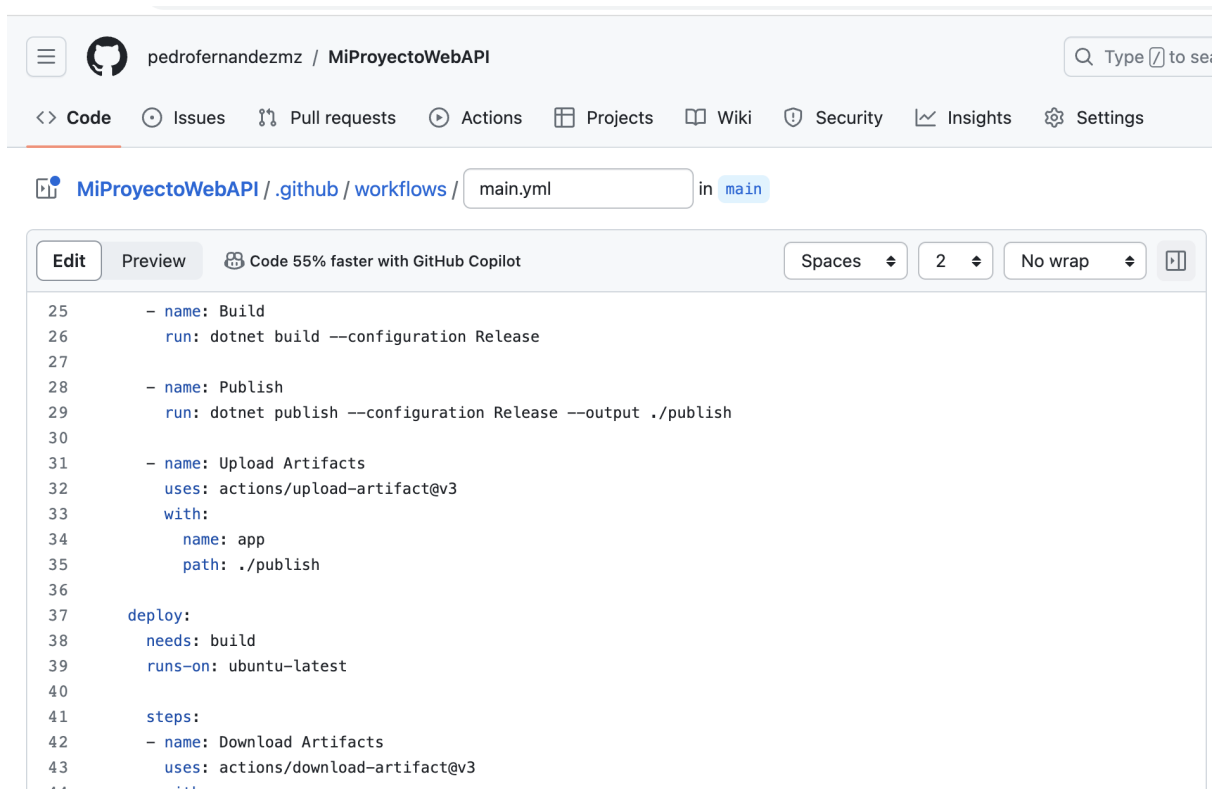


# Trabajo Práctico 8 - Herramientas de construcción de software en la nube

## Desarrollo:

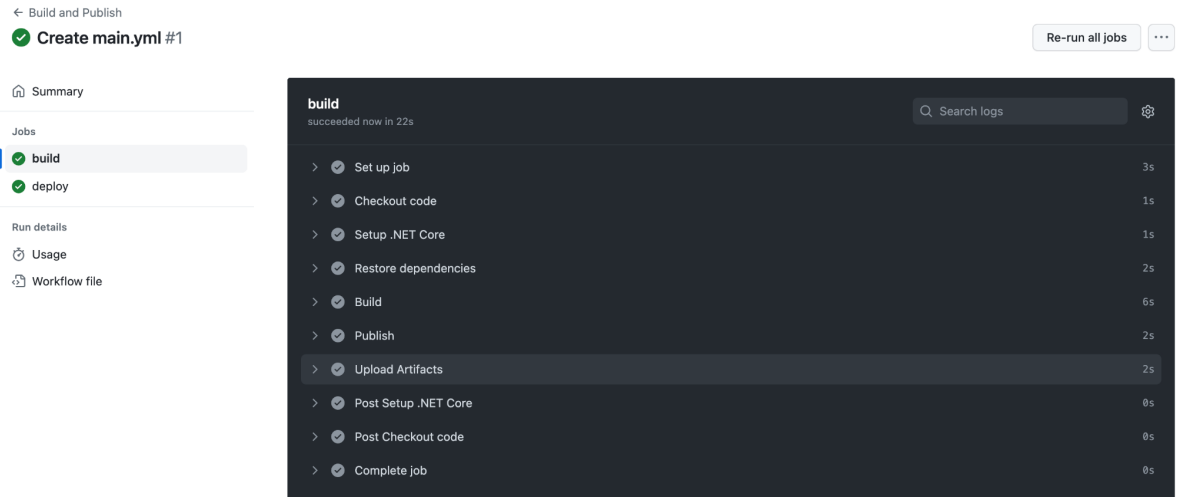
### Configurando GitHub Actions

- Repetir el ejercicio 6.1 del trabajo práctico [trabajo práctico 7](#) para el proyecto SimpleWebAPI, pero utilizando GitHub Actions.
- En GitHub, en el repositorio donde se encuentra la aplicación SimpleWebAPI, ir a la opción Actions y crear un nuevo `workflow`.
- El nombre de archivo puede ser `main.yml` y tendrá un contenido similar al siguiente:



```
25     - name: Build
26       run: dotnet build --configuration Release
27
28     - name: Publish
29       run: dotnet publish --configuration Release --output ./publish
30
31     - name: Upload Artifacts
32       uses: actions/upload-artifact@v3
33       with:
34         name: app
35         path: ./publish
36
37     deploy:
38       needs: build
39       runs-on: ubuntu-latest
40
41     steps:
42     - name: Download Artifacts
43       uses: actions/download-artifact@v3
44       with:
```

- Guardar el archivo (hacemos commit directamente en GitHub por ejemplo) y vemos la ejecución del worflow y sus logs.



- Explicar paso a paso que realiza el pipeline anterior.

