



Setup Hugo with Travis CI and GitHub Pages

This article aims to introduce and show how to generate your [Hugo](#) site on [Travis CI](#) and then deploy it automatically to [GitHub Pages](#).

- [Have your Hugo website ready](#)
- [GitHub Pages setup](#)
- [Travis CI setup](#)
 - [Obtain the deploy.sh script](#)
 - [Generate an access token for GitHub](#)
 - [Setup .travis.yml](#)
 - [Cross compiling Hugo](#)
- [Wrapping up](#)

Have your Hugo website ready

The first step is to have your Hugo website working locally at least. It should build without errors. The default output folder is `public`. NOTE: if you changed that using `publishdir` in your config file please change it in the following steps.

For more information, please refer to [Hugo's documentation](#).

GitHub Pages setup

For GitHub Pages, there are a few options available. You can have User/Organization Pages or Project Pages. The steps presented here should work for both but the different branch names might differ depending on how you want to organize your repository.

Usually you have a branch with the website sources and another one with the website generated files. In my case, I have a `sources` branch and a `master` branch because I'm using User Pages (same applies to Organization Pages). For Project Pages, you'd probably have the sources in the `master` branch (in a sub-folder for example) and then the generated files in the `gh-pages` branch (as required by GitHub).

The important thing is to have your sources in the right branch in the first place. For User/Organization pages, it should be any branch other than `master`. For Project Pages, it should be any branch other than `gh-pages`.

There are plenty of tutorials on how to create a new branch in Git and submit it to your GitHub account. For more information, please refer to [User, Organization, and Project Pages](#).

Travis CI setup

On Travis CI, you can build and deploy your website automatically.

Obtain the `deploy.sh` script

You need this script in your root folder :

<https://github.com/X1011/git-directory-deploy/blob/master/deploy.sh>

Generate an access token for GitHub

You need to generate a new “Personal Access Token” from this GitHub page : <https://github.com/settings/tokens>

Click on “Generate new token” and follow the instructions. You need to select the `public_repo` or `repo` scopes. Once the token is generated, you need to copy it.

Then you need to encrypt it so you can use on Travis CI, [follow those instructions](#) and then run this command :

```
travis encrypt GIT_DEPLOY_REPO=https://GENERATED_TOKEN@
```

Replace `GENERATED_TOKEN` with the generated token earlier and `username` by your GitHub username and `reponame` by the repository name.

Setup `.travis.yml`

In your `sources` branch, you have to create a `.travis.yml` file which will be picked up by Travis CI and be used to configure your build

process.

The contents of .travis.yml should be as follows, **note instructions inlined** :

```
env:
  global:
    - secure: "... " # replace by the output from travis
    - GIT_DEPLOY_DIR=public # this is the default output
    - GIT_DEPLOY_BRANCH=master # this is the target branch
    - GIT_DEPLOY_USERNAME="Travis CI" # dummy name
    - GIT_DEPLOY_EMAIL=user@example.com # replace by your email
  branches:
    only:
      - sources # replace by master for Project Pages

install:
  - rm -rf public || exit 0 # cleanup previous run

script:
  - binaries/hugo # generate!

after_success:
  - cp .travis.yml public # all branches need this file
  - bash deploy.sh # run the deploy script
```

NOTE the [latest version of .travis.yml will always be at my sources branch.](#)

Cross compiling Hugo

You could setup your Travis CI to install a Go version, but it will be faster to cross compile the hugo binary yourself and then deploy it in your sources branch.

Run from your root folder those following commands, line by line :

```
mkdir binaries  
cd binaries  
env GOPATH="`pwd`" go get -v github.com/spf13/hugo  
env GOPATH="`pwd`" GOOS=linux GOARCH=amd64 go build -v  
git add hugo
```

This will add the hugo binary compiled for a Travis VM to Git's index. Don't forget to commit it.

Wrapping up

By the end of these steps, you should now have something like this in your root folder :

- [root folder]
 - [archetypes]
 - [binaries]
 - hugo
 - [content]
 - [data]
 - [layouts]
 - [static]
 - [themes]
 - **.travis.yml**
 - config.toml
 - **deploy.sh**

Most of those are Hugo's default folders when you create a new site. The important files are `.travis.yml` and `deploy.sh`.

Once you push everything to GitHub, don't forget to activate your project on Travis CI so it will start builds.