

# Integración de GitHub Actions con Docker usando docker/build-push-action

Julian F. Latorre

20 de agosto de 2024

## Índice

<b>1. Introducción</b>	<b>1</b>
<b>2. Requisitos Previos</b>	<b>1</b>
<b>3. Pasos a Seguir</b>	<b>2</b>
3.1. Crear un Repositorio en GitHub . . . . .	2
3.2. Configurar el Flujo de Trabajo de GitHub Actions . . . . .	2
3.3. Configurar Secretos en GitHub . . . . .	2
3.4. Probar el Flujo de Trabajo . . . . .	3
<b>4. Explicación del Flujo de Trabajo</b>	<b>3</b>

## 1. Introducción

La acción `docker/build-push-action` de GitHub Actions es una herramienta que permite construir y publicar imágenes de Docker directamente desde tu flujo de trabajo de GitHub. En esta guía, aprenderemos a integrar esta acción en un proyecto de GitHub para automatizar el proceso de construcción y publicación de imágenes de Docker.

## 2. Requisitos Previos

Antes de comenzar, asegúrate de tener lo siguiente:

- Una cuenta de GitHub.
- Una cuenta en Docker Hub.
- Docker instalado en tu máquina local.

## 3. Pasos a Seguir

### 3.1. Crear un Repositorio en GitHub

- Inicia sesión en tu cuenta de GitHub.
- Crea un nuevo repositorio o utiliza uno existente donde quieras integrar la acción `docker/build-push-action`.

### 3.2. Configurar el Flujo de Trabajo de GitHub Actions

- En tu repositorio de GitHub, crea un nuevo archivo `.github/workflows/docker-build-push.yml`.
- Agrega el siguiente contenido al archivo:

```
name: Docker Build and Push
on:
  push:
    branches: [ main ]
  jobs:
    build-and-push:
      runs-on: ubuntu-latest
      steps:
        - uses: actions/checkout@v3

        - name: Set up Docker Buildx
          uses: docker/setup-buildx-action@v1

        - name: Login to Docker Hub
          uses: docker/login-action@v3
          with:
            username: ${ secrets.DOCKER_USERNAME }
            password: ${ secrets.DOCKER_PASSWORD }

        - name: Build and push
          uses: docker/build-push-action@v6
          with:
            context: .
            push: true
            tags: |
              ${ secrets.DOCKER_USERNAME }/${ secrets.DOCKER_REPOSITORY }:latest
              ${ secrets.DOCKER_USERNAME }/${ secrets.DOCKER_REPOSITORY }:${ github.sha }
```

- Reemplaza `${ secrets.DOCKER_USERNAME }` y `${ secrets.DOCKER_REPOSITORY }` con tus propios valores.

### 3.3. Configurar Secretos en GitHub

- En tu repositorio de GitHub, ve a la sección "Settings» "Secrets".
- Crea los siguientes secretos:

- `DOCKER_USERNAME`: Tu nombre de usuario de Docker Hub.
- `DOCKER_PASSWORD`: Tu contraseña de Docker Hub.
- `DOCKER_REPOSITORY`: El nombre del repositorio de Docker Hub donde publicarás las imágenes.

### 3.4. Probar el Flujo de Trabajo

- Realiza un cambio en el código de tu aplicación y haz un commit a la rama `main`.
- GitHub Actions se encargará de construir y publicar la imagen de Docker en tu registro.
- Verifica que la imagen se haya publicado correctamente en tu cuenta de Docker Hub.

## 4. Explicación del Flujo de Trabajo

Veamos en detalle lo que hace cada paso del flujo de trabajo:

- `actions/checkout@v3`: Este paso descarga el código fuente de tu aplicación en el entorno de GitHub Actions.
- `docker/setup-buildx-action@v1`: Configura el entorno de construcción de Docker, habilitando la funcionalidad de `buildx` para construir imágenes de manera más eficiente.
- `docker/login-action@v3`: Autentica tu cuenta de Docker Hub utilizando los secretos configurados anteriormente.
- `docker/build-push-action@v6`: Este es el paso principal que construye y publica la imagen de Docker. Aquí se especifica el contexto de construcción, se habilita el envío de la imagen (push) y se definen las etiquetas de la imagen.