



GitHub Composite Actions

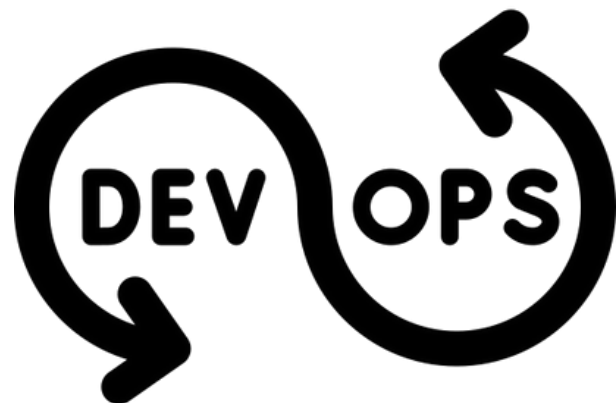


Agenda

- Introducción a Github Actions
- GitHub Actions y CI/CD
- Que es un Composite Actions
- Estructura de una Composite Action



GitHub Actions



En este repositorio puedes encontrar los recursos necesarios para realizar las sesiones practicas.

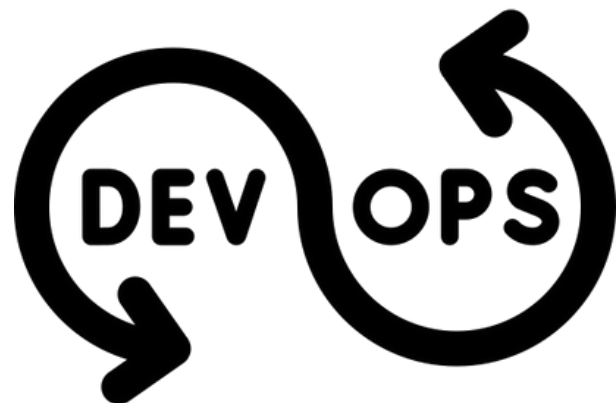
<https://github.com/BCP-training/BCP-training>

Introducción

- GitHub Actions es una herramienta de automatización de flujos de trabajo que se integra directamente en la plataforma de GitHub, permitiendo a los desarrolladores crear, gestionar y ejecutar tareas automatizadas en respuesta a eventos en sus repositorios



GitHub Actions



¿Que son las GitHub Actions?



- **Componentes reutilizables**
- **Impulsado por eventos**
- **Entradas y Salidas**
- **Control de Versiones**
- **Integracion en el Marketplace**

Ventajas de Uso de GitHub Actions

GitHub Actions y las acciones personalizadas proporcionan una potente herramienta para optimizar el flujo de trabajo de desarrollo, mejorar la calidad del software y fomentar la colaboración en equipo.

- **Integración Nativa**
- **Flexibilidad**
- **Escalabilidad**
- **Eficiencia en CI/CD**
- **Mejora de la calidad del código**

- **Colaboración Mejorada**
- **Gestión de Secretos**
- **Personalización**
- **Reducción de Errores**

GitHub Actions: Fundamentos

- Elementos clave
- Workflows, Jobs & Steps
- Eventos (Workflow Trigger)
- Creando un workflow de ejemplo

Elementos Clave

Workflows

- Agrega a un repositorio
- Contiene uno o mas **Jobs**
- Es lanzado por eventos
- Los workflows se define en la carpeta
.github/workflows/<nombre>.yml

