Github Actions Katas

This series of katas will go through the basic steps in github actions, making you able to make CI builds in the end.

Look into the labs folder for exercises.

Building a CI pipeline in GitHub Actions

In this workshop we will be using a small java service which uses Gradle to build the application.

The application is found in the `app` directory, though the details of the implementation are not interesting for the purposes of this katas. There are a number of shell scripts that help with building the application, these are located in the `ci` directory.

The purpose of this katas is to use the small java application to exemplify how to use Github Actions to build, test and package your applications.

We ultimately want a pipeline that has the following jobs:

- **Clone down:** makes the git clone, and prepares the repo for being distributed to the parallel steps
- **Test:** runs the gradle test command found in ci/unit-test-app.sh
- **Build:** runs the gradle build command found in ci/build-app.sh
- **Build docker:** runs both building of the docker image, and pushes it up to the hub
- **Component test:** runs a docker-compose file with a python test to test the application.

Setup

You need to have your own fork of this workshop repository in order for the exercises to work.

Tasks

We are going to create the file that will be used to run the pipeline from, called `.github/workflows/hello-world.yml`.

▶ :bulb: This requires git email and name to bee configured on your machine. If you have not done this, here are the commands to set it up

You need to provide your email and name to git with the following commands.

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

- Fork this repository from the Github website into your own account
- Git clone the project from your own fork down to your VM instance, which is provided to you.
- Leave the browser open and go back to the repository on your computer.

- In your repository on the instance, make a folder called `.github` and inside that one, another folder called `workflows`.

▶ :bulb: terminal commands to do it

- inside that folder, add a file called hello-world.yml (The file path will be `.github/workflows/hello-world.yml` relative to the root of the repository).

▶ :bulb: terminal commands to do it

- Push your changes to github, and navigate to the repository to check that the file has been created.

Now we have set up the fundamentals to run a basic `hello world` build in Github Actions. However, running it would result in an error, as there is nothing in the file.

So in the next section we will be creating the simplest possible workflow.

Table of content

The list here is the natural progression of the labs, but they are made so that they can be worked on independently.