用 Travis CI 自动部署 hexo

Mar 20, 2016

前言

hexo 是当下一个比较流行的静态网站生成器，一般用户的使用方法都是将 hexo 项目的文件放到本地，然后编写文章，最后运行部署的命令将网站部署到一些代码托管网站(如：github)。但是，这样的使用方法有时会带来一些不便。

首先，如果我想在另外一台设备上面写博客，必须将整个项目拷贝过来，完成之后要保持两台设备的内容是一致的，不然就会导致部署之后有不同的地方。

在这种情况下一般会将整个项目都托管到 github 上面。但是这又会导致另外一个问题，每次有改动的时候不但要部署博客，还要提交项目的代码，这又增加了操作的步骤。

之前也有不少文章用不同的方法解决上述的问题，例如利用 Dropbox 同步或者利用 Github 的 Webhooks 进行自动部署。这些方法需要付出一定的成本，因为都需要利用到一台 VPS 去完成。而今有一个更加简单而且免费的方法去完成 hexo 的自动部署，就是利用 Travis CI。

Travis CI

顾名思义，Travis CI 是一个持续集成(Continuous integration，简称CI)的工具。它可以在公共的 Github 仓库上免费使用。

构建

在 Github 建立代码库

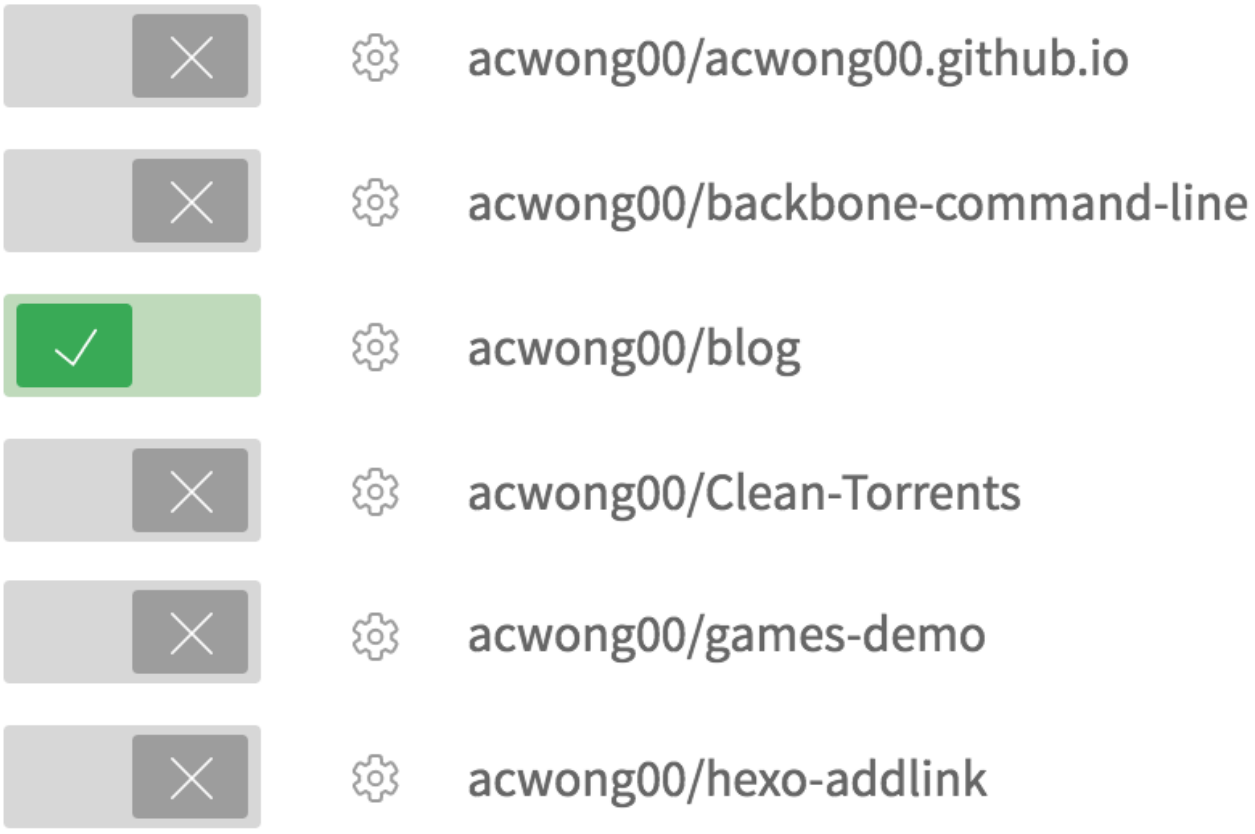
首先，要在 Github 上建立一个代码仓库，要将自己 hexo 博客 push 到上面。hexo 项目作为运行部署的项目，然后 Github Page 的项目作为部署的目标项目。