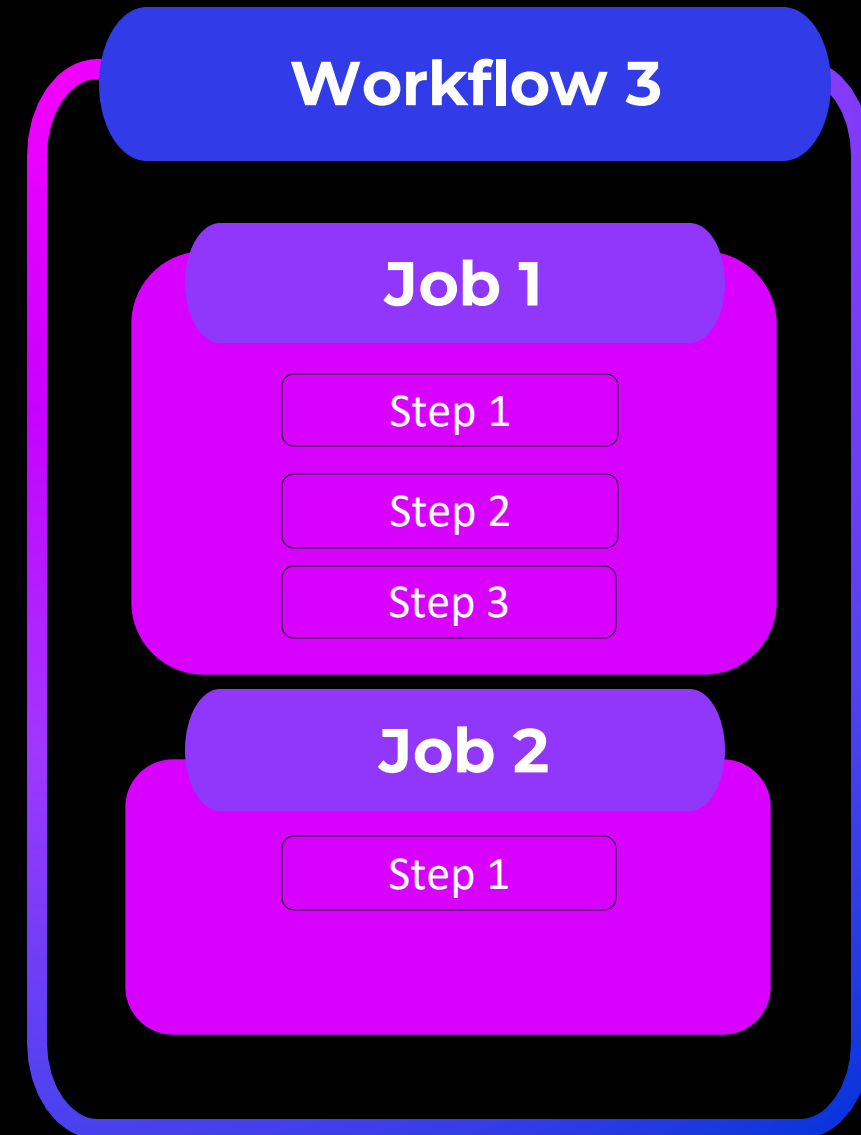
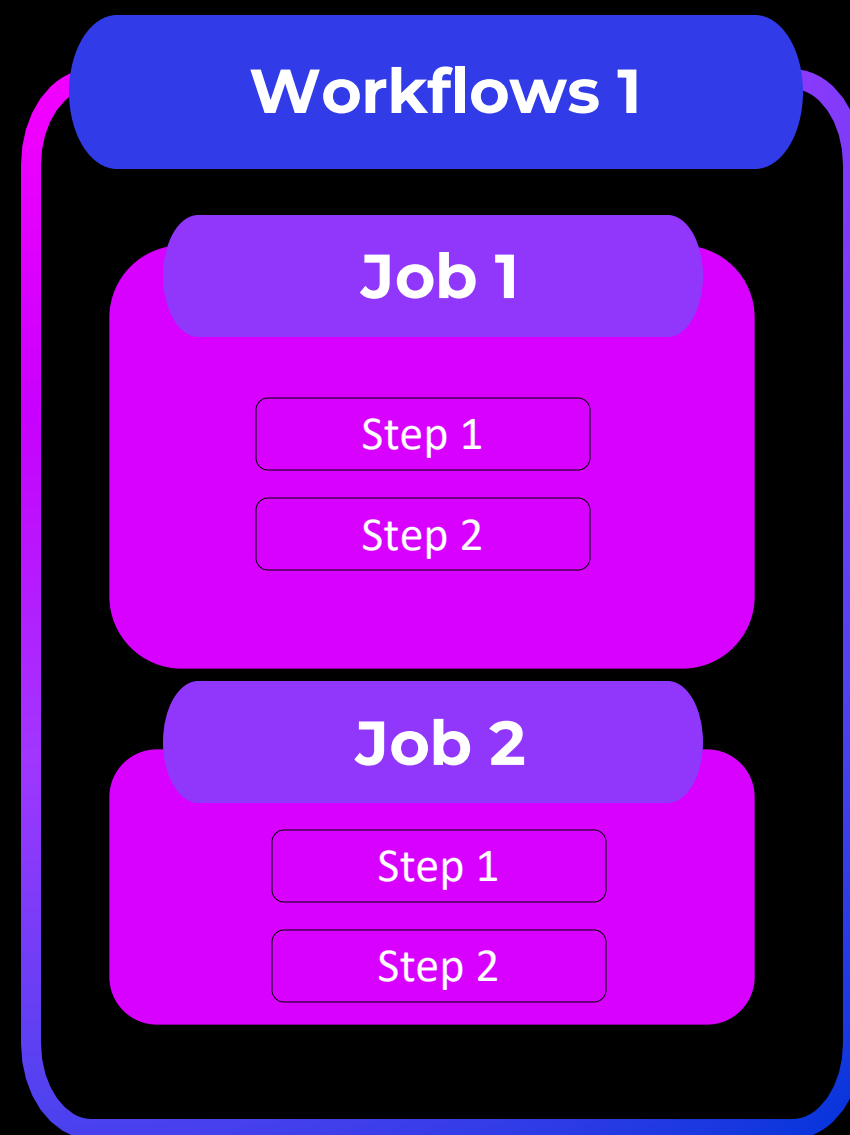
Jobs

