

Integration with Git Lab CI

This guide will show you how to integrate Kiuwan into GitLab CI.

Contents:

- [Introduction to the plugin](#)
- [Installing the plugin](#)
 - [1. Insert or add a Kiuwan step in the YAML definition](#)
 - [3. Define credentials as secret variables](#)
 - [4. Perform an analysis](#)

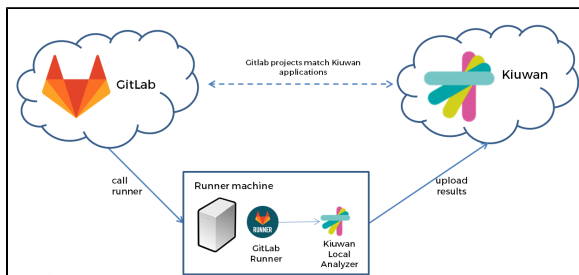
Introduction to the plugin

GitLab is a powerful tool for software development, security, and operations that enables concurrent DevOps, making the software lifecycle faster.

In continuous integration and continuous delivery (CI/CD) environments, it is very common (and recommended) to ensure the security and quality of the software under development.

Kiuwan allows baseline or delivery analyses as a step in the pipeline defined in GitLab.

The Kiuwan analysis will be executed by a GitLab runner.



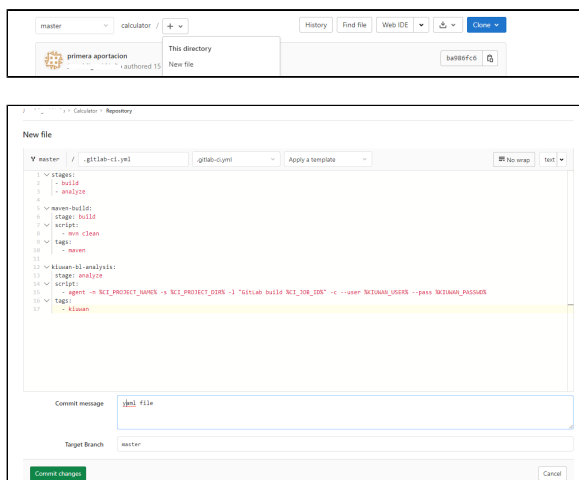
Installing the plugin

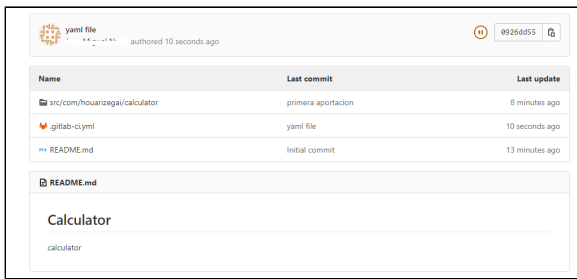
To run a Kiuwan analysis, follow these steps:

1. Insert or add a Kiuwan step in the YAML definition

In the project home, add a YAML file with the steps to invoke Kiuwan.

i The tags used in this step are important because they will be used in the configuration of the local agents (runners).





Here is a very simple case with a Maven step and a Kiuwan baseline analysis:

```
stages:
  - build
  - analyze

maven-build:
  stage: build
  script:
    - mvn clean
  tags:
    - maven

kiuwan-bl-analysis:
  stage: analyze
  script:
    - agent -n %CI_PROJECT_NAME% -s %CI_PROJECT_DIR% -l "GitLab build %
CI_JOB_ID%" -c --user %KIUWAN_USER% --pass %KIUWAN_PASSWD%
  tags:
    - kiuwan
```

2. Install Kiuwan Local Analyzer in the runner machine

As we can see in the previous example, the call to Kiuwan is through a command line.

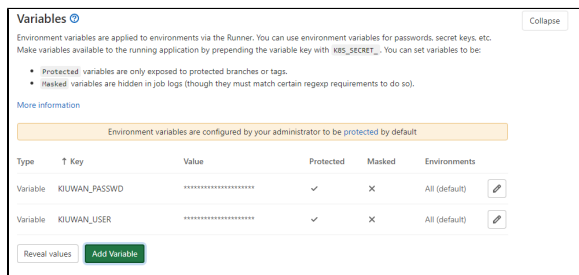
The [Kiuwan Local Analyzer](#) must be installed where the GitLab runner will be installed. [Here](#) are instructions on how to install it.

In addition, add the folder `KLA_HOME/bin` to the environment variable `PATH`.

3. Define credentials as secret variables

Kiuwan credentials should be added as secret variables. These variables will be applied to environments via the runner.

To add the variables go to Settings -> CI/CD menu and clicking on Expand and add both variables.



4. Perform an analysis

To perform a Kiuwan analysis, we need to install a GitLab runner (local agent) that will be responsible to execute the steps of the pipeline.

To install a gitlab runner, follow these instructions: <https://docs.gitlab.com/runner/install/>

When we create the agent it is necessary to specify the tags of the steps we want to analyze with the runner.

In the previous example, the tags used are **maven** and **kiuwan**:

```
Administrator: Windows PowerShell
PS C:\GitLab-Runner> ./gitlab-runner.exe register
Runtime platform arch=386 os=windows pid=12448 revision=8fa89735 version=13.6.0
Enter the GitLab Instance URL (for example, https://gitlab.com/):
https://gitlab.com/
Enter the registration token:
Enter a description for the runner:
[oci:k8s-40846521]-MyRunner
Enter tags for the runner (comma-separated):
maven,kiuwan
Registering runner: succeeded runner=dev08paf
Enter an executor: docker-windows, parallels, ssh, docker-machine, docker-ssh-machine, kubernetes, custom, docker, docker-ssh, shell, virtualbox:
shell
Runner registered successfully. Feel free to start it, but if it's running already the config should be automatically reloaded!
PS C:\GitLab-Runner>
```

Once this is done, perform an analysis in the pipeline screen to test the configuration.

