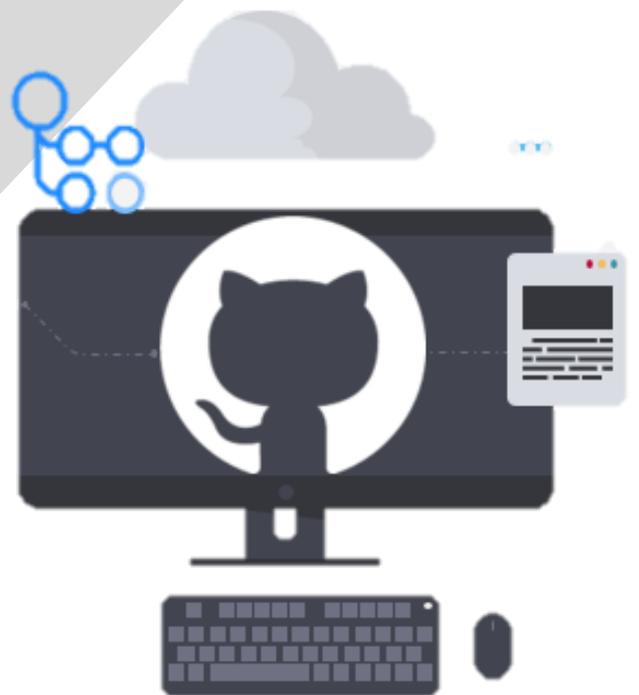


GitHub Actions

Arquitectura del Software



*Ángel Luis Álvarez Iglesias, UO17919
Yeray Rodríguez Granda, UO284247
Coral Izquierdo Muñiz, UO285240*

Introducción

En esta entrada de blog hablaremos de los diferentes temas que se tratan en el podcast [SE Radio 567](#) con Dave Cross desde CI/CD y diferentes herramientas y servicios de automatización como GitHub Actions. Además, nos apoyaremos en la [documentación oficial de GitHub](#) para ampliar información sobre GitHub Actions, workflows y servicios de automatización de dependencias como Dependabot.

Dave Cross

Dave Cross es un ingeniero de software y dueño de la consultora Magnum Solutions Ltd en Londres. En 1988 comienza su carrera profesional como programador y es autor de varios libros como GitHub Actions Essentials, Perl Template Toolkit, The Best of Perl Hacks y Perl Taster. Además, tiene experiencia como programador en empresas como Walt Disney y BBC Technology y actualmente además de ser el dueño y directos de Magnum Solutions Ltd, desde enero de 2024 es ingeniero de software en Tusker.

GitHub Actions

GitHub Actions es una plataforma de integración y despliegue continuo lanzada públicamente en 2019 que permite automatizar un pipeline de compilación, prueba e implementación dentro de GitHub. Permite ejecutar workflows cuando ocurren otros eventos en un repositorio de GitHub, que son procesos automatizados configurables que ejecutan uno o varios trabajos y se definen con un archivo YAML en un repositorio. Son activados manualmente o de forma automática al suceder un evento en el repositorio y se suelen definir en “.github/workflows”. Un ejemplo puede ser desplegar automáticamente el contenido de un repositorio en GitHub Pages cuando se realiza un push a la rama main.

Un fichero YAML es un lenguaje de declaración de datos, que sirve para almacenar archivos de configuración y definen workflows. Se estructuran en un encabezado que proporciona un nombre para el flujo de trabajo y especifica la arquitectura en la que se ejecutará y las definiciones de trabajos que contienen el código a ejecutar. Estos trabajos pueden subdividirse en pasos individuales, y cada paso representa un fragmento de código.

Internamente, GitHub proporciona máquinas virtuales de Linux, Windows y macOS para ejecutar los workflows, pero además permite alojar runners en tu propio centro de datos o arquitectura en la nube. En la entrevista, se debate sobre si usar los contenedores que vienen predefinidos en GitHub o usar contenedores propios y ejecutarlos en la infraestructura de GitHub pagando por ese uso en vez de la forma gratuita que proporciona la plataforma.

CI/CD

Es una práctica de desarrollo de software que combina la Integración Continua (CI) y el Despliegue Continuo (CD) para automatizar los procesos desde la integración de código hasta el despliegue proporcionando un flujo de desarrollo más ágil y eficiente. CI/CD evita errores y fallos en el código sin interrumpir el ciclo de desarrollo y de actualización del software ayudando a disminuir la complejidad, aumentar la eficiencia y optimizar los workflows.

