**Travis CI Integration with GitHub**

Benedict Wong

**Introduction:**

Travis-CI is a Continuous-Integration tool that automates testing, deploying and running software projects. Setting up Travis-CI to work with a GitHub repository is easy and quick, allowing developers to make sure that any new changes pass regression testing.

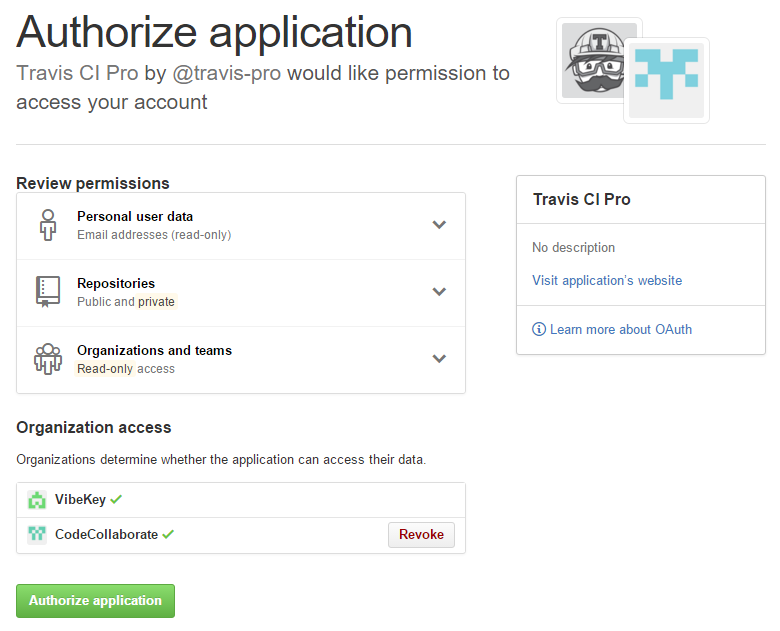
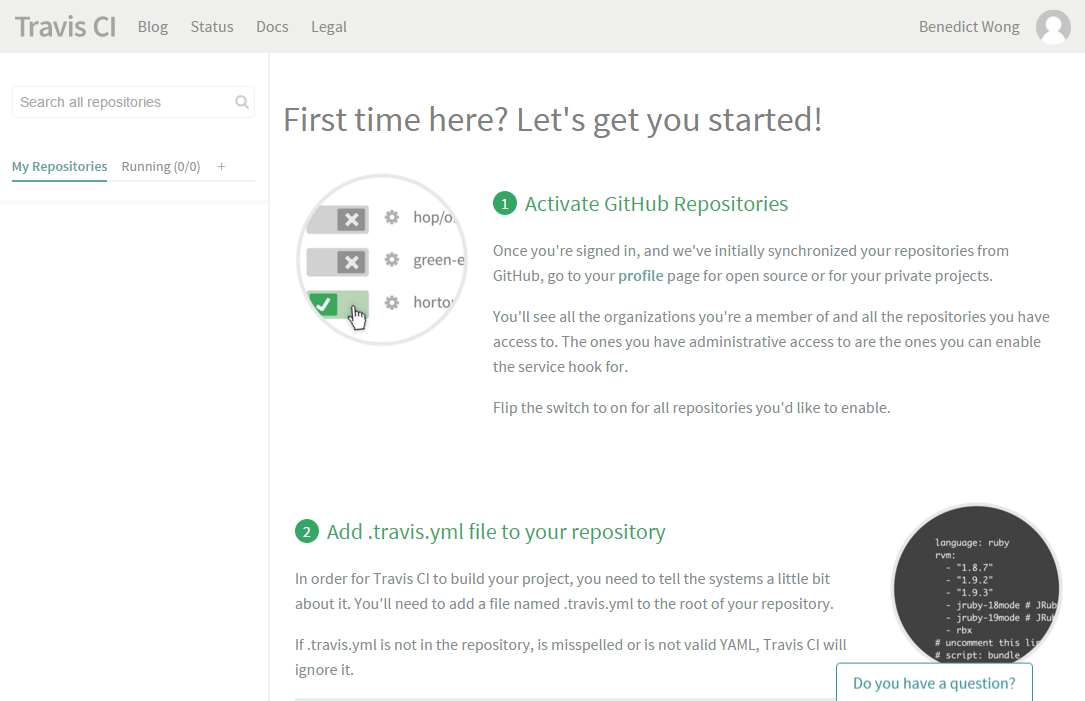
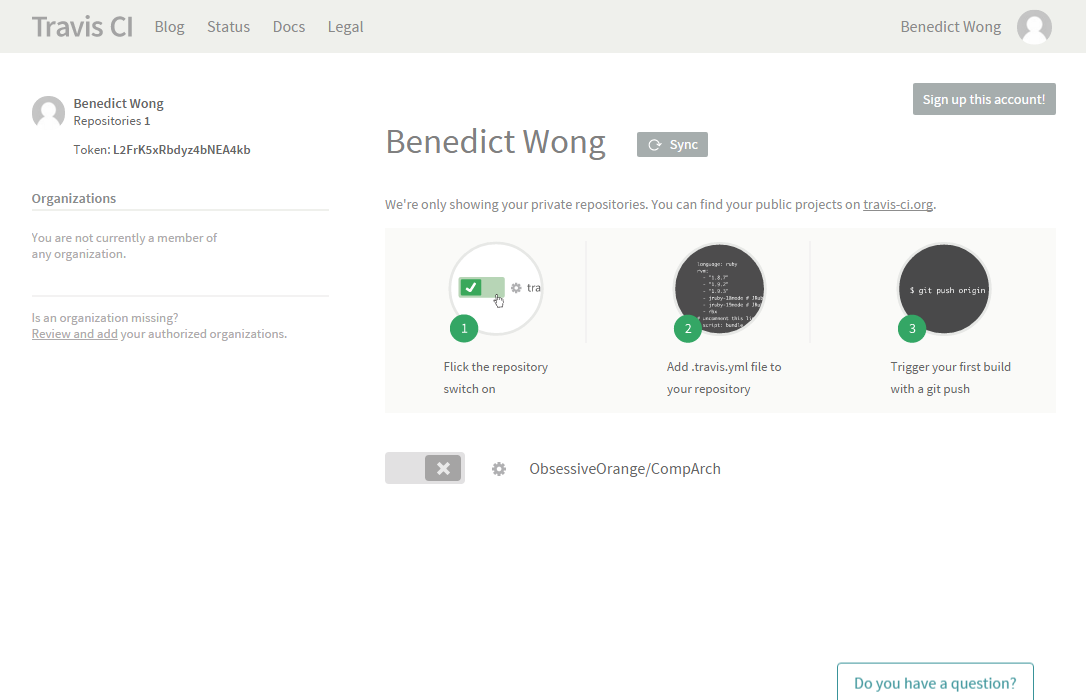
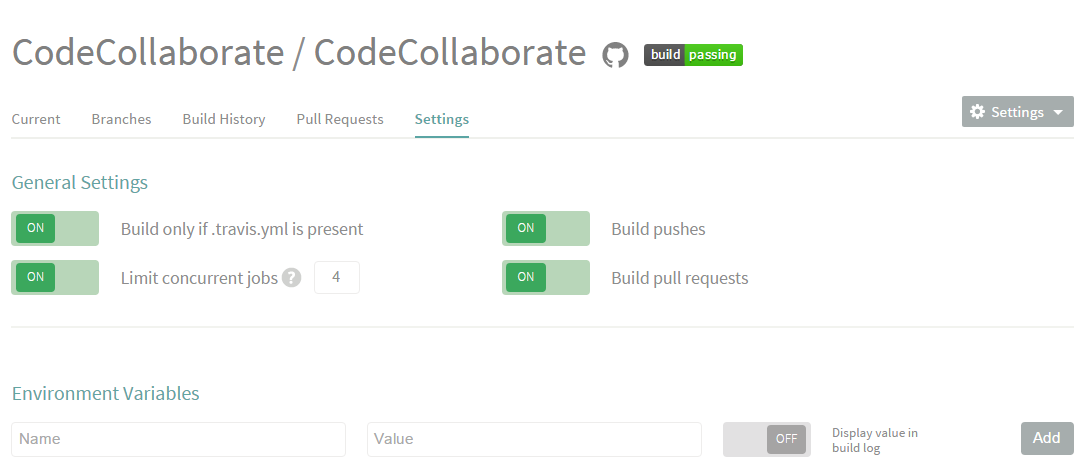
**Goal:**