El pipeline en el archivo `main.yml` es un workflow de GitHub Actions que realiza una serie de pasos para compilar una aplicación .NET Core, publicarla y luego, en una etapa de implementación, descargar los artefactos generados y realizar una acción de implementación (en este caso, simplemente muestra un mensaje "Deploy").

La primera sección configura cuándo se ejecutará el flujo de trabajo. En este caso, se ejecutará cuando se active manualmente (`workflow_dispatch`) y cuando se haga un push a la rama `main`.

Luego se define un trabajo llamado "build" que se ejecutará en una máquina Ubuntu de última versión (`ubuntu-latest`) y utiliza la acción `actions/checkout@v3` para clonar el repositorio de GitHub en la máquina de ejecución.

En el siguiente paso "Setup .NET Core" y utiliza la acción `actions/setup-dotnet@v3` para configurar el entorno .NET Core en la máquina de ejecución. En este caso, se configura la versión `7.0.x`.

"Publish" ejecuta `dotnet publish` con la configuración de "Release" y especifica que los archivos publicados se guardarán en el directorio `./publish`.







Esta primera parte del flujo de trabajo se encarga de construir y publicar la aplicación .NET Core. Luego, pasa a la etapa de implementación

"Deploy to Server" ejecuta una acción de implementación, que en este caso es muy básica y muestra el mensaje "Deploy". En una implementación real, este sería el lugar donde se realizaría la implementación real en un servidor.