La Integración Continua (CI) implica la automatización de la integración y prueba del código. Cada vez que un desarrollador realiza cambios en el código y lo sube a un repositorio, se realiza una automatización que integra los cambios nuevos y posteriormente se ejecutan pruebas automáticas. El objetivo principal es solucionar los problemas de integración lo antes posible asegurándose de que el código se integre sin conflictos y que el software funcione correctamente.

Un ejemplo de uso sería la ejecución de pasos que incluyan la configuración de un entorno como Node.js, la instalación de dependencias y la ejecución de pruebas.

El Despliegue Continuo (CD) automatiza el proceso de entrega del software a entornos de prueba de forma que después de superar las pruebas tras la integración de código, se despliegue automáticamente la aplicación en un entorno de producción. El objetivo principal es automatizar el despliegue final de una aplicación reduciendo los posibles errores humanos asociados con el despliegue manual y acelerar el tiempo desde el proceso de desarrollo hasta la disponibilidad del producto. Un ejemplo de uso sería el despliegue automático de un entorno de pruebas a un entorno de producción tras pasar correctamente todas las pruebas realizadas.

Event driven architecture

En el podcast se comenta la arquitectura basada en eventos (Event driven architecture) que permite definir workflows que se activan en respuesta a eventos específicos en tu repositorio, como push, pull requests, creación de issues y creación de tags. También se pueden desencadenar flujos de trabajo por eventos temporizados, por activación manual o incluso por solicitudes HTTP externas, de forma que se pueden programar tareas como si fuesen cron jobs. Un cron job es una tarea que se ejecuta de manera automatizada en los sistemas operativos y para la ejecución de este trabajo, el sistema emite órdenes de manera regular. Cada cron job está compuesto por tres componentes: el script que se debe llevar a cabo, el comando que el script ejecuta y la acción o tarea del script. Básicamente los cron jobs sirven para automatizar tareas, que se ejecutarán cada cierto tiempo y se pueden programar estas tareas para que ejecuten lo que se desee.

Usos y ejemplos

Definiendo un fichero YAML, podemos definir Dependabot con un archivo dependabot.yaml en .github, un servicio de automatización de GitHub que ayuda a mantener las dependencias de un proyecto actualizadas. Analiza los archivos de configuración de dependencias para identificar nuevas versiones disponibles de las bibliotecas utilizadas. Además, podemos utilizar GitHub Pages para publicar sitios webs sin necesidad de configurar un servidor web propio. En el podcast, Dave Cross comenta tres proyectos diferentes usando GitHub Actions con Git-scraping, CPAN y static websites.

El "git-scraping" es una práctica innovadora popularizada por Simon Willison que puede utilizarse para recopilar datos de interés de la web mediante el empleo de GitHub Actions. Esta técnica utiliza la funcionalidad de cron job de GitHub para automatizar la extracción de datos de archivos JSON en un flujo de trabajo específico. Además, aprovecha las capacidades de Git para comparar las diferencias entre la versión actual de los datos extraídos con las anteriores, lo que permite mantener un historial de cambios. Un ejemplo de aplicación de esta técnica es la monitorización de incendios forestales en California. Willison recopila datos relevantes sobre incendios forestales de fuentes abiertas y los almacena en una base de datos utilizando Git-scraping. Posteriormente, utiliza Datasette, una herramienta de visualización de datos, para crear sitios web que permiten explorar y analizar estos datos de diversas maneras, incluyendo gráficos que muestran la evolución temporal de los incendios.

Otro caso interesante es una aplicación web desarrollada por Dave Cross llamada "CPAN dashboard". Esta plataforma está diseñada para que los desarrolladores de módulos CPAN (Comprehensive Perl Archive Network) puedan realizar pull requests para agregar sus propios módulos. "CPAN dashboard" emplea GitHub Actions para ejecutar software que recopila información sobre todos los módulos CPAN de un autor utilizando la API de meta CPAN. Luego, esta información se utiliza para generar visualizaciones que muestran cómo están funcionando los módulos en diferentes servicios de integración continua (CI). Esta herramienta proporciona una visión clara y actualizada del ecosistema de CPAN y facilita la colaboración entre desarrolladores.

Además de estos proyectos, también se puede utilizar GitHub Actions para la creación de sitios web estáticos. Por ejemplo, se pueden configurar flujos de trabajo en GitHub que obtengan feeds RSS de varios sitios web, los combinen en uno nuevo y los publiquen automáticamente. Este proceso se puede automatizar para que se repita cada cierto tiempo, garantizando que el sitio web esté siempre actualizado con las últimas noticias o contenido de interés. GitHub Actions permite además activar el

flujo de trabajo cuando se modifica el archivo de configuración que lista todos los feeds que se están agregando. La página web generada de este modo, se publica en un sitio web de GitHub Pages.