

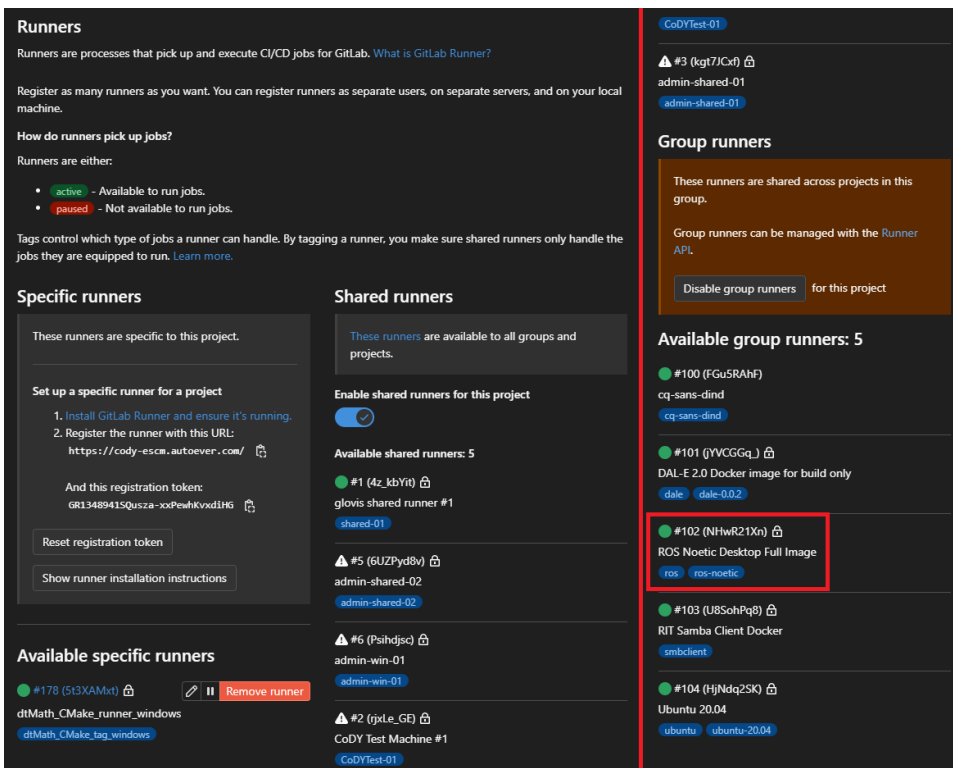
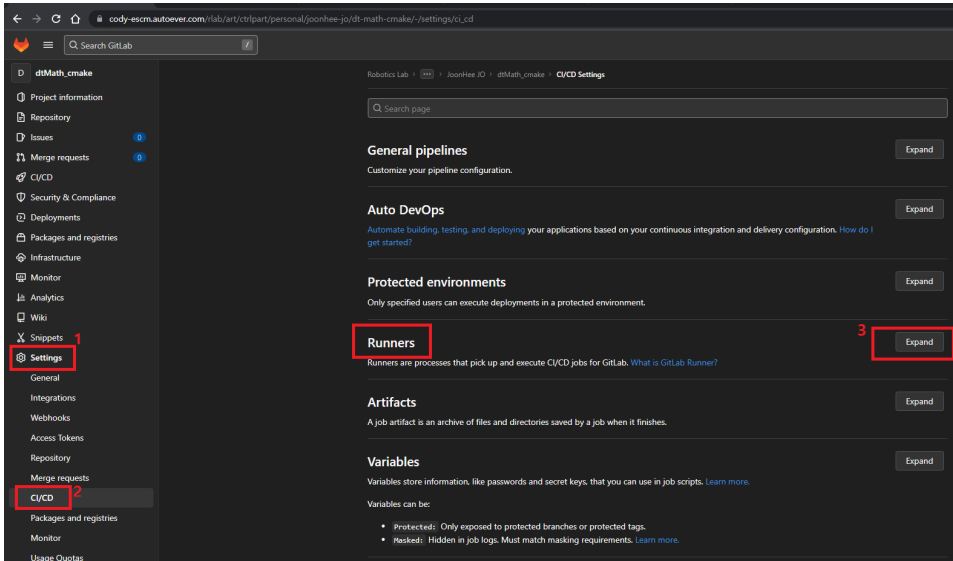
○ GitLab CI/CD를 이용한 컴파일/빌드 자동화

