

01_introduccion_github_actions

May 16, 2023

1 Introducción a *Github Actions*.

Durante el ciclo de vida del desarrollo de software (*SDL*) es común toparse con procesos que pueden ser complejos, repetitivos, tediosos, o que consumen tiempo en su realización y que son susceptibles de ser automatizados. A estos procesos se le conoce *toil* o esfuerzo innecesario. Estos esfuerzos no generan valor y una vez automatizados, no necesita de una supervisión humana.

Uno de los objetivos de la práctica de *Site Reliability Engineering (SRE)* es la de disminuir este tipo de esfuerzos mediante la automatización.

<https://sre.google/workbook/eliminating-toil/>

Algunas de estas actividades pueden ser:

- Añadir colaboradores.
- Cerrar *issues* y *pull requests* viejos.
- Realizar *pull requests*.
- Generar etiquetas.
- Arreglar *issues* y hacer un *pull requests*.
- Verificar un *pull request* y enviarlo a la rama principal.
- Generar una nueva versión con un número.
- Crear un *pipeline* para probar y construir código.

Github Actions es un servicio de *Github* que permite configurar y ejecutar flujos de trabajo a partir de eventos que ocurren en un repositorio de *Github*. Este servicio permite realizar operaciones de *CI/CD* tanto en el entorno de *Github* como en plataformas de nube e incluso *on-premise*.

Es posible configurar un flujo de trabajo de *GitHub Actions* para que se active cuando ocurra un evento en un repositorio, tal como un *pull request* o la creación de un *issue*. Los flujos de trabajo contienen uno o más trabajos (*jobs*) que pueden ejecutarse en orden secuencial o en paralelo. Cada trabajo se ejecutará dentro de su propio ejecutor (*runner*) de máquina virtual, o dentro de un contenedor, y tiene uno o más pasos que ejecutan un *script* o una acción (*action*).

<https://docs.github.com/en/actions/learn-github-actions/understanding-github-actions>

1.1 Flujos de trabajo (*workflow*).

Los flujos de trabajo (*workflows*) son procesos automatizados y configurables, capaces de correr uno o más trabajos (*jobs*) de forma simultánea y/o paralela.

Los *workflows* se definen mediante un documento *YAML* (con extensión *.yaml*) y pueden ser ejecutadas de forma manual, mediante un *trigger* o mediante la *API REST* de *GitHub*.

Los *workflows* pueden construir, probar empaquetar, liberar y desplegar cualquier proyecto en *GitHub*.

Un repositorio de *GitHub* puede contener uno o más *workflows*, los cuales se localizan en el subdirectorio `.github/workflows/`.

Ejemplo:

- El siguiente código contiene la plantilla de un *workflow* que despliega infraestructura en *GCP* mediante *Terraform* al hacer un *push* en la rama `main`.

```
name: 'Terraform CI'

on:
  push:
    branches:
      - main
  pull_request:

jobs:
  terraform:
    name: 'Terraform'
    runs-on: ubuntu-latest

    # Use the Bash shell regardless whether the GitHub Actions runner is ubuntu-latest, macos-
    defaults:
      run:
        shell: bash

    steps:
      # Checkout the repository to the GitHub Actions runner
      - name: Checkout
        uses: actions/checkout@v2

      # Install the latest version of Terraform CLI and configure the Terraform CLI configuration
      - name: Setup Terraform
        uses: hashicorp/setup-terraform@v1

      # Initialize a new or existing Terraform working directory by creating initial files, load
      - name: Terraform Init
        run: terraform init
        env:
          GOOGLE_CREDENTIALS: ${ secrets.GOOGLE_CREDENTIALS }

      # Checks that all Terraform configuration files adhere to a canonical format
      - name: Terraform Format
        run: terraform fmt -check

      # Generates an execution plan for Terraform
```

```

- name: Terraform Plan
  run: terraform plan
  env:
    GOOGLE_CREDENTIALS: ${ secrets.GOOGLE_CREDENTIALS }

  # On push to main, build or change infrastructure according to Terraform configuration f
  # Note: It is recommended to set up a required "strict" status check in your repository .
- name: Terraform Apply
  if: github.ref == 'refs/heads/main' && github.event_name == 'push'
  run: terraform apply -auto-approve
  env:
    GOOGLE_CREDENTIALS: ${ secrets.GOOGLE_CREDENTIALS }

```

Fuente: <https://github.com/vikramshinde12/terraform-gcp-github-actions/blob/main/.github/workflows/workflow.yaml>

Esta obra está bajo una Licencia Creative Commons Atribución 4.0 Internacional.

© José Luis Chiquete Valdivieso. 2023.