

DevOps Course Assignment: Deploy SonarQube and Use It!

Objective

In this assignment, you will deploy SonarQube, add a proxy with a custom domain, and integrate a CI/CD scan job into the pipeline. You will use Docker Compose to deploy SonarQube with a PostgreSQL database, Nginx to add a reverse proxy with a custom domain, and GitLab CI/CD to run the SonarQube scanner on a sample project from the previous assignment (Assignment 1).

Part 1

Read materials

- **Learn SonarQube:** <https://medium.com/@mrdevsecops/introduction-to-sonarqube-and-its-features-27c2e030068a>
- **Getting Started:** <https://dev.to/tacianosilva/running-a-sonarqube-server-and-client-with-docker-7bk>
- **Docker Compose for Production Use:** <https://github.com/SonarSource/docker-sonarqube/blob/master/example-compose-files/sq-with-postgres/docker-compose.yml>

Part 2

Deploy SonarQube

- Refer to the above links and deploy SonarQube using Docker Compose with a PostgreSQL database. Use host directories for data persistence instead of volumes.
- Install Nginx and add a reverse proxy to SonarQube with a custom domain. The custom domain should be "sonarqube.devops.course" and the server should be accessible on port 80. You can edit the **/etc/hosts** file to map the custom domain to the server's IP address or local IP address.
- Ensure that the SonarQube server is accessible at "<http://sonarqube.devops.course>" and running properly. You can use the default admin credentials (**admin/admin**) to log in to the SonarQube server.

Part 3

Add CI/CD scan job

- Create a new repository on GitLab, register a runner again, and push a sample project from epy previous assignment to the repository with a **.gitlab-ci.yml** file.
- Add a SonarQube analysis job (e.g., "**sonar-scan**") to the CI/CD pipeline before the "**deploy**" job. The job should run the SonarQube scanner on the sample project and publish the results to the SonarQube server. Get the SonarQube token from the SonarQube server and add it as a secret variable in the GitLab repository.
- The job should be triggered automatically on every push to the main branch. The CI/CD pipeline should run the "**deploy**" job if and only if the SonarQube analysis job is successful.

Deliverables

A zip file containing the following three directories:

- A sample project with the following files:
 - .gitlab-ci.yml file.
 - sonar-project.properties file.
- A directory containing proof of the following:
 - Screenshots of the SonarQube server running on the custom domain.
 - Screenshots of the SonarQube analysis job running successfully on the GitLab CI/CD pipeline.
 - Screenshots of the SonarQube server showing the analyzed project in the dashboard.
- A directory containing:
 - The Docker Compose file for SonarQube with PostgreSQL.
 - The Nginx configuration file for the reverse proxy.

Grading Criteria

- Proper setup of the SonarQube server and the reverse proxy.
- Correct implementation of the SonarQube analysis job in the CI/CD pipeline.