开启 Travis CI

第二步，我们需要有一个 Travis CI 的账号，直接进入 Travis CI 官网，用自己的 Github 账号授权登录

即可。

然后可以看到当前账号的所有代码仓库，接下来将博客项目的状态设置为启用。



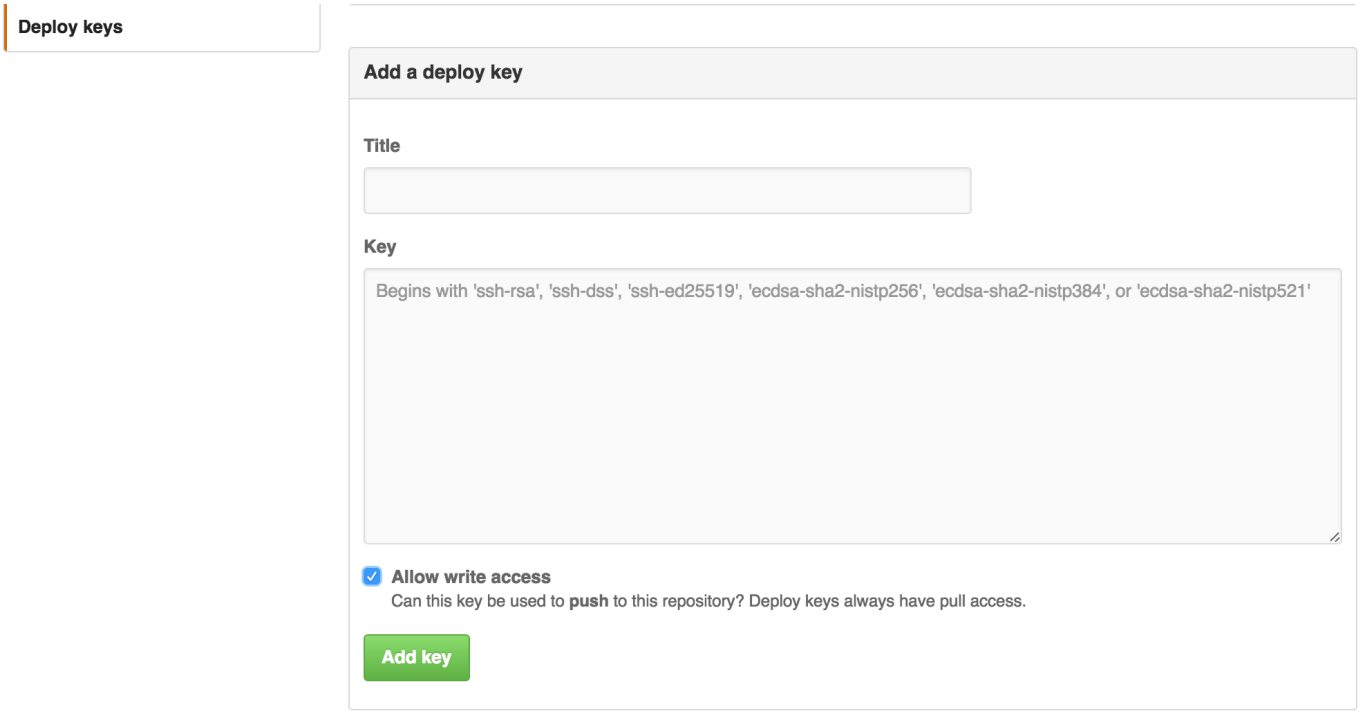
创建 SSH key

第三步，创建一个部署在 Travis CI 上面的 SSH key 利用这个 SSH key 可以让 Travis CI 向我们自己的项目提交代码(也就是将博客部署到 `gh-page`)。

BASH

```
$ ssh-keygen -t rsa -C "youremail@example.com"
```

得到 `id_rsa.pub` 和 `id_rsa` ，然后将有 `pub` 后缀的配置到 `gh-page` 的 Deploy key。



记得要将 `Allow write access` 的选项选上，这样 Travis CI 才能获得 push 代码的权限。

加密私钥

刚才讲公钥文件配置好了，然后就要配置私钥文件，在 hexo 项目下面建立一个 `.travis` 的文件夹来放置需要配置的文件。

首先要安装 travis 命令行工具(如果在国内的网络环境下建议安装之前先换源)。

BASH

```
$ gem install travis
```

用命令行工具登录：

BASH

```
$ travis login --auto
```

然后将刚刚生成的 `id_rsa` 复制到 `.travis` 文件夹，用命令行工具进行加密：

BASH

```
$ travis encrypt-file id_rsa --add
```

这个时候会生成加密之后的秘钥文件 `id_rsa.enc` ，原来的文件 `id_rsa` 就可以删掉了。

这时可以看到终端输出了一段

```
openssl aes-256-cbc -K $encrypted_xxxxxxxxxxx_key -iv $encrypted_xxxxxxxxxxx_iv
```

