appveyor-ci Documentation

Release 1.0.0

Pierre Fernique

May 07, 2019

Contents

This collection of scripts for **AppVeyor CI** can be used with the following `appveyor.yml` file:

```
platform:
  - x86
  - x64

environment:
  matrix:
    # Add here environement variables to control the AppVeyor CI build

install:
  - git clone https://github.com/StatisKit/appveyor-ci.git appveyor-ci
  - cd appveyor-ci
  - call install.bat

before_build:
  - call before_build.bat

build_script:
  - call build_script.bat

after_build:
  - call after_build.bat

deploy:
  provider: Script
  on:
    branch: master

before_deploy:
  - call before_deploy.bat

deploy_script:
  - call deploy_script.bat

after_deploy:
  - call after_deploy.bat

on_success:
  - call on_success.bat

on_failure:
  - call on_failure.bat

on_finish:
  - call on_finish.bat
```

In the `matrix` section of the `environment` section, you can use the following environement variables to control the **Appveyor CI** build:

- `CONDA_VERSION` equal to `2` (default) or `3`. Control the **Conda** version used for the build.

If you want to:

- Build a **Conda** recipe, you should define these environment variables:
  - `CONDA_RECIPE`. The path to the **Conda** recipe to build. This path must be relative to the repository 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**.

- 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.

Note: It is recommanded to define the environment variables `ANACONDA_LOGIN` (resp. `DOCKER_LOGIN`), `ANACONDA_PASSWORD` (resp. `DOCKER_PASSWORD`) and `ANACONDA_OWNER` (resp. `DOCKER_OWNER`) in the `Settings` pannel of **Travis CI** instead of in the `appveyor.yml`.

More details and illustrations are given in the following guides.

CHAPTER 1

Organization Guide

For organizations, it is recommanded to fork this repository and to adapt the `config.bat` file in which you should give:

- **Conda** channels used for builds and installs,
- **Anaconda** label used for uploads.

For example, let us consider the `config.bat` written for the **StatisKit** organization:

1. The `TEST_LEVEL` environment variables is used in **Conda** recipes to control the test launched (e.g., code:*1* is for unit tests).
2. The `r` **Conda** channels is added for all repositories.
3. Uploads made on the `release` label of the **Anaconda** `statiskit` channel are only allowed for `master` branches. Otherwise, the label is changed to `develop`.
4. `develop` and `release` are the only accepted labels for uploads made on the **Anaconda** `statiskit` channel.
5. For uploads on:
   - Another **Anaconda** channel than `statiskit`, the channels used by **Conda** are `statiskit` (with the `main` label and `develop` labels) and the one given by the code:*ANACONDA_OWNER* environment variable (with the `main` and the label given by the `ANACONDA_LABEL` environment variable if given).
   - The `statiskit` **Anaconda** channel, the channel used by **Conda** is `statiskit` (with the `main` label and the label given by the `ANACONDA_LABEL` environment variable if given).

   Note: In order to prevent **Anaconda** channel collision for the `release` label on the `statiskit` channel (e.g. with **AppVeyor CI**), the `release` label is changed to `appveyor-release`.

CHAPTER 2

Repositories Guide

To activate **AppVeyor CI** for a **GitHub** repository, refers to this page.

Within the **StatisKit** organization, there exits 2 types of deployment for repositories:

- Repositories for realease deployment (e.g., StatisKit). The goal of these repositories is to build all source code that is designed to be installed in the same **Conda** environment and to test them together. To do so,
  - all **Conda** packages are built and deployed to the `release` label (given the environment variable `ANACONDA_LABEL`) without considering the `develop` label.
  - Once all packages are deployed to the `release` label and have been tested, in a last job, packages are moved from the `release` channel to the `main` channel (given by the environment variable `ANACONDA_RELABEL`).

  Warning: These type of repositories must contain `fast_finish: True` in the `matrix` field. Otherwise, the last job moving the packages on the `release` channel to the `main` would be executed even if one job failed.

- Repositories for continuous deployment (e.g., ClangLite).