This document is aimed at providing a setup guide for integrating Travis CI with GitHub, with automated releases on tags that fit into a specific format.

**Requirements:**

* GitHub repository that you have admin access to

**Walkthrough:**

1. Sign in to Travis-CI.org with your GitHub account.
   1. Go to <https://travis-ci.org/>
   2. Click  in the top right
   3. Sign in, allowing permissions as necessary.
      1. Make sure to allow permissions on the repository that you want to integrate Travis-CI on.
      2. 
2. Add your repository
   1. Click on the “+” button near the top of the left (vertical) repository pane.
      1. 
   2. Turn on Travis-CI for your repository
      1. 
   3. Hit the gear button next to the switch to configure the options, setup as follows:
      1. 
   4. Setup any other environment variables as needed.
   5. Setup Travis-CI
      1. See the following sample text from a Travis.yml file

|  |  |
| --- | --- |
| language: go | Specifies the language(s) that will be used in tests/builds |
| go: | Specifies the versions of the languages to test against. Specify multiple versions to test compatibility in each. |
| - 1.5.1 |
| branches: | Only test and release on master, stable and release branches. |
| only: |
| - master |
| - stable |
| - release |
| - /v\d+\.\d+\.\d+.\*/ |
| install: | Pre-installation scripts or shell commands. Use to setup dependencies. In this case, run this external script |
| - "./install-dependencies.sh" |
| script: | Run the tests and builds. In this case, do both separately. |
| - go test CodeCollaborate.go |
| - go build CodeCollaborate.go |
| deploy: | Configure deployment to central release repository. |
| on: | Only build on tagged commits to master, stable and release |
| tags: true |
| all\_branches: true |
| provider: releases | Push this to GitHub releases |
| api\_key: <API KEY> | Use this API Key (Generate from Travis CLI) |
| file: CodeCollaborate | Which file(s) to push |
| skip\_cleanup: true | Do not cleanup files; it needs to be pushed first. |

* 1. Configure the above as needed. The items that may need to be changed are in red. (more information [here](http://docs.travis-ci.com/user/customizing-the-build/))
     1. Get an api\_key in the GitHub Settings page. MAKE SURE IT ONLY HAS public\_repo permission.
     2. Alternatively, follow the instructions [here](http://docs.travis-ci.com/user/deployment/releases/) for an encrypted api\_key

1. You’re done with setup! See https://travis-ci.org/<User>/<Project> for the build status.
2. Tagging a commit as a release.
   1. The configuration above
   2. From Git CLI/Git Bash:
      1. “git tag –a v.1.3.2-rc4 –m “Stability fixes and performance improvements”
      2. “git push –tags”
   3. The Travis server should now build and upload a binary file to GitHub Releases for others to download.