

    - docker image tag node-app:latest $DOCKERHUB\_USER/node-app:latest

    - docker push $DOCKERHUB\_USER/node-app:latest

deploy\_job:

  stage: deploy

  script:

    - docker compose up -d

**8. Verify SonarQube Analysis**

* Open your project in SonarQube:  
  http://<your-server-ip>:9000/projects
* Check code quality metrics, vulnerabilities, and bugs

**9. Generate Reports**

If you want to download reports (PDF/HTML), install the **CNES Report plugin**:

⚠️ Make sure the plugin version matches your SonarQube version

1. Download .jar file for your SonarQube version:

docker exec -it sonarqube bash

cd /opt/sonarqube/extensions/plugins

wget https://github.com/cnescatlab/sonar-cnes-report/releases/download/4.1.0/sonar-cnes-report-4.1.0.jar

exit

docker restart sonarqube

1. Go to SonarQube UI → Project → **“More → Generate Report”**

**10. Auto-Start SonarQube After Reboot**

Docker already handles this with --restart always.  
If you want to manually ensure it runs:

sudo systemctl enable docker

sudo systemctl restart docker

docker start sonarqube