

CI / CD 實作工作坊

實作1-1

```
wget https://  
www.dropbox.com/s/  
mqapakoo42z93kt/demo.zip
```

解壓縮後，將檔案複製
到你的 GitLab 專案

實作1-1

git push

實作1-2

ssh 登入 VM

```
ssh vagrant@127.0.0.1 -p 2222
```

實作1-2

```
# 註冊 gitlab-runner  
sudo gitlab-runner register
```

實作2-1

修改 .gitlab-ci.yml ，新增

```
stages:  
  - build
```

實作2-1

修改 .gitlab-ci.yml，新增

build-test:

stage: build

tags:

- "my-runner"

script:

- export CI_COMMIT=\$(echo \$CI_BUILD_REF | cut -b 1-6)

- touch \$CI_PROJECT_NAME-\$CI_BUILD_REF_NAME-

\$CI_COMMIT

- composer install

- npm install

- npm run dev

- rm -rf node_modules

artifacts:

paths:

- "."

實作2-2

修改 .gitlab-ci.yml , 新增

build-release:

stage: build

tags:

- "my-runner"

script:

- export CI_COMMIT=\$(echo \$CI_BUILD_REF | cut -b 1-6)

- export

ARTIFACT_NAME="\$CI_PROJECT_NAME"-"\$CI_BUILD_REF_NAME"-"\$CI_COMMIT".tar.gz

- touch \$CI_PROJECT_NAME-\$CI_BUILD_REF_NAME-\$CI_COMMIT

- composer install --no-dev

- npm install

- npm run production

- rm -rf node_modules

- cd ..

- tar -zc \$CI_PROJECT_NAME/ -f artifacts.tar.gz

- cp artifacts.tar.gz \$CI_PROJECT_NAME/

- mv artifacts.tar.gz /tmp/\$ARTIFACT_NAME

- ansible-playbook \$CI_PROJECT_NAME/ansible/upload_artifacts.yml -e

target_host=localhost -e artifact_name=\$ARTIFACT_NAME

artifacts:

paths:

- "artifacts.tar.gz"

實作3-1

修改 .gitlab-ci.yml ，新增 stage

stages:

- build
- unit-test

實作3-1

修改 .gitlab-ci.yml , 新增

phpunit:

stage: unit-test

tags:

- "my-runner"

script:

- composer run-script post-root-package-install
- composer run-script post-create-project-cmd
- php vendor/bin/phpunit -c phpunit.xml --coverage-text --colors=never

dependencies:

- build-test

實作4-1

修改 .gitlab-ci.yml，新增 stage

stages:

- build
- unit-test
- stg-deploy

實作4-1

修改 .gitlab-ci.yml , 新增

stg-deploy:

stage: stg-deploy

tags:

- "my-runner"

script:

- export CI_COMMIT=\$(echo \$CI_BUILD_REF | cut -b 1-6)

- mv artifacts.tar.gz /tmp/artifacts.tar.gz

- ansible-playbook ansible/auto-deploy.yml -i ansible/

inventory/demo -e target_host=stg -e host_ip=\$stg_host_ip -e

user_pass=\$stg_user_pass -e

CI_PROJECT_NAME=\$CI_PROJECT_NAME

dependencies:

- build-release

environment:

name: staging

url: http://\$stg_domain_name

實作5-1

修改 .gitlab-ci.yml ，新增 stage

stages:

- build
- unit-test
- stg-deploy
- stg-test

實作5-1

修改 .gitlab-ci.yml，新增

test:

stage: stg-test

tags:

- "my-runner"

script:

- curl http://\$stg_domain_name

實作6-1

修改 .gitlab-ci.yml , 新增 stage

stages:

- build
- unit-test
- stg-deploy
- stg-test
- prod-deploy

實作6-1

修改 .gitlab-ci.yml，新增

prod-deploy:

stage: prod-deploy

tags:

- "my-runner"

only:

- master

script:

- export CI_COMMIT=\$(echo \$CI_BUILD_REF | cut -b 1-6)

- mv artifacts.tar.gz /tmp/artifacts.tar.gz

- ansible-playbook ansible/auto-deploy.yml -i ansible/inventory/demo -e

target_host=prod -e host_ip=\$prod_host_ip -e user_pass=\$prod_user_pass

-e CI_PROJECT_NAME=\$CI_PROJECT_NAME

- curl http://\$prod_domain_name

dependencies:

- build-release

environment:

name: production

url: http://\$prod_domain_name

when: manual

實作7-1

修改 .gitlab-ci.yml , 新增 stage

stages:

- build
- unit-test
- stg-deploy
- stg-test
- prod-deploy
- automation

實作7-1

修改 .gitlab-ci.yml , 新增

close-project:

stage: automation

script:

- echo "do something like rm -
rf .* ?"

when: manual

實作8-1

修改 `.gitlab-ci.yml`，在各個 job 新增 `only`。

實作8-1

舉例

```
stg-deploy:  
  stage: stg-deploy  
  only:  
    - master  
  tags:  
    - "my-runner"
```

實作9-1

將 demo 內的檔案複製到 GitHub
專案

實作9-1

新增 .travis-ci.yml

sudo: false

dist: xenial

stages:

- build
- test

jobs:

include:

- stage: build

script:

- composer install
- npm install
- ls

- stage: test

script:

- composer install
- composer run-script post-root-package-install
- composer run-script post-create-project-cmd
- php vendor/bin/phpunit -c phpunit.xml --coverage-text --colors=never