

Gitlab CI with Docker

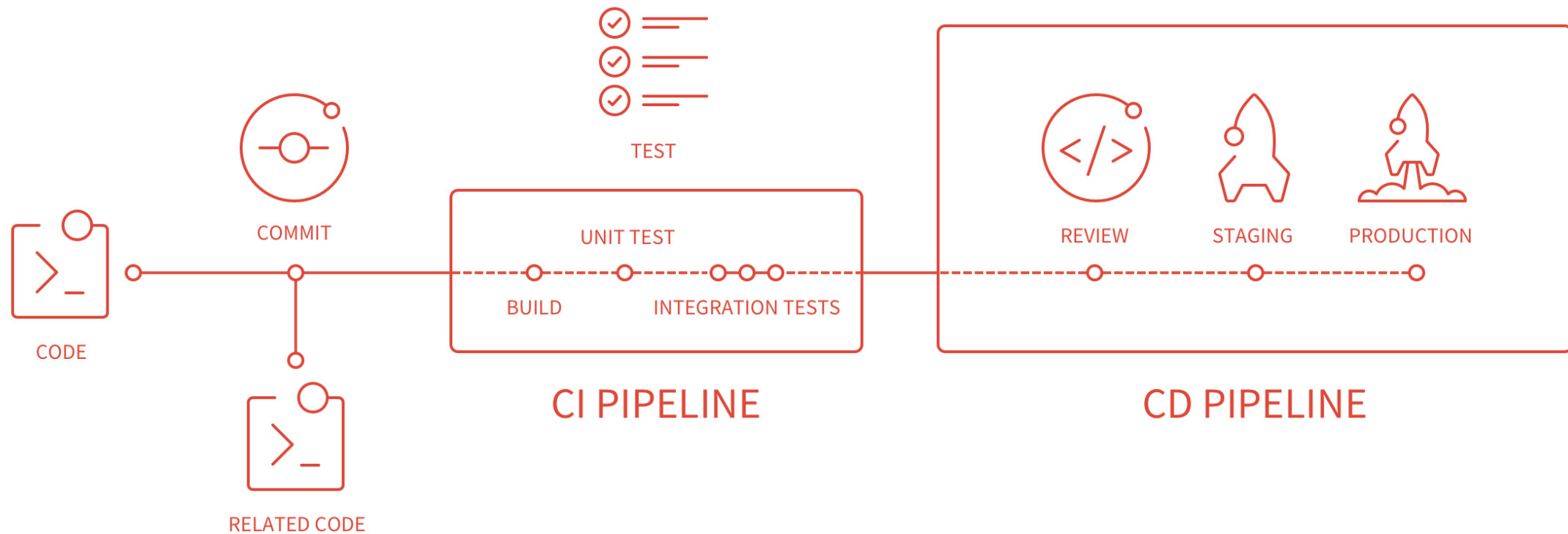


wuxingsheng@iplusbot.com

2017-06-21

Concept

pipeline -> stage -> job



trigger: push merge request http triggers

skip: [skip ci] [ci skip]

2 Steps

- Add `.gitlab-ci.yml`

```
variables:
  GIT_STRATEGY: fetch

build-1404:
  image: ros1404:latest
  script:
    - source /opt/ros/indigo/setup.bash
    - tools/lib_update_posix.sh
    - tools/build_without_lib_update.sh
  only:
    - triggers
    - master
    - zsy_dev
```

- Add `ci runners`

distributed workers, web api

.gitlab-ci.yml

- Global config
 - **stages**
 - image, services, variables, cache
 - before_script, after_script
- Job config
 - **script** (*required*)
 - stage, when, allow_failure, dependencies, artifacts
 - image, services, variables, cache, environment, coverage
 - before_script, after_script
 - **only, except**
 - tags
- Templates (. &) (<< *)

Examples: use of only/except

- only on special branch

```
only:  
  - master  
  - my_dev  
except:  
  - tags
```