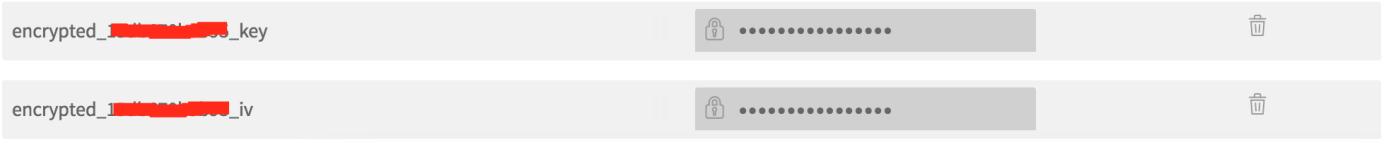
这样格式的信息，这是 travis 用来解密 `id_rsa.enc` 的 key，先保存起来，后面配置 `.travis.yml` 会用到它。

为了让 git 默认连接 SSH 还要创建一个 `ssh_config` 文件。在 `.travis` 文件夹下创建一个 `ssh_config` 文件，输入以下内容：

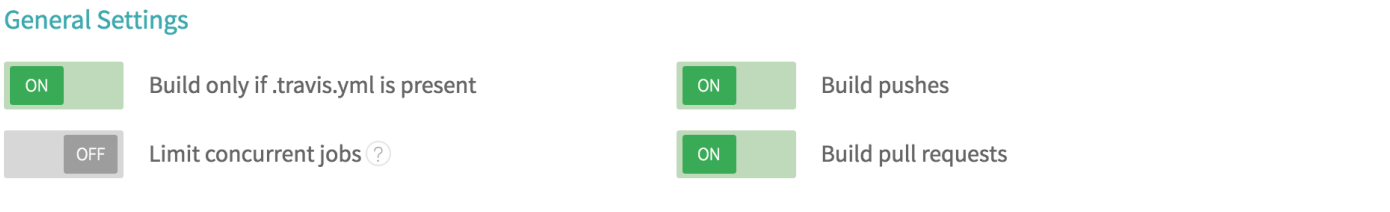
```
Host github.com
    User git
    StrictHostKeyChecking no
    IdentityFile ~/.ssh/id_rsa
    IdentitiesOnly yes
```

现在进入 travis CI 设置页面

可以看到刚刚 travis 命令行生成的解密 key



顺便把上面的开关打开



这样，当向项目 push 代码的时候 travis CI 就会根据 `.travis.yml` 的内容去部署我们的项目了。

.travis.yml

最后就要配置 `.travis.yml` 。在项目的根目录创建 `.travis.yml` 文件。

```
# 配置语言及相应版本
language: node_js

node_js:
```

```
    - "4"


  # 项目所在分支
  branches:
    only:
    - master



  # 配置环境
  before_install:
  # 替换为刚才生成的解密信息
  - openssl aes-256-cbc -K $encrypted_xxxxxxxxxxxx_key -iv $encrypted_xxxxxxxxxxxx_iv -in .tr
  # 改变文件权限
  - chmod 600 ~/.ssh/id_rsa
  # 配置 ssh
  - eval $(ssh-agent)
  - ssh-add ~/.ssh/id_rsa
  - cp .travis/ssh_config ~/.ssh/config
  # 配置 git 替换为自己的信息
  - git config --global user.name 'acwong'
  - git config --global user.email acwong00@gmail.com

  # 安装依赖
  install:
  - npm install hexo-cli -g
  - npm install

  # 部署的命令
  script:
  - npm run deploy  # hexo clean && hexo g -d
```

好了现在只要向项目 push 代码就可以触发部署了，进入https://travis-ci.org就可以看到部署的过程了。

后记

在部署了一遍之后发现，运行 `npm install` 安装 node 的库时候占据了部署的很大一部分时间，这里有一个技巧，可以将 `node_modules` 缓存起来，这样可以节省部署的时间。

```
 # .travis.yml 配置
cache:
  directories:
    - node_modules
```

最后

`.travis.yml` 的完整代码可以看我的 `.travis.yml` 文件。博客的完整代码可以看这里。

感谢您的阅读，有问题欢迎与我交流。

本文地址 http://blog.acwong.org/2016/03/20/auto-deploy-hexo-with-travis-CI/

#hexo #自动部署 #Travis CI

NEXT