- Se define un Runner
- Contiene uno o más **Steps**
- Corre en paralelo o secuencial
- Puede tener un condicional

Steps

- Ejecuta un Shell script o un Action
- Se puede usar un custom o actions de terceros
- Los Steps se ejecutan en orden
- Puede tener un condicional

Workflow, Jobs & Steps



Eventos (Workflow Triggers)

Relacionado con el repositorio

push

create

issues

pull_request

fork

issues_comment

watch

otros...

Workflow

workflow_dispatch

repository_dispatch

scheduler

workflow_call

Questions?





¿Que es una Composite Action?



Concepto de GitHub Composite Actions



GitHub Actions

- Los GitHub Composite Actions son un tipo de acción dentro de GitHub Actions que permiten agrupar varios pasos de ejecución en un solo archivo. Esto facilita la creación de acciones más complejas y organizadas, que pueden ser reutilizadas en diferentes flujos de trabajo.
- A diferencia de las acciones individuales, que generalmente realizan una tarea específica, los Composite Actions permiten combinar múltiples comandos o scripts en un solo paquete, lo que simplifica la lógica de los flujos de trabajo y mejora la mantenibilidad.

Definición y Casos de Uso

Las GitHub Composite Actions son una herramienta eficaz para crear flujos de trabajo más estructurados y eficientes, facilitando la automatización de tareas complejas en el desarrollo de software.

Despliegue de aplicaciones

Puedes crear una composite action para encapsular todos los pasos necesarios para desplegar una aplicación

Configuración de entornos de prueba

Una composite action puede configurar automáticamente un entorno de prueba. Esto puede incluir la instalación de dependencias, la configuración de bases de datos y la inicialización de servicios necesarios.

Automatización de tareas de mantenimiento

Puedes crear una composite action para llevar a cabo tareas de mantenimiento regular, como la limpieza de repositorios, la actualización de dependencias y la generación de informes.

Integración herramientas de análisis

Una composite action puede combinar varios pasos para integrar herramientas de análisis de código y generación de informes de calidad. Esto permite ejecutar análisis de estática y generar reportes de cobertura.

Composite Actions vs Actions Individuales



Las composite actions son útiles para tareas más complejas y estructuradas que involucren varios pasos, mientras que las action individual son ideales para tareas simples y específicas

- Agrupa múltiples pasos (steps) en un solo archivo YAML. Permite definir varios comandos y procesos relacionados dentro de una única acción, facilitando su organización.
- Más fácil de mantener y reutilizar, ya que encapsula lógica compleja en una sola unidad. Ideal para procesos que requieren varios pasos, como despliegues o configuraciones.
- Puede aceptar múltiples entradas y devolver salidas, lo que permite interacciones más complejas entre pasos.
- Ideal para tareas más complejas que requieren múltiples etapas y combinaciones de operaciones.