CI/CD를 통해 개발 pc가 아닌 곳에서 컴파일과 빌드가 정상 동작하는지 확인하기 위해 아래의 문서를 작성합니다.

1. Group Runner 이용 (등록과정 생략)

GitLab CI/CD를 이용하기 위해서는 Runner를 등록해야 하는데, Ubuntu의 경우 로보틱스랩에서 등록한 shared runner를 이용하는 것으로 등록 없이 바로 CI/CD 이용이 가능합니다. (개인적인 Runner 등록은 아래의 참조 사이트를 보시면 됩니다.) 따라서, Ubuntu에서 개발자에게 필요한 최소의 컴파일/빌드 자동화 과정만 설명합니다.

아래와 같이, 자동화 하려는 GitLab 프로젝트에서 settings → CI/CD → Runners Expand → Group runners 에 등록된 ros-noetic 을 이용합니다.



2. 프로젝트 폴더에서 '.gitlab-ci.yml' 파일 생성

GitLab은 runner 등록이 되어 있고, .gitlab-ci.yml 파일이 있으면 자동으로 CI/CD를 실행하기 때문에, .gitlab-ci.yml 을 작성하는 것으로 작업을 마무리할 수 있습니다. 여러 기능 없이 GitLab 프로젝트의 여러 commit들 중 tag가 업데이트 될때마다 compile/build를 수행하도록 작성하겠습니다. 예시 코드는 dtMath_cmake 프로젝트에서 확인 가능합니다.

a. 아래 파일 작성후 업로드

b. 원하는 commit에 버전 정보를 표시한 tag 생성 (ex: v1.0.0)

코드에서 '- ./bin/<executable name>' 에 실행 파일 이름 작성후 주석 풀면 실행까지 테스트해볼 수 있습니다. (프로그램이 자동으로 종료되는 경우 해당. while 문 등으로 프로세스를 계속 소유하고 있는 프로그램의 경우 해당 X)

```
# in .gitlab-ci.yml
stages:
  - test
  - build

# Job name 'build'
build_linux:
  rules:
    - if: $CI_COMMIT_TAG =~ /.*/
      when: always
    - when: never
    # - when: always # temporary set until windows build complete (230226)
# Runs at 'build' stage
stage: build
tags:
  # - dtMath_CMake_tag_windows
  - ros-noetic
script:
  - apt update && DEBIAN_FRONTEND=noninteractive apt install -y git-core
  - rm -rf build # Guarantee that tests and coverage are using latest src and tests
  - mkdir build
  - cd build
  - cmake ..
  - cmake --build . --config Release
#   - ./bin/<executable name>

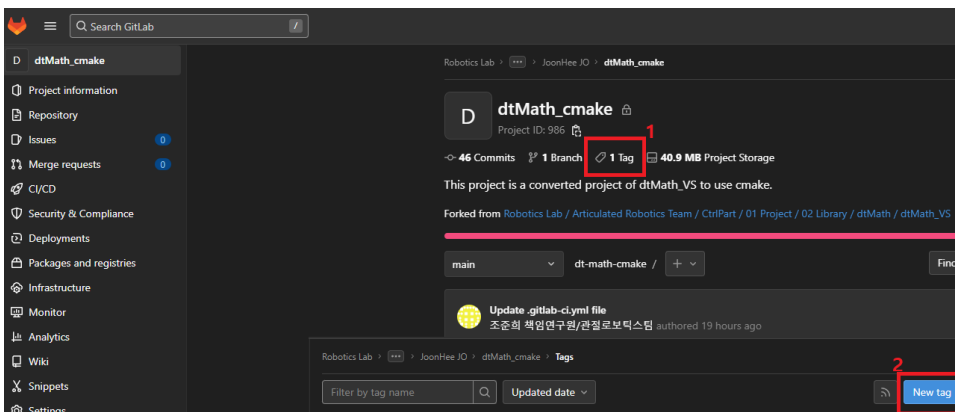
include:
  - template: Security/SAST.gitlab-ci.yml
  - template: Code-Quality.gitlab-ci.yml

code_quality:
  rules:
    - if: $CI_COMMIT_BRANCH == "main"
    - if: $CI_PIPELINE_SOURCE == "merge_request_event"
  services:
  tags:
    - cq-sans-dind

sast:
  stage: test
```

