Simple CMake integration with Gitlab Pipelines

3 min read · Oct 18, 2021

Yann Cardaillac Follow

Listen Share

Today, I wish to show how to integrate a simple CMake app in gitlab CI/CD, I wish to keep that simple to be used as a side note for later reuse. We will also go through a very simple gitlab-runner set up as a Docker container.

For that I will be using the CMake tutorial that can be found here :

https://cmake.org/cmake/help/latest/guide/tutorial/A%20Basic%20Starting%20Point.html

I've learned quite a bunch of stuff, I've summed that up a little through my commits if you're interested. For instance it's possible to make tests directly out of the box with CMake, it's also possible to generate code with tools directly from the CMake.

What we are doing in the CMake tutorial is a very simple program that basically runs a custom sqrt. The point of the CMake Tutorial was simply to go around several widely used cmake features.

I've been copying the tutorial part to a public gitlab so that I would be able to work on the CMake in a simpler manner:

https://gitlab.com/yccorp/simple-cmake-gitlab-ci

So first if you don't have any gitlab runners yet, either use the free gitlab minutes or register a gitlab runner on your own. You can refer to the documentation that is really great. https://docs.gitlab.com/runner/install/index.html

On my side I've been using gitlab-runner straight out of a docker, the documentation for that is here :

https://docs.gitlab.com/runner/install/docker.html

Basically run gitlab/gitlab-runner:latest and use /srv/gitlab-runner from host as /etc/gitlab-runner on container :

```
docker run -d --name gitlab-runner --restart always \
-v /srv/gitlab-runner/config:/etc/gitlab-runner \
-v /var/run/docker.sock:/var/run/docker.sock \
gitlab/gitlab-runner:latest
```

then register a runner :

```
docker run --rm -it -v /srv/gitlab-runner/config:/etc/gitlab-runner gitlab/gitl
```

Go through the registering process, once it's done you should see your runner available on gitlab.

But hey, there's no chance this docker embeds the dev tools we need. For that there's several solutions we might use. The proper one would be to use a docker image with the dev tools and use gitlab-runner docker with that docker image.

However in my case I don't have any docker registry available easily so I'll stick with the gitlab/gitlab-runner docker and upgrade it with my needed tools in order to use the gitlab-runner as a shell runner. That works but you should probably have your own Dockerfile image for building.

So here's my dummy Dockerfile :

https://gitlab.com/yccorp/simple-cmake-gitlab-ci/-/blob/simple-ci/.docker/Dockerfile :

```
FROM gitlab/gitlab-runner:latest

RUN apt-get update && \
        apt-get install -y --no-install-recommends make cmake g++
```

gistfile1.txt hosted with ❤ by GitHub view raw

Build your docker image and find it with

```
docker image ls
```



```
docker run -d --name cutom-gitlab-runner --restart always \
-v /srv/gitlab-runner/config:/etc/gitlab-runner \
-v /var/run/docker.sock:/var/run/docker.sock \
e39a9647ba9b
```

Now that we have a runner let's quickly see how the .gitlab-ci.yml is working with the beginning of the file:

https://gitlab.com/yccorp/simple-cmake-gitlab-ci/-/blob/simple-ci/.gitlab-ci.yml :

```
stages :
  - build
  - test
  - release
```

.gitlab-ci.yml hosted with ❤ by GitHub view raw

Here's what we're doing, we are defining three stages for our pipeline : build, test and release. The pipeline can not advance to the next stage before the previous one is done, runners can pick any job from the current stage and run them, jobs in stages can happen in parallel in case you have enough workers to run them.

```
build-project:
  stage : build
  script:
    - mkdir Step7_build
    - cd Step7_build
    - cmake ../Step7 && cmake --build . && make
```

.gitlab-ci.yml hosted with ❤ by GitHub view raw

Then we are defining a simple job in the stage build, called build-project. What it does is that it executes all the commands that are listed afterwards.

What you might want to do is to wrap that bunch of shell commands inside an executable script file so that it's easier to maintain.

Then according to the CMake tutorial I've added a two other jobs to do some simple testing and to provide the a release as a .deb artifact.

```
release-project:
  stage: release
  script:
    - mkdir Step7_build
    - cd Step7_build
    - cmake ../Step7 && cmake --build . && make
    - cpack -G DEB
  artifacts:
    paths:
      - ./Step7_build/Tutorial-*-Linux.deb
```

.gitlab-ci.yml hosted with ❤ by GitHub view raw

We are providing the resulting .deb as an artifact that can be downloaded straight out of gitlab.

Of course there's few optimization to be done, for instance it's not necessary to rebuild everything everytime, we might as well provide the Step7_build directory to the other jobs. But the purpose of this article was only to have a quick how to about gitlab pipelines, that is why I allowed myself some shortcuts.