- Normalmente se enfoca en realizar una tarea específica. Cada acción se define en su propio archivo y generalmente ejecuta un solo comando o función.
- Puede requerir más esfuerzo para mantener si se utiliza en múltiples flujos de trabajo, ya que cada acción se configura por separado. Mejor para tareas simples y específicas que no requieren múltiples pasos.
- Generalmente se limita a recibir entradas y devolver salidas de una única operación
- Más adecuada para tareas simples y directas, donde una única operación es suficiente.

Ventajas



Las composite actions mejoran la organización, la reutilización y la claridad en los flujos de trabajo de GitHub Actions, lo que resulta en una mayor eficiencia y calidad en el desarrollo de software.

Organización y claridad

- Agrupación de pasos
- Mantenimiento simplificado

Reutilización

- Uso en múltiples flujos de trabajo
- Consistencia

Reducción de Código

- Centralización de lógica

Mejoras en la colaboración

- Claridad para el equipo
- Facilidad de uso

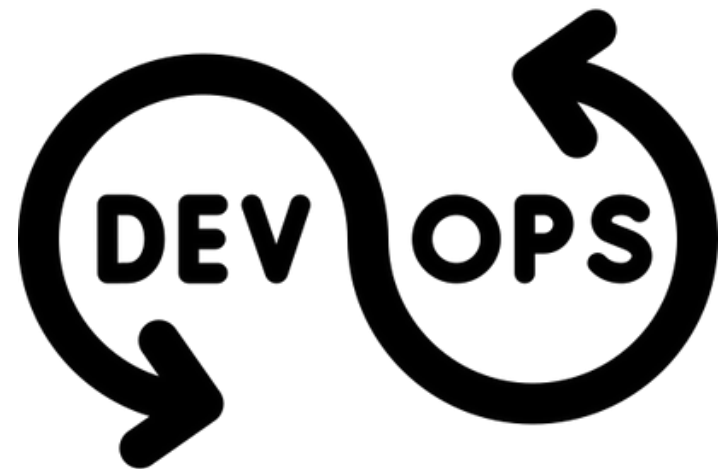


Estructura de una Composite Action



Estructura de una composite action

El archivo action.yml es el componente central de una Composite action en GitHub Actions. Este archivo define cómo se comporta la acción y especifica sus características, pasos, entradas y salidas



- Nombre (name)
- Descripción (Description)
- Entradas (inputs)
- Salidas (Outputs)
- Uso de composite
- Dependencias y Repositorio

Uso de runs para especificar multiples pasos



En el contexto de un composite action en GitHub Actions, la sección runs es donde se define cómo se ejecutará la acción y especificas múltiples pasos (steps) que se llevarán a cabo.

Ventajas de definir múltiples pasos

- Organización
- Reutilización
- Mantenibilidad
- Flexibilidad

```
Code Blame 32 lines (29 loc) · 835 Bytes

1  name: "Build"
2  description: "A composite action to set up and run a Python project"
3  inputs:
4    python-version:
5      description: "Python version to use"
6      required: true
7    requirements:
8      description: "Path to requirements file"
9      required: true
10 outputs:
11   test-result:
12     description: "Test results output"
13 runs:
14   using: "composite"
15   steps:
16     - name: Set up Python
17       uses: actions/setup-python@v5
18       with:
19         python-version: "${{ inputs.python-version }}"
20
21     - name: Install dependencies
22       run: |
23         python -m pip install --upgrade pip
24         pip install -r "${{ inputs.requirements }}"
25
26     - name: Run tests
27       run: |
28         pytest > result.log
29       id: run_tests
30
31     - name: Output test result
32       run: echo "::set-output name=test-result::$(cat result.log)"
```