3. Tag 생성

Tag 생성은 업로드 된 commit에 tag를 지정합니다.



Robotics Lab > JoonHee JO > dtMath_cmake > New Tag

New Tag

Do you want to create a release with the new tag? You can do that in the [New release page](#).

Tag name **'v' 포함하여 작성**

v1.0.0

Create from

main

Existing branch name, tag, or commit SHA

Message

version 1.0.0

Optionally, add a message to the tag. Leaving this blank creates a [lightweight tag](#).

Create tag Cancel

Robotics Lab > JoonHee JO > dtMath_cmake > Tags

Filter by tag name

Updated date

New tag

Tags give the ability to mark specific points in history as being important

v1.0.0

7991cc38 · Update .gitlab-ci.yml file · 19 hours ago

version 1.0.0

4. CI/CD 확인

GitLab 프로젝트 → CI/CD → Pipelines → Status 에서 passed 를 통해 동작 완료를 확인할 수 있으며 오류가 있으면 failed가 됩니다. passed를 누르면 pipeline에서 진행된 내용들을 확인할 수 있으며, 세부 내용 선택 시, gitlab-runner가 수행한 상세 작업과정들을 확인할 수 있고, 오류 발생시 에러메시지 또한 확인 가능합니다.

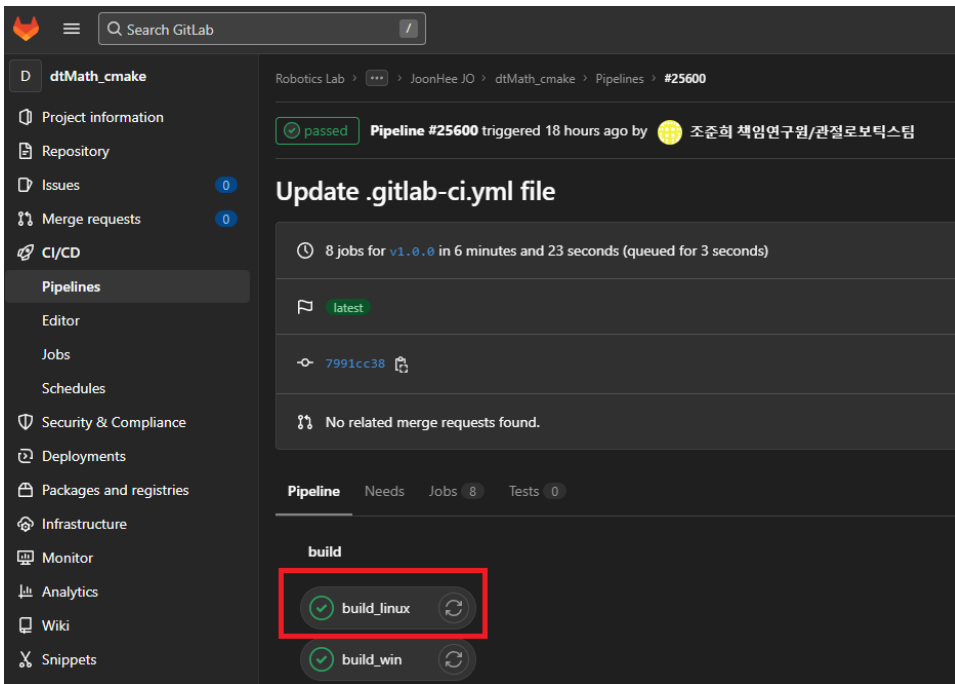
cody-escm.autoover.com / lab/art/ctrlpart/personal/joonhee-jo/dt-math-cmake/-/pipelines