And voilà, here's the result, you can see the simple pipeline build and artifacts here :

https://gitlab.com/yccorp/simple-cmake-gitlab-ci/-/pipelines/386857449

Gitlab Ci Cd Pipeline Cmake Tutorial

Follow

Written by Yann Cardaillac

17 followers · 3 following

I'm an Embedded Linux Engineer based in Nantes (France) and I'll be using this blog to have and give easy access to what I love and learn or want to share.

No responses yet

Write a response

What are your thoughts?

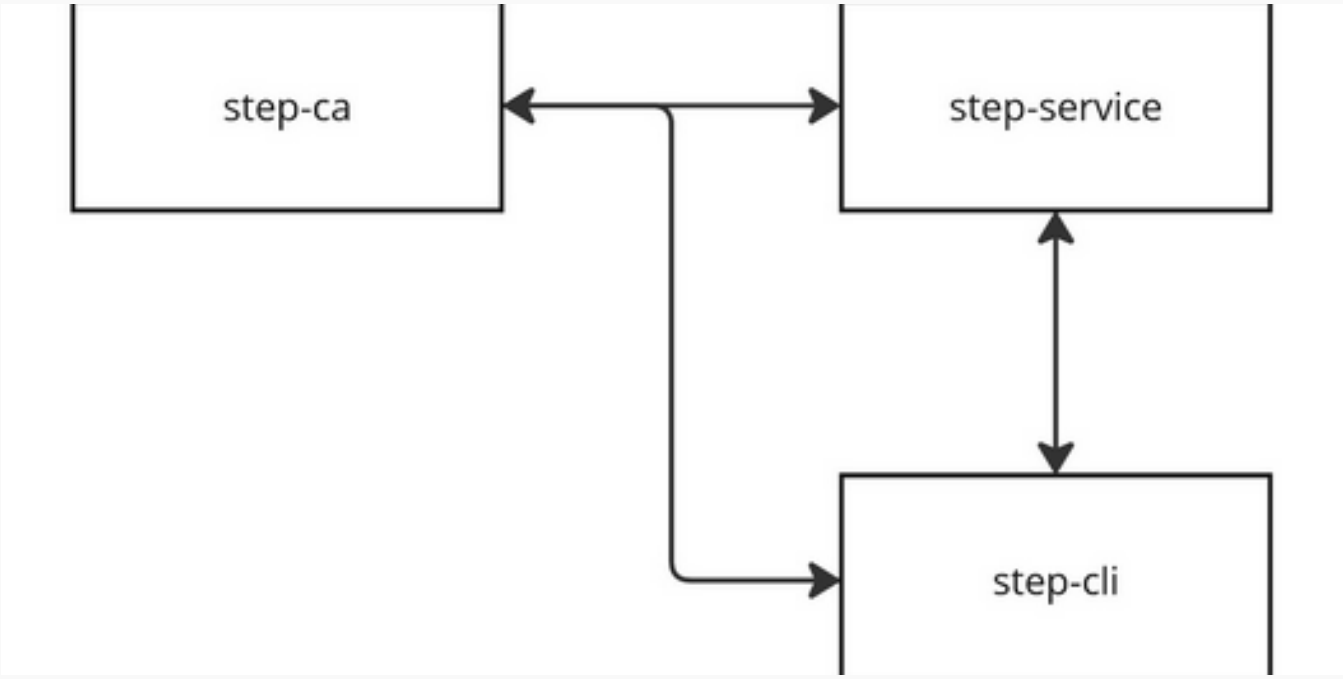
More from Yann Cardaillac

Yann Cardaillac

Ubuntu 22.04 in simple Kiosk mode

Simple ubuntu server installation in an embedded X86 target

Jul 27, 2022 1 1



Yann Cardaillac

Learning PKI with step ca

This post explains the process of setting up a Public Key Infrastructure (PKI) using step-ca. As part of my efforts to enhance IoT fleet...

Feb 7 1 1

Yann Cardaillac

App Versioning using CMake

The title may be somewhat misleading. Naturally, in this project, I utilized Git. The objective was to furnish a C++ application with its...

Mar 29, 2024 2

Yann Cardaillac

Yocto Multiboot image (recovery/system or more)

One of my clients needed to create a custom WIC image with a dual recovery/system partitioning strategy.

Jul 5, 2024 10

See all from Yann Cardaillac

Recommended from Medium

In DevOps.dev by Mohammad Faisal Khatri

How to Install and Set Up Jenkins with Docker Compose

A complete step-by-step guide to install and run Jenkins using Docker Compose with detailed explanation and walk-through.

May 16

Minimal Devops

CI Anywhere with Dagger: Run the Same Pipeline Locally and in GitHub Actions

Why wait for CI to tell you something's broken? With Dagger, your pipelines run before you push.

Jun 19 7

David Regalado

What is Gitflow?

How to get started with it

Jan 20 103 1

In AI & QA Nexus by Suresh Madhusanka Rodrigo

Mastering GitLab CI/CD: Creating Your Ultimate Customized Pipeline — Part 2

In my previous blog post on GitLab CI/CD, we explored the basics by creating a simple pipeline with two stages — Build and Test — using...

Feb 2