Creacion de una Composite Action

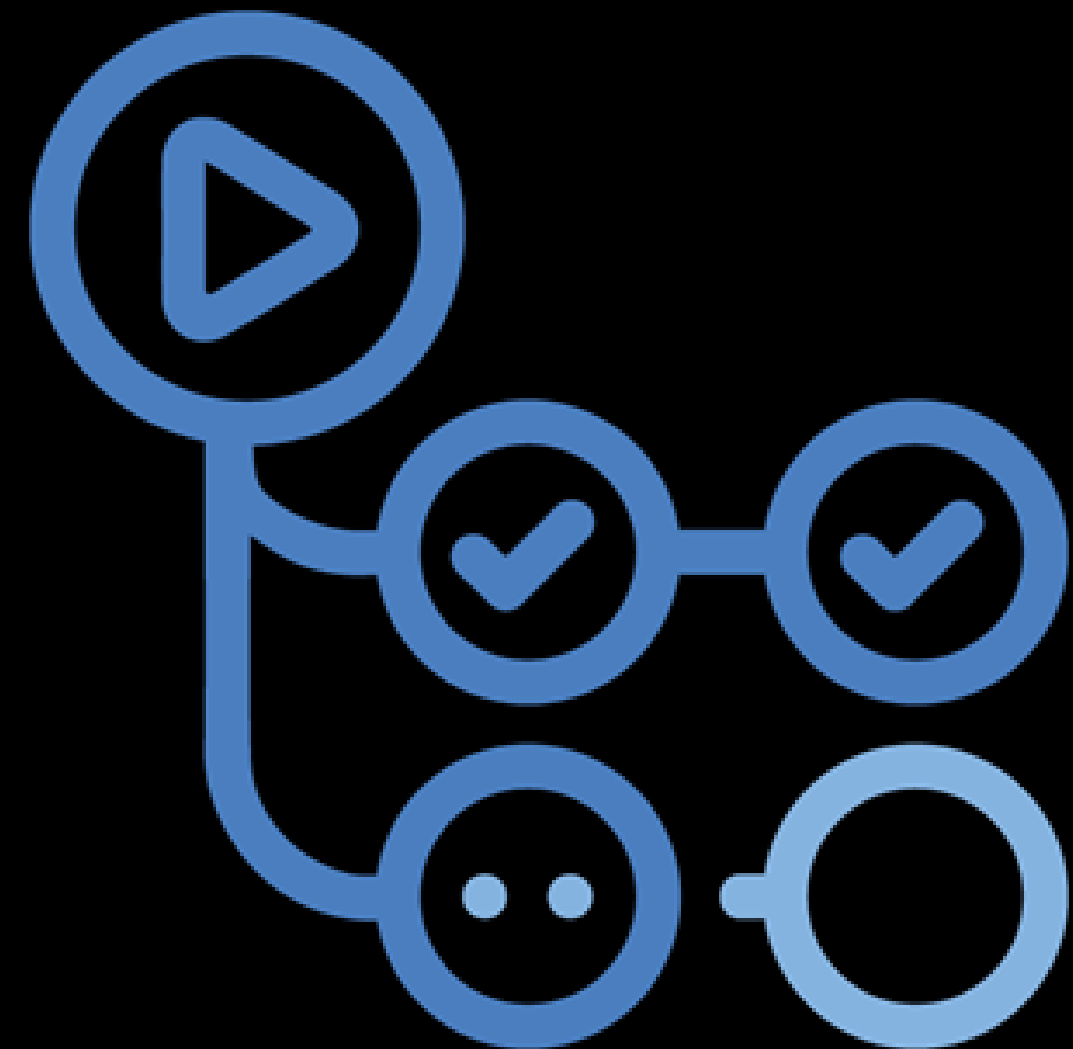


Pasos para crear una composite action

- Crear una composite action en GitHub Actions implica varios pasos.

Paso a Paso

- Crear la estructura de archivos (clonar – crear carpeta para la action
 - Crear estructura .github/actions/action.yml
- Definir action.yml (bloque de directivas)
- Crear un repositorio
- Publicar la action (confirmar y subir los cambios)
- Usar la action en un flujo de trabajo
- Verificar la ejecución





Demo

Creación de una Composite Action



Questions?



GitHub Composite Actions