Robotics Lab > JoonHee JO > dtMath_cmake > Pipelines

All 15 Finished Branches Tags

Filter pipelines

Status	Pipeline	Triggerer	Stages
passed	Update .gitlab-ci.yml file #25600 v1.0.0 7991cc38		
passed	Update .gitlab-ci.yml file #25589 main 7991cc38		



The image shows the GitLab CI/CD interface for a project named **dtMath_cmake**. The left sidebar contains navigation links: Project information, Repository, Issues, Merge requests, CI/CD, Pipelines (selected), Editor, Jobs, Schedules, Security & Compliance, Deployments, Packages and registries, Infrastructure, Monitor, Analytics, Wiki, and Snippets.

The main content area displays the pipeline **#25600**, which was triggered 18 hours ago by **조준희 책임연구원/관철로보틱스팀**. The pipeline status is **passed**. Below this, it shows **Update .gitlab-ci.yml file** and **8 jobs for v1.0.0 in 6 minutes and 23 seconds (queued for 3 seconds)**.

The pipeline jobs are listed as **latest** and **7991cc38**. A red box highlights the **build** stage, which includes two jobs: **build_linux** and **build_win**, both marked as successful.

```

1 Running with gitlab-runner 14.2.0 (508e2995)
2   on R05 Noetic Desktop Full Image N9M21Xn
3 Resolving secrets
4
5 Preparing the "docker" executor
6 Using Docker executor with image osrf/ros:noetic-desktop-full ...
7 Using locally found image version due to "if-not-present" pull policy
8 Using docker image sha256:b848f32ea0f9d93bfac7217f2433c191758baad27dcd7a29afe012350bccc for osrf/ros:noetic-de
644bf81b1b65053e76dd10062b7a1e032b75e71929 ...
9 Preparing environment
10
11 Running on runner-ohw2lin-project-986-concurrent-0 via TNG3M03...
12 Getting source from Git repository
13
14 Fetching changes with git depth set to 50...
15 Reinitialized existing Git repository in /builds/r/lab/art/ctrlpart/personal/joonhee-jo/dt-math-cmake/.git/
16 Checking out 7991cc38 as v1.0.0...
17
18 Removing build/
19 Removing extlib/googletest/
20 Removing extlib/opencv/
21 Skipping Git submodules setup
22 Executing "stop_script" stage of the job script
23
24 Using docker image sha256:b848f32ea0f9d93bfac7217f2433c191758baad27dcd7a29afe012350bccc for osrf/ros:noetic-de
644bf81b1b65053e76dd10062b7a1e032b75e71929 ...
25 $ set -e; apt-get update; apt-get install -y git-core
26
27 WARNING: apt does not have a stable CLI interface. Use with caution in scripts.
28 Get:1 http://packages.ros.org/ros/ubuntu focal InRelease [4679 B]
29 Get:2 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
30 Get:3 http://archive.ubuntu.com/ubuntu focal InRelease [265 kB]
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

```

※ 참조 사이트

- GitLab Runner 등록 1
- GitLab Runner 등록 2
- GitLab Runner 등록 3
- GitLab CI 과정 1
- GitLab CI 과정 2
- .gitlab-ci.yml
-