

用 Travis-CI 搭建 Github Pages 网站

目录

- [用 Travis-CI](#)
- [用 Travis-CI 部署](#)
- [用 Travis-CI 搭建 Github](#)
- [用 Web 部署静态网站](#)

2015年2月21日

本文档记录了我使用 Travis-CI 部署静态网站的过程。

用 Github Pages 部署静态网站，使用 [Jekyll](#) 生成静态网站。Github Pages 使用 [Jekyll](#) 生成静态网站，使用 `push` 到 Github Pages 仓库，Github Pages 会自动生成静态网站。

用 [Pelican](#) 生成静态网站，使用 `pelican` 生成静态网站，使用 `python` 生成静态网站。

`Windows` `Linux/OSX/Unix-like` `python` `wordpress` `web` `Android` `SL4A` `python` `pelican` `Android` `git`

Travis-CI

The diagram compares two development models:

- Agile Development:** Shows a long period of development (represented by a long bar) followed by a single commit (represented by a single bar).
- Extreme Programming (XP):** Shows frequent, small commits (represented by many small bars) throughout the development process.

The text "git commit" is shown at the bottom, indicating the action of committing code.

Travis-CI ithub repo
github

Travis-CI

1. 在 Github 上创建 repo
 2. 在 Travis-CI 上添加 repo
 3. 在 Travis-CI 上配置 build script
 4. 在 Travis-CI 上配置 build matrix
 5. 在 Travis-CI 上配置 build cache
 6. 在 Travis-CI 上配置 build notifications
 7. 在 Travis-CI 上配置 build artifacts
 8. 在 Travis-CI 上配置 build security
 9. 在 Travis-CI 上配置 build environment
 10. 在 Travis-CI 上配置 build dependencies



ON

□ Travis-CI □□□□ Github Repo □□□□□

```

    repo: "https://github.com/your-repo"
    language: "python"
    python: "3.7"
    install:
      - pip install -r requirements.txt
    script:
      - python test.py

```

1 language: python

```
1 language: python
2
3 python:
4   - "2.7"
5
6 before_install:
7   - sudo apt-add-repository ppa:chris-lea/node.js -y
8   - sudo apt-get update
9   - sudo apt-get install nodejs ditaa doxygen parallel
10
11 install:
12   - sudo pip install pelican
13   - sudo pip install jinja2
14   - sudo pip install babel
15   - sudo pip install beautifulsoup4
16   - sudo pip install markdown
17   - sudo npm install -g less
18   - wget "http://downloads.sourceforge.net/project/plantuml
/plantuml.jar?r=&ts=1424308684&use_mirror=jaist" -O plantu
ml.jar
19   - sudo mkdir -p /opt/plantuml
20   - sudo cp plantuml.jar /opt/plantuml
21   - echo "#! /bin/sh" > plantuml
22   - echo 'exec java -jar /opt/plantuml/plantuml.jar "$@"' >>
plantuml
23   - sudo install -m 755 -D plantuml /usr/bin/plantuml
24   - wget https://bintray.com/artifact/download/byvoid/openc
c/openc-1.0.2.tar.gz
25   - tar xf openc-1.0.2.tar.gz
26   - cd openc-1.0.2 && make && sudo make install && cd
..
27   - sudo locale-gen zh_CN.UTF-8
28   - sudo locale-gen zh_HK.UTF-8
29   - sudo locale-gen en_US.UTF-8
30   - sudo locale-gen ja_JP.UTF-8
31
32 script:
33   - git clone --depth 1 https://github.com/farseerfc/pelican-
plugins plugins
34   - git clone --depth 1 https://github.com/farseerfc/pelican-
bootstrap3 theme
```

```
bootstrap theme
```

```
35 - mkdir output
```

```
36 - env SITEURL="farseerfc.me" make publish
```

Travis-CI 環境で Ubuntu 12.04 LTS で python をインストールしようとしたら、python 2.7 をインストールする際に pip をインストールする before_install ではなく install が必要で build errored ではなく build fail になる script を修正して build fail を回避する

less.js を ppa を使わずに nodejs でインストールする less を opencv 1.0.2 を Ubuntu で opencv をインストール(0.4)して doxygen を opencv をインストールして pelican をインストールして 4 locale を locale-gen で Ubuntu を Linux でインストールして Ubuntu をインストール

.travis.yml を push して github で travis を clone して travis を build して passing して build を

Travis-CI と Github

travis-ci と Github Pages を使った Github の ssh key を github repo に travis の key をインストールする

Github の Personal Access Token

Applications / **New personal access token**

Token description

travis blog push

What's this token for?

Select scopes

Scopes *limit* access for personal tokens. [Read more about OAuth scopes.](#)

☐ repo ⓘ
 ☐ repo:status ⓘ
 ☐ repo_deployment ⓘ

☒ public_repo ⓘ
 ☐ delete_repo ⓘ
 ☐ user ⓘ

☐ user:email ⓘ
 ☐ user:follow ⓘ
 ☐ admin:org ⓘ

☐ write:org ⓘ
 ☐ read:org ⓘ
 ☐ admin:public_key ⓘ

☐ write:public_key ⓘ
 ☐ read:public_key ⓘ
 ☐ admin:repo_hook ⓘ

☐ write:repo_hook ⓘ
 ☐ read:repo_hook ⓘ
 ☐ admin:org_hook ⓘ

☐ gist ⓘ
 ☐ notifications ⓘ

Generate token

? Personal access tokens function like ordinary OAuth access tokens. They can be used instead of a password for Git over HTTPS, or can be used to [authenticate to the API over Basic Authentication](#).

Github Personal Access Token App Token Travis-CI

Personal Access Token Token public_repo Token

travis

2015221

travis encrypt ruby travis api

repo pubkey

```
1 curl -H "Accept: application/vnd.travis-ci.2+json" https://api.travis-ci.org/repos/<github-id/repo>/key | python2 -m json.tool | grep key | sed 's/.*"key": "(.*)"/\1/' | xargs -0 echo -en | sed 's/ RSA/' > travis.pem
```

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

```
1  echo -n 'GIT_NAME="Jiachen Yang" GIT_EMAIL=farseerf
c@gmail.com GH_TOKEN=<Personal Access Token>' | ope
nssl rsautl -encrypt -pubin -inkey travis.pem | base64 -w0
```

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

```
1  $ gem install travis
```

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

```
1  $ travis encrypt 'GIT_NAME="Jiachen Yang" GIT_EMAIL=farse
erfc@gmail.com GH_TOKEN=<Personal Access Token>'
```

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

```
1  env:
2  - secure: "long secure base64 string"
```

1 echo -n <github-id/repo> | openssl rsa -pubkey 1 -in travis.pem | openssl

```

1  script:
2    - git config --global user.email "$GIT_EMAIL"
3    - git config --global user.name "$GIT_NAME"
4    - git config --global push.default simple
5    - git clone --depth 1 https://github.com/farseerfc/pelican-plugins
6    - git clone --depth 1 https://github.com/farseerfc/pelican-bootstrap3
7    - git clone --depth 1 https://$GH_TOKEN@github.com/farseerfc/farseerfc.github.io
8    - env SITEURL="farseerfc.me" make publish
9
10  after_success:
11    - cd output
12    - git add -A
13    - git commit -m "update from travis"
14    - git push --quiet

```

git push --quiet

 \$GH_TOKEN URL

travis after_success script

 push after_success build

 make github git push make

 github github

Web

repo Travis-CI

 build passing Readme.md

Github Web Github

 commit Travis-CI

