

Using the features of gitlab-CI together with cmake

Florian Goth

SFB 1170, Projekt Z03

30. Januar 2017

Default Branch

master

Tags

Separate tags with commas.

Sharing & Permissions

Project Visibility (?)
Project access must be granted explicitly to each user.

Private

Repository
View and edit files in this project

Merge requests
Submit changes to be merged upstream

Only team members

Builds
Submit, test and deploy your changes before merge

Only team members

Snippets
Share code pastes with others out of Git repository

Disabled

Issues
Lightweight issue tracking system for this project

Only team members

Wiki
Pages for project documentation

Only team members

☒ Allow users to request access
Allow users to request access if visibility is public or internal.

Enabled

LFS
Git Large File Storage


Enabled

☐ Container Registry
Enable Container Registry for this project

Merge Requests

☐ Only allow merge requests to be merged if the build succeeds
Builds need to be configured to enable this feature

☐ Only allow merge requests to be merged if all discussions are resolved



Project Z03

2/9

The builds are executed on runners. Per project runners are possible.

To start serving your builds you can either add specific Runners to your project or use shared Runners

Specific Runners

How to setup a specific Runner for a new project




1. Install a Runner compatible with GitLab CI (checkout the [GitLab Runner section](#) for information on how to install it).
2. Specify the following URL during the Runner setup: <https://git.physik.uni-wuerzburg.de/ci>
3. Use the following registration token during setup: `NYxoznNB7CqVY-n6zz2b`
4. Start the Runner!

Shared Runners

GitLab Shared Runners execute code of different projects on the same Runner unless you configure GitLab Runner Autoscale with MaxBuilds 1 (which it is on GitLab.com).

[Disable shared Runners](#) for this project

Available shared Runners : 5

	244ba550	
	wtpp011	#17
	docker jessie	
<hr/>		
	6ef4e091	
	wtpp010	#11
	docker jessie	
<hr/>		
	0494cf43	
	wtpp009	#10
	docker jessie	

- Configuration is done using `.gitlab-ci.yml`
- The employed language is YAML
- `.gitlab-ci.yml` is under version control.

```
---
```

```
key: value
```

```
map:
```

```
  key1: "foo:bar"
```

```
  key2: value2
```

```
list:
```

```
  - element1
```

```
  - element2
```

```
# This is a comment
```

```
listOfMaps:
```

```
  - key1: value1a
```

```
    key2: value1b
```

```
  - key1: value2a
```

```
    key2: value2b
```

```
---
```

```
types:
```

```
  - build
```

```
  - test
```

```
MyApp:
```

```
  type: build
```

```
  script:
```

```
    - make
```

Integrations

Webhooks can be used for binding events when something is happening within the project.

URL

Secret Token

Use this token to validate received payloads. It will be sent with the request in the X-Gitlab-Token HTTP header.

Trigger

☒ **Push events**

This URL will be triggered by a push to the repository

☐ **Tag push events**

This URL will be triggered when a new tag is pushed to the repository

☐ **Comments**

This URL will be triggered when someone adds a comment

☐ **Issues events**

This URL will be triggered when an issue is created/updated/merged

☐ **Confidential Issues events**

This URL will be triggered when a confidential issue is created/updated/merged

☐ **Merge Request events**

This URL will be triggered when a merge request is created/updated/merged

☐ **Build events**

This URL will be triggered when the build status changes

☐ **Pipeline events**

This URL will be triggered when the pipeline status changes

☐ **Wiki Page events**

This URL will be triggered when a wiki page is created/updated

SSL verification

☒ **Enable SSL verification**







[Add Webhook](#)

Webhooks (0)

No webhooks found, add one in the form above.

Project services

Project services allow you to integrate GitLab with other applications

Service	Description	Last edit
 Asana	Asana - Teamwork without email	
 Assembla	Project Management Software (Source Commits Endpoint)	
 Atlassian Bamboo CI	A continuous integration and build server	
 Bugzilla	Bugzilla issue tracker	
 Buildkite	Continuous integration and deployments	
 Builds emails	Email the builds status to a list of recipients.	4 months ago

zeta

test project

☆ Star

0

F Fork

0

SSH

git@git.physik.uni-wuerzburg.de:fgoth/zeta.g

📄

👤

+

🌐 Global

17 commits

1 branch

0 tags

0.26 MB

Add Changelog

Add License

Add Contribution guide

✖ failed

eb07e76f now with test coverage. · about 21 hours ago by

Florian Goth

Test Program

🔍

PREREQUISITES

Libraries: pthread (optional) Compiler: g++/clang++/msvc/mingw buildsystem: makefile A C++11 conforming compiler is required.

✖ failed

1 second

3 minutes ago

```

gitlab-ci build: runner 3.2.3 (a479607)
Using Shell executor...
Running on wtag007...
Cloning repository...
Cloning into '/home/gitlab-runner/builds/73af4c2f/0/fgoth/zeta'...
Checking out eb07e76f at master
Note: checking out 'eb07e76f' at master.
You are in 'detached HEAD' state. You can look around, make experimental
changes and commit them, and you can discard any commits you make in this
state without impacting any branches by performing another checkout.
If you want to create a new branch to retain commits you create, you may
do so (now or later) by using -b with the checkout command again. Example:

  git checkout -b new_branch_name

HEAD is now at eb07e76f... initial import.
$ make build
$ cd build
$ make pthread
make: *** No rule to make target 'pthread'. Stop.

ERROR: Build failed with status 1

```

Build #79

Retry

Cancel

Run

Duration: 1 second

Created: 3 minutes ago

Finished: 3 minutes ago

Runner: #2

Commit

eb07e76f

Branch: master

Author: Florian Goth

Message: initial import.

- gitlab is stupid. The success of a step is determined by its return value.
- Tests can be integrated using a variety of methods:
 - separate commands
 - using an external framework(CTest, DejaGNU, xUnit variants, Google Test, CppUnit, CXXTest, CUnit)
- test coverage can be forwarded to gitlab via the RegEx string in the configuration.
- gitlab has the additional advantage that it can be integrated with dockerized images.

Prerequisites: A working CTest makefile

types:

- build
- test

MyApp:

type: build

script:

- cmake -E make_directory build
- cd build
- cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Release ..
- cmake --build . --target all --config Release

Testsuite:

type: test

script:

- cmake -E make_directory test
- cd test
- cmake -G "Unix Makefiles" -DCMAKE_BUILD_TYPE=Release ..
- cmake --build . --target all --config Release
- ctest -O log.txt
- cat log.txt | grep "tests passed" | cut -d " " -f 1

Commit `eb07e76fdae47ddb4497a18358c4335a6219257f`

Download as Browse Files Revert Cherry-pick

Authored by **Florian Goth** 8 minutes ago

1 parent `ce26dccc`

`master` ✖ build: failed

now with test coverage.

Changes Builds

3 builds in 31 seconds Retry failed

Status	Build ID	Ref	Stage	Name	Duration	Finished at	Coverage
✓ passed	#111	master	build	MyApp	10 seconds	7 minutes ago	100.0%
✖ failed	#113	master	test	Testsuite	11 seconds	3 minutes ago	33.0%

Retried builds

Status	Build ID	Ref	Stage	Name	Duration	Finished at	Coverage
✖ failed	#112	master	test	Testsuite	10 seconds	7 minutes ago	100.0%