

Using GitLab CI for DUNE modules

Ansgar Burchardt¹

TU Dresden

DUNE User Meeting 2017, Heidelberg



¹Ansgar.Burchardt@tu-dresden.de

Motivation

What are CI systems?

Why GitLab CI?

How does GitLab CI work?

DUNE-specifics

Motivation

Why did I look into CI?

- ▶ Supporting DUNE modules for different DUNE releases and compiler versions
- ▶ Started to implement a CI system for those.
- ▶ Now also used for the DUNE core modules

Why this talk?

- ▶ Some documentation
- ▶ Show that CI is not much work

What are CI systems?

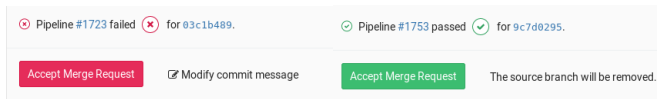
- ▶ Software to support continuous integration
- ▶ Run tests automatically
 - ▶ ... in different environments
 - ▶ ... for proposed changes
- ▶ Why?
 - ▶ More confidence that changes don't break code
 - ▶ “master” branch stays usable
- ▶ How good this works depends on quality of the test suite

Why GitLab CI?

- ▶ Various CI systems:
Jenkins, Buildbot, Travis CI, GitLab CI, ...
- ▶ Why GitLab CI?
 - ▶ Low maintenance
Somebody else already maintains our GitLab server
 - ▶ Integration into GitLab web interface
 - ▶ Easy to provide different environments

How does GitLab CI work?

- ▶ User pushes changes to Git repository
- ▶ GitLab server schedules builds on CI runners
- ▶ Runner runs tests and sends results to GitLab server
- ▶ Status is shown in GitLab (and optionally mail notifications):



- ▶ What to do? Described in `.gitlab-ci.yml`

.gitlab-ci.yml

variables:

DUNECI_BRANCH: releases/2.5

before_script:

- duneci-install-module https://.../dune-common.git
- duneci-install-module https://.../dune-geometry.git
- duneci-install-module https://.../dune-uggrid.git

debian:9--gcc:

image: duneci/base:9

script: duneci-standard-test

debian:9--clang:

image: duneci/base:9

script: duneci-standard-test --opts=/duneci/opts.clang

allow_failure: true

Images? What images?

- ▶ Docker containers for different environments
- ▶ Easy to generate and share container images (Dockerfile)
- ▶ Can be built incrementally:
 - ▶ Distribution base image
 - ▶ Pre-installed build environment
 - ▶ Pre-installed DUNE core modules
 - ▶ Pre-installed DUNE modules
- ▶ All builds happen in a clean environment

Dockerfile: base build environment

```
FROM debian:stretch
[...]
RUN apt-get update && apt-get dist-upgrade [...] \
    && apt-get install --no-install-recommends --yes \
    [... clang, cmake, gcc, git, superlu ...]
    && apt-get clean && rm -rf /var/lib/apt/lists/*
RUN adduser [...] --home /duneci duneci
USER duneci
WORKDIR /duneci
RUN mkdir -p /duneci/bin /duneci/modules
ENV DUNE_CONTROL_PATH=./duneci/modules
ENV PATH=/duneci/bin:$PATH
COPY opts.clang opts.gcc /duneci/
COPY dune-ctest /duneci/bin/
RUN ln -s dune-ctest /duneci/bin/duneci-ctest
COPY duneci-install-module /duneci/bin/
COPY duneci-standard-test /duneci/bin/
```

Dockerfile: pre-install DUNE core modules

```
FROM duneci/base:9
```

```
USER duneci
```

```
RUN duneci-install-module https://.../dune-common.git \  
  && duneci-install-module https://.../dune-geometry.git \  
  && duneci-install-module https://.../dune-uggrid.git \  
  && duneci-install-module https://.../dune-grid.git \  
  && duneci-install-module https://.../dune-istl.git \  
  && duneci-install-module https://.../dune-localfunctions.git \  
  && duneci-install-module https://.../dune-typetree.git \  
  && duneci-install-module https://.../dune-functions.git
```

duneci-standard-test

- ▶ From `.gitlab-ci.yml`:

```
debian:9--gcc:
```

```
  image: duneci/base:9
```

```
  script: duneci-standard-test
```

- ▶ Builds module and tests, and run tests via `dune-ctest`
- ▶ Passes options to `dunecontrol`
- ▶ Just one place to update to, e.g., build in parallel (`make -jN`)

duneci-install-module

- ▶ From `.gitlab-ci.yml`:

```
before_script:
```

- `duneci-install-module https://.../dune-common.git`
- `duneci-install-module https://.../dune-geometry.git`
- `duneci-install-module https://.../dune-uggrid.git`

- ▶ Download and build DUNE modules
- ▶ `--branch [branch]` to choose a different branch
- ▶ or `DUNECI_BRANCH` environment variable:

```
variables:
```

```
  DUNECI_BRANCH: releases/2.5
```

- ▶ Support for feature branches:
when running tests for `dune-A`, branch `feature/X`,
will try to use `feature/X` in dependencies when it exists
(controlled via `DUNECI_BRANCH_AUTO` variable)

Maintenance of the GitLab runner

```
apt install docker.io
```

```
docker pull gitlab/gitlab-runner:latest
```

```
docker stop gitlab-runner
```

```
docker rm -v gitlab-runner
```

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

```
docker exec gitlab-runner gitlab-runner register
```

```
emacs /srv/gitlab-runner/config/config.toml
```

```
# concurrent = ...
```

```
# hostname = "ci" or hostname too long for OpenMPI
```

```
crontab -e # cleanup old Docker containers
```

Outlook

- ▶ Mostly happy with the current state; a few minor bugs.
- ▶ Publish the Docker images used
- ▶ Test reverse dependencies?
Change in dune-geometry also triggers build of dune-grid
- ▶ Move repository from
`https://gitlab.math.tu-dresden.de/ansgar/dune-docker`
to `https://gitlab.dune-project.org`