

Devops Code Description



- 1- The Developer raises a PR to merge code to the main branch
- 2- After the PR is merged, a pipeline is triggered by GitLab.
- 3- A first GitLab-runner will then be invoked.
- 4- The runner launches Sonarqube Scanner CLI container to check the codes. Once the codes have been checked the results are sent to GitLab.
- 5- At this point GitLab sends the results to Sonarqube UI in order to compare the results from the scan with a set of preset quality gates. A final result (pass or fail) is then determined. If the result is determined to be a fail, a notification is sent to the development channel on Slack, however if the code results show to be passed the pipeline proceeds.
- 6- A second runner will then be invoked at this step.
- 7- The 2nd runner will launch an on demand Maven container that will mount the current directory inside the container. Maven will run the following steps, maven clean, validate, compile, test, package, and verify. By completing these steps we can obtain the Artifact.
- 8- At this step a final Runner will be invoked.
- 9- At this stage the Dockerfile in the repository is used.
- 10- That Dockerfile is used to build the docker image(s) that will be utilized to deploy the application.
- 11- (Skipped / Missing)
- 12- Once the image is built, the agent will login to the company repository on Dockerhub to push the image(s) built into the repository.
- 13- At this stage Docker-compose is used to deploy the application.
- 14- Docker-compose will pull the image(s) previously pushed into the company repository from Dockerhub.
- 15- At this step Docker Compose will take the pulled image(s) and deploy the application via the Apache Tomcat web server.
- 16- At this point, the application will be available in the browser
- 17- The Dev team can access the application on the Internet to proceed with further testing and if they are satisfied with the image(s) they will then notify the Testing Team.