Ingeniería de Software 3
TRABAJO PRÁCTICO N°8

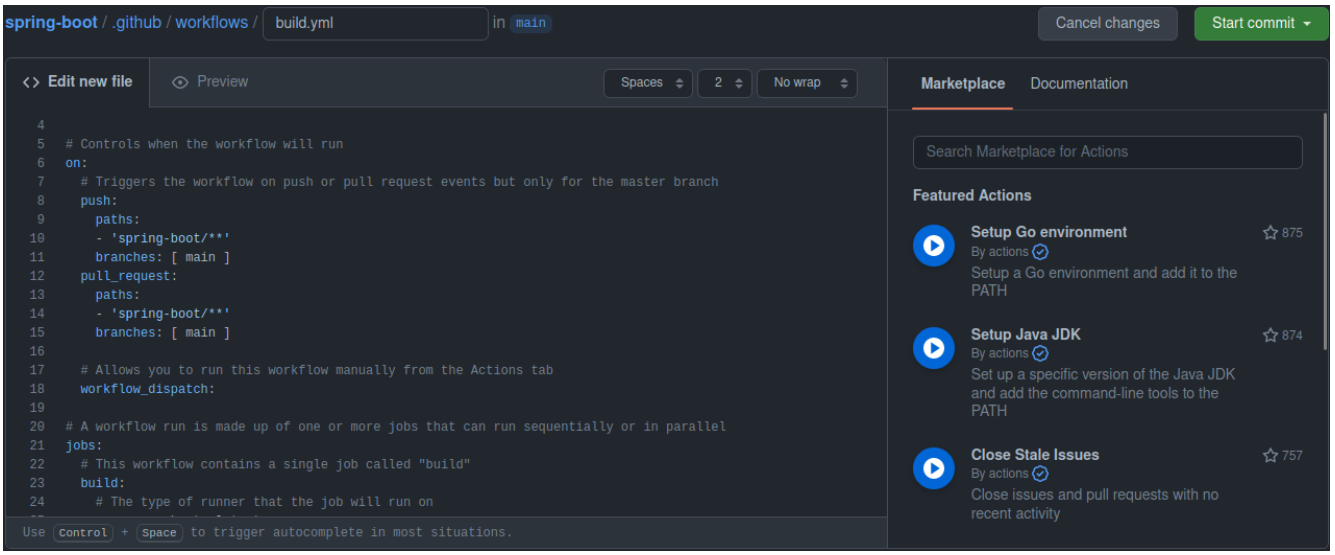
Profesor: Ing. Fernando Bono

Alumno: Menel Angelo (1804789)

EJERCICIO 1:

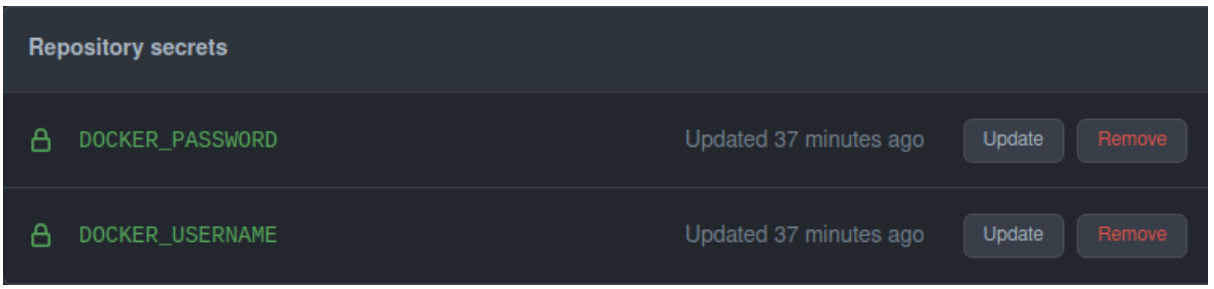
- Github Actions:
  - Pros:
    - GitHub actions are just consecutive docker runs. Very easy to reason about and debug. Reproducing the build environment for container-based Travis is possible, but more difficult. On GitHub actions it's just a docker build docker run away.
    - The individual *actions* in a *workflow* are isolated by default. You can use a completely different computing environment for, say, compilation and testing. Other traditional CI would run all "stages" (~ actions) in the same computing environment. Again, GitHub actions are much easier to reason about and debug.
    - The main.workflow spec (a subset of the HCL and really just a directed acyclic graph) is open source. The whole thing is a pretty thin wrapper around Docker anyway, so platform lock-in is arguably minimal.
    - You have ready access to the GitHub API with (somewhat limited) authentication out of the box.
    - There *might* be a vibrant community (marketplace?) where people can share actions. For example, I'm reusing deploy actions build by different people in different ecosystems.
  - Cons:
    - No native caching. You get *image* and *layer caching* , but nothing else. To build artifacts, you have to roll your own cache (via AWS, Azure, etc. ...), which can be a lot of work.
    - **No support for pull requests from forks**. It's again a bit complicated, and understandable from a security standpoint, but it's currently not possible to run actions a) against the secrets of the receiving repo of a fork PR (base), and/or b) against the *would-be* merge result of a fork PR (that's what travis does). For a workflow that involves forks, that makes GitHub actions largely unusable as CI/CD tools.
    - The quality and breadth of published GitHub actions (at least on the marketplace) is still pretty low / limited.

EJERCICIO 2:



EJERCICIO 3:

- El ejercicio planteado está en Github Action build.yml.
- A diferencia de la referencia que nos dejo en el TP8, borre de environment el **REGISTRY** y de **IMAGE_NAME** borre el ${{ github.repository }} para poder modificar el nombre del username de docker hub.
- En **Settings>Secrets>Actions>New repository secrets** configurar el username y password de Docker Hub.



  - Estos secrets.DOCKER_USERNAME y secrets.DOCKER_PASSWORD los uso para loguearme en docker hub.