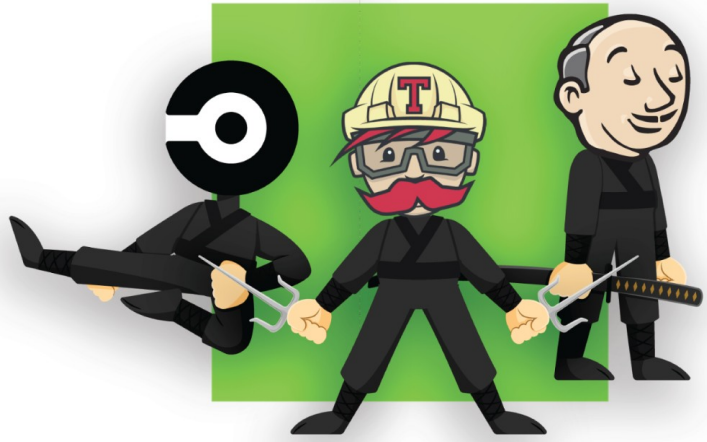
Initial experiences with...

GitHub Actions

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

Sadie Bartholomew
NCAS CMS Group Meeting, 5th June '20



Overview

- Official tagline: "*automate all your software workflows*" i.e. automated building, testing, deployment, etc.
- Often termed "CI/CD" (continuous integration &/or delivery). Other key players, with varying free plans:
  - Circle CI (ninja1) | Travis CI (ninja 2) | Jenkins (ninja 3)
- Comprehensive (but dense) documentation available

Image from site linked to 'key players' text above (on *https://djangostars.com*)

Terminology

- Key terms (demonstrated in practice later):
  - **action**: a set of self-contained tasks that have a specific purpose & can be called upon
  - **workflow**: a full set of automated steps (a "pipeline") configured to run under Actions, one per file (like a Cylc suite – in fact due to be renamed a Cylc workflow for v.8!)
  - **trigger**: some configured event that starts the workflow running
  - **run**: an execution of the steps in a workflow
  - **job**: a separate run of a workflow in a specific environment (e.g. job *A* on Python 3.7 & Mac OS, job *B* on 3.8 & Ubuntu)
  - **matrix**: different setups to run jobs on, potentially in parallel

Basic setup

- Define under `<repo root>/.github/workflows`
- Separate files `<workflow name>.yml` (or `.yaml`) for separate workflows
- Configuration format is YAML:
  - format based around key-value pairs separated by colons
  - see later for examples, or e.g. the YAML homepage, which is itself written as YAML)

Basic workflow templates

- Start with (& keep referring to) official & community 'starter workflows'!
  - official examples from GitHub & user creations on the Marketplace
  - working examples in many languages
  - very useful!

- Often these can be used for your project by adaption i.e. act as pre-configured templates for your workflows

- So take a look through these before writing anything. It is very unlikely you will need to write from scratch!

Workflows created

- I set up workflows on two of our repos, both based on a basic starter workflow for Python packages found here in the official starter-workflows repo
- I set up variations in triggering events to manage resource as (locally) cf-python runs take ~1 hour, cfdm runs only <5 minutes
- To run the test suite for cf-python …
- To run the test suite for cfdm …

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

My experience & tips*

- Be wary of resource limitations, though it may be completely free for public repos (?) e.g. documented here
- Triggering can occur across branches – I found this the most difficult/confusing aspect
- Use a YAML validator before pushing a workflow file
- You can include other (people's) workflows within your own, e.g. the miniconda setup workflow I made use of

National Centre for Atmospheric Science
NATURAL ENVIRONMENT RESEARCH COUNCIL

* with health warning!

In summary…

- GitHub Actions is “CI/CD” so provides automated building, testing, deployment, etc.
- Some competitor systems but this is a built-in part of GitHub & has a good (& probably the best) free package
- YAML-based configuration that I found intuitive & simple
- Starter workflows are available, both official & community-based, & are very useful
- Some aspects are tricky (e.g. triggering from & to branches)