travis-ci Documentation

Release 1.0.0

Pierre Fernique

Jun 16, 2020

Contents

This repository contains scripts designed to be used in `.travis.yml` files of GitHub repositories. For more information considering **Travis CI** refers to its documentation.

Note: It can be convenient to work in a `travis.yml` file instead of `.travis.yml` file. To do so, create the symoblic link `.travis.yml` to the `travis.yml` file.

These scripts are designed to be used with the following `.travis.yml` file:

```
os:
  - linux
  - osx

sudo: required

services:
  - docker

env:
  # Add here environement variables to control the Travis CI build

install:
  - git clone https://github.com/StatisKit/travis-ci.git travis-ci --depth=1
  - cd travis-ci
  - source install.sh

before_script:
  - source before_script.sh

script:
  - source script.sh

after_success:
  - source after_success.sh

after_failure:
  - source after_failure.sh

before_deploy:
  - source before_deploy.sh

deploy:
  skip_cleanup: true
  provider: script
  on:
      all_branches: true
  script: bash deploy_script.sh

after_deploy:
  - source after_deploy.sh

after_script:
  - source after_script.sh
```

Note: The `config.sh` script is executed from within the `install.sh` script.

Travis CI builds are decomposed into jobs. These scripts allow to run different kind of jobs:

CHAPTER 1

Build a **Conda** recipe

To build a **Conda** recipe, you need to use the following environment variables:

- `CONDA_VERSION` equal to `2` (default) or `3`. Control the **Conda** version used for the build.
- `CONDA_RECIPE`. The path to the **Conda** recipe to build. This path must be relative to the repository's root.
- `ANACONDA_LOGIN` (optional). The usename used to connect to the **Anaconda Cloud** in order to upload the **Conda** recipe built.
- `ANACONDA_PASSWORD` (optional). The usename's password used to connect to the **Anaconda Cloud** in order to upload the **Conda** recipe built.
- `ANACONDA_OWNER` (optional). The channel used to upload the **Conda** recipe built. If not given, it is set to the `ANACONDA_LOGIN` value.
- `ANACONDA_DEPLOY` (optional). Deployment into the **Anaconda Cloud**. If set to `true` (default if `ANACONDA_LOGIN` is provided), the **Conda** recipe built will be deployed in the **Anaconda Cloud**. If set to `false` (default if `ANACONDA_LOGIN` is not provided), the **Conda** recipe built will not be deployed in the **Anaconda Cloud**.
- `ANACONDA_LABEL` equal to `main` by default. Label to associate to the **Conda** recipe deployed in the **Anaconda Cloud**.
- `ANACONDA_CHANNELS` (optional). Additional **Conda** channels to consider.
- `TRAVIS_WAIT` (optional). See this page for more information.

Note: It is recommanded to define the environment variables `ANACONDA_LOGIN`, `ANACONDA_PASSWORD` and `ANACONDA_OWNER` in the `Settings` pannel of **Travis CI** instead of in the `.travis.yml` (see this page). This is due to 2 major reasons:

- These variables tends to be shared between various jobs (e.g., all jobs with a `CONDA_RECIPE` environment variable).
- These variables tends to be overriden in forks and **GitHub** pull requests should not modify these values.

- Run a **Jupyter** notebook, you should define these environment variables:

- `JUPYTER_NOTEBOOK`. The path to the **Jupyter** notbook to run. This path must be relative to the repository root.
- `CONDA_ENVIRONMENT`. The path to the **Conda** environment to use when runnning the **Jupyter** notebook.

> **Warning:** Channels given in the `CONDA_ENVIRONMENT` will be overriden by channels added to the **Conda** configuration by the script `config.sh`.

CHAPTER 2

Build a **Docker** context

To build a **Docker** context, you need to use the following environment variables:

- `DOCKER_CONTEXT`. The path to the **Docker** context to build. This path must be relative to the repository root.
- `DOCKER_LOGIN` (optional). The usename used to connect to the **Docker Hub** in order to upload the **Docker** image built.
- `DOCKER_PASSWORD` (optional). The usename's password used to connect to the **Docker Hub** in order to upload the **Docker** image built.
- `DOCKER_OWNER` (optional). The channel used to upload the **Docker** image built. If not given, it is set to the `DOCKER_LOGIN` value.
- `DOCKER_DEPLOY` (optional). Deployment into the **Docker Hub**. If set to `true` (default if `DOCKER_LOGIN` is provided), the **Docker** image built will be deployed in the **Docker Hub**. If set to `false` (default if `DOCKER_LOGIN` is not provided), the **Docker** image built will not be deployed in the **Docker Hub**.
- `TRAVIS_WAIT` (optional). See this page for more information.

Warning: A **Docker** context can only be built on the Linux OS of **Travis CI**.

Note: It is recommanded to define the environment variables `DOCKER_LOGIN`), `DOCKER_PASSWORD` and `DOCKER_OWNER` in the `Settings` pannel of **Travis CI** instead of in the `.travis.yml` (see this page). This is due to 2 major reasons:

- These variables tends to be shared between various jobs (e.g., all jobs with a `DOCKER_CONTEXT` environment variable).
- These variables tends to be overriden in forks and **GitHub** pull requests should not modify these values.

The jobs defined in your `.travis.yml` and the order in which there are runned depend on your repository objective. For example, in the **StatisKit** software suite 3 kins of **GitHub** repositories are considered:

Warning: If a job failed on a given OS, all flowwing jobs on the same OS will fail.