- only when new tag pushed

```
only:  
  - tags
```

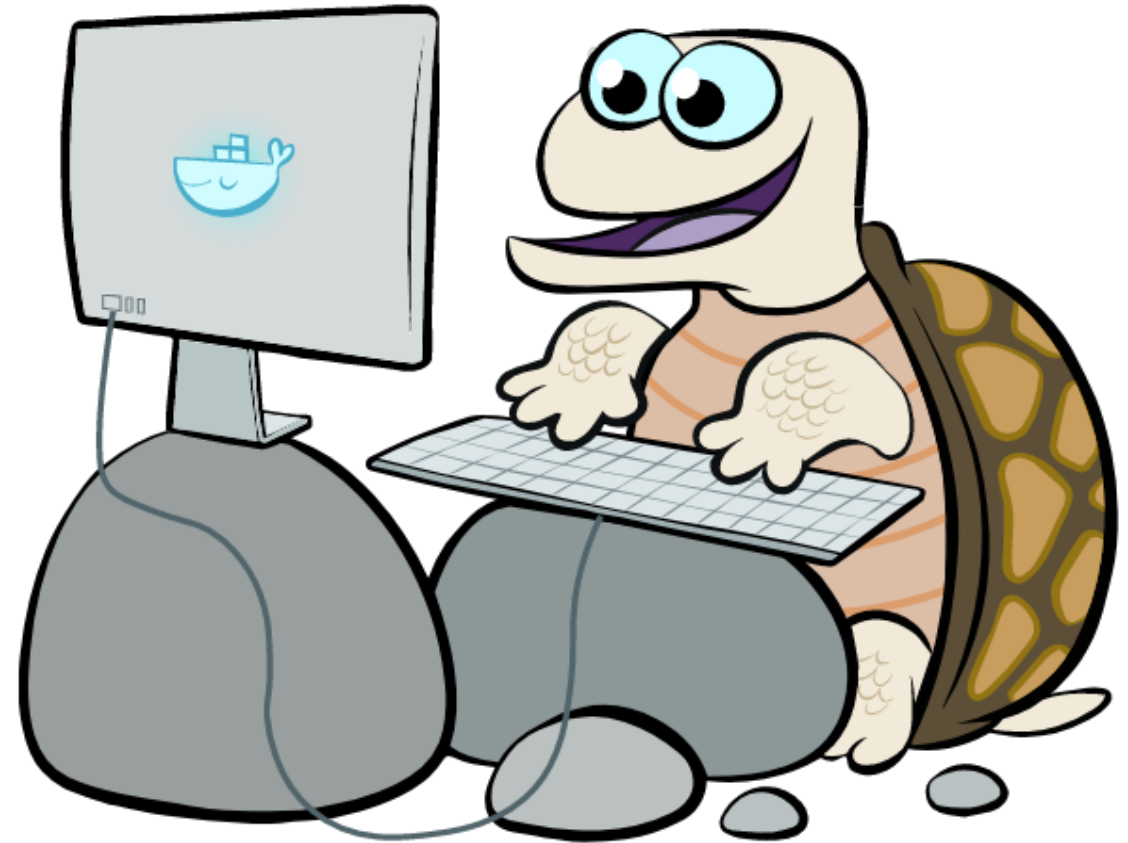
- only can be triggered

```
only:  
  - triggers
```

Runners

prepare env -> pull -> build -> done

- shell
run on local machine
- docker
run in docker, remote or local
- ssh
any server through ssh
- other ...



Add Runner

- install gitlab-ci-multiple-runner

```
$ curl -L https://packages.github.com/install/repositories/runner/gitlab-ci-mult.  
$ sudo apt-get update  
$ sudo apt-get install gitlab-ci-multi-runner
```

- register executors/runners

```
$ sudo gitlab-ci-multi-runner register \  
    --url "http://172.18.118.28/ci" \  
    --registration-token "XXXXXXXXXXXX"
```

tokens can be found at `gitlab -> project settings -> runner`

Example: <http://172.18.118.28/wuxc/ci-test/runners>

Can use runner tags for assign jobs.

