

GITHUB ACTIONS

David Muñoz Río – UO289792

Ben James Coleman – UO284238

Emilio García Fernández -UO287892

Introducción

Las GitHub Actions son una característica proporcionada por GitHub que permite automatizar flujos de trabajo dentro de un repositorio. Los flujos de trabajo pueden incluir tareas como la Integración Continua (CI), el Despliegue Continuo (CD) y una gran variedad de acciones automatizadas, como pruebas, análisis estático, notificaciones de eventos, entre otras. Al ser tan flexibles, permiten a los usuarios personalizar los procesos de desarrollo según sus necesidades específicas.

Configuración de Github Actions

Para configurar una GitHub Action, se debe crear un archivo YAML en el directorio `.github/workflows` dentro del repositorio, que contendrá la definición del flujo de trabajo, incluyendo los eventos que lo activarán y los pasos que serán ejecutados. Se ha de especificar los eventos que desencadenarán el flujo de trabajo, como `push` o `pull_request`, permitiendo así que la acción responda automáticamente a los cambios en el repositorio.

Los pasos pueden incluir tareas como la compilación del código, la ejecución de pruebas, el despliegue de la aplicación, etc. Las acciones pueden ser predefinidas o personalizadas y estas son fragmentos de código reutilizables que permiten la automatización de tareas.

Activación y ejecución de flujos de trabajo

Los flujos de trabajo en GitHub se activan automáticamente en respuesta a eventos como `pull requests`, inserciones de código y creación de etiquetas. También pueden ser ejecutados manualmente a través de la interfaz de usuario de GitHub.

En el caso de una ejecución manual, el evento `workflow_dispatch` permite a los usuarios iniciar manualmente la ejecución de un flujo de trabajo específico sin necesidad de realizar cambios en el repositorio.

¿Dónde se ejecutan los flujos de trabajo?

Al activar un flujo de trabajo, ya sea de forma manual o mediante un evento, se lanza cada tarea del flujo dentro de su propio servidor. Estos servidores son los `runners`. A diferencia de servidores AWS o Azure, GitHub aplica un enfoque `serverless`, es decir, sin servidores. No hace falta especificar datos regionales, el único objetivo es ejecutar las tareas en cualquier servidor disponible.

¿Cuál es la variedad de runners?

Por defecto, GitHub dispone de máquinas virtuales y contenedores de tipo Ubuntu Linux, Microsoft Windows o macOS. Al especificar estos entornos, cada tarea se ejecuta en una

máquina fresca proporcionada de manera rápida por GitHub. En la configuración del flujo de trabajo se especifica el entorno de ejecución deseado, que puede ser una versión en particular de un sistema anteriormente mencionado, el más reciente o un grupo de máquinas, optando por cualquier máquina libre del grupo especificado. Para aquellos proyectos que requieran de más recursos o potencia de computación, existen runners grandes para los entornos predeterminados. Tras elegir la configuración deseada, se crean múltiples instancias escalables del runner grande que se ajustan a los requisitos de cada trabajo. Al activar un flujo sobre este entorno, se suele llamar al grupo de instancias creadas.

¿Y el resto de los sistemas operativos?

Alternativamente, existe la opción de ejecutar un runner personalizado con el hardware, sistema operativo, herramientas software y la configuración deseada, todo de manera local. Estos runners pueden ser máquinas físicas, virtuales, en contenedores o en la nube. Los runners de GitHub, en comparación, son más rápidos, simples y seguros. Si se usa un runner en local, se debe tomar especial cuidado con los flujos de trabajo automáticos en repositorios públicos, ya que existe la posibilidad de que un usuario cree una rama con código malicioso. Y que active un flujo de trabajo mediante un pull-merge, que luego desencadene una serie de cambios maliciosos sobre el runner, que pueden afectar a la máquina host, la red de esta o la inserción de datos peligrosos y no deseados a la máquina. Para remediar esto, se recomienda usar los runners locales en repositorios privados, o tener un sistema de seguridad y roles como contramedida.

Ventajas e inconvenientes

Como todo, hay una serie de aspectos que cabe tener en cuenta al trabajar con GitHub Actions. Estas pueden influir en cómo operamos con ellas y en nuestro punto de vista sobre las mismas.

Ventajas

- **Automatización de tareas.** El uso de GitHub Actions puede ser bastante útil para automatizar tareas. Gracias a esto, podemos ejecutar conjuntos de pruebas, desplegar, generar documentación... automáticamente, bastando con subir el código al repositorio en cuestión.
- **Alta configurabilidad.** Debido a la variedad de opciones que presenta, es fácilmente configurable, permitiendo intercambiar unas piezas con otras para personalizar el conjunto de operaciones que queremos realizar sobre nuestro repositorio.
- **Compatibilidad entre lenguajes.** Por supuesto, estas acciones pueden ejecutarse independientemente del lenguaje en que esté escrito el código. Esto favorece enormemente el desarrollo, ya que el desarrollador no tiene por qué ajustarse a un lenguaje en concreto.

Inconvenientes

- **Dependencia de la plataforma.** Como su propio nombre indica, GitHub ofrece en exclusividad esta herramienta, por lo que será difícil acostumbrarse al cambio a

otro sistema de control de versiones. No obstante, es probable que otro sistema implemente este mismo mecanismo, pero puede que no.

- **Curva de aprendizaje inicial.** Al igual que es difícil dejar de usarlo, también lo es acostumbrarse a él al principio, sobre todo por su alta configurabilidad.
- **Coste computacional.** Si el número de operaciones a ejecutar es muy alto, es probable que tarden en completarse todos, por lo que es necesario ajustar bien el conjunto de acciones a realizar.

Aplicaciones

- **Librerías de utilidades.** Emplear una batería de pruebas exhaustiva puede ser de utilidad si estamos manteniendo y ampliando una librería.
- **Aplicaciones web y móviles.** Un despliegue continuo y eficiente en varios servidores de forma simultánea podría favorecer al desarrollo de un proyecto web.
- **Sitios web estáticos.** Realizando las configuraciones correctas, se puede hacer un despliegue rápido y eficiente de una página web estática sin muchas complicaciones.