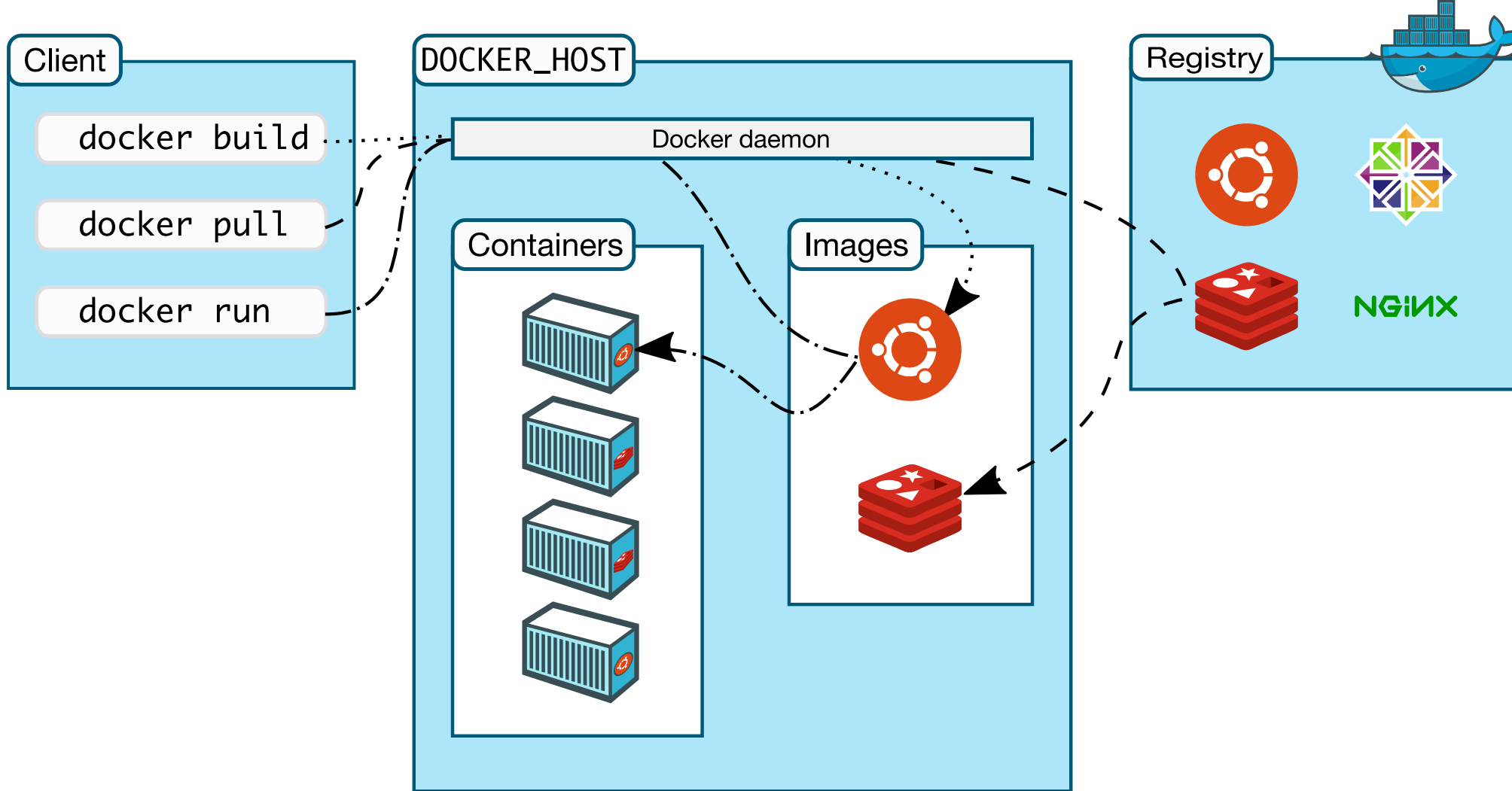
Use Shell Runners -- Example

- register a shell runner
- use job's tags

```
before_script:
  - echo "hello from global before_script"

after_script:
  - echo "bye from global after_script"

test-before-after:
  tags: bash
  before_script:
    - echo "hello from before_script"
  script:
    - echo "2b or not 2b, it's a problem"
  after_script:
    - echo "bye from after_script"
```

Use Docker Runners

- install docker

```
$ sudo apt-get install -y docker.io
```

basic operation: ps images pull push run start rm rmi commit export

- build images

Dockerfile private registry

- register runners

image

Config and Run Runners

- config file

root: `/etc/gitlab-runner/toml`

others: `~/.gitlab-runner/config.toml`

- global config

`concurrent(default 1)` `log_level` `check_interval`

- runner config

- common `output_limit` `url` `name` ...

- ssh runner `host` `port` `user` `password` `identity_file`

- docker runner `pull_policy` `docker args...`

Current Usage & TODOs

- CURRENT 8-projs, simple, only build scripts
- TODO run different job according to changed-files
- TODO add notice after ci
- TODO add trigger from other platform
- TODO add project tests
- TODO collect build result (artifacts)
- TODO deploy with docker

Q&A

Thanks!