### 3- Configurar un worflow en GitHub Actions para generar una imagen de Docker de SimpleWebApi y subirla a DockerHub

- En GitHub Actions generar una acción que genere una imagen de docker con nuestra aplicación SimpleWebAPI y la suba a DockerHub
- Generar `secretos` y los `pasos` necesarios para subir la imagen a Docker Hub.  
[Referencia](#)
- Paso 1: Configurar las credenciales de Docker Hub en tu repositorio de GitHub:

En tu repositorio de GitHub, ve a "Settings" (Configuración) > "Secrets" (Secretos). Haz clic en "New repository secret" (Nuevo secreto del repositorio). Define dos secretos: uno para el nombre de usuario de Docker Hub y otro para la contraseña de Docker Hub. Puedes nombrar estos secretos como DOCKERHUB\_USERNAME y DOCKERHUB\_PASSWORD, respectivamente.

Repository secrets		
 DOCKERHUB_PASSWORD	Updated now	 
 DOCKERHUB_USERNAME	Updated now	 

- Paso 2: Crear un workflow para construir y subir la imagen de Docker:

main
MiProyectoWebAPI / .github / workflows / docker.yml

pedrofernandezmz
Create docker.yml
✕

Code
Blame
27 lines (21 loc) · 741 Bytes
Code 55% faster with GitHub Copilot

```

1  name: Docker Image CI
2
3  on:
4    workflow_dispatch:
5    push:
6      branches: [ "main" ]
7
8  jobs:
9    build:
10     runs-on: ubuntu-latest
11     steps:
12       - name: Checkout code
13         uses: actions/checkout@v2
14
15       - name: Build the Docker image
16         run: docker build . --file Dockerfile --tag ${ secrets.DOCKERHUB_USERNAME }}/simple-web-api-gh:latest
17
18       - name: Log in to Docker Hub
19         run: docker login -u ${ secrets.DOCKERHUB_USERNAME } -p ${ secrets.DOCKERHUB_PASSWORD }
20
21       - name: Push Docker image to Docker Hub
22         run: |
23           docker push ${ secrets.DOCKERHUB_USERNAME }}/simple-web-api-gh:latest
24
25       - name: Clean up

```

Name	Last commit message
..	
docker.yml	Create docker.yml
main.yml	Create main.yml

- Paso 3: Verificar en DockerHub que la imagen ha sido subida

Search Docker Hub
Explore Repositories Organizations Help

piterfmz
Search by repository name
All Content
Create repository

**piterfmz / simple-web-api-gh**  
Contains: Image | Last pushed: a minute ago

Inactive ☆ 0 📦 0 Public

**piterfmz / test-node**  
Contains: Image | Last pushed: 5 days ago

Inactive ☆ 0 📦 5 Public

- Paso 4: Descargar la imagen

```

Last login: Tue Oct  3 15:00:53 on ttys000
[pedrofernandez@MacBook-Air-de-Pedro ~ % docker pull piterfmz/simple-web-api-gh:latest
latest: Pulling from piterfmz/simple-web-api-gh
7dbc1adf280e: Downloading [=====>] 19.1MB/31.42MB
969d48310aaa: Download complete
2194c6af6861: Downloading [=====>] 17.54MB/32.45MB
98003354b5af: Downloading [=====>] 153B/153B
a5db88be08ca: Downloading [=====>] 2.925MB/10.12MB
e3c1b00f0a67: Waiting
4f4fb70ef54: Waiting
0bffb7f7118b: Waiting

```

- Paso 5: Crear el contenedor

```

pedrofernandez@MacBook-Air-de-Pedro ~ % docker run --name myapi -d -p 8080:80 --platform linux/amd64 piterfmz/simple-web-api-gh
07bcb20d0f90118ce23c6cf66f3f0d7279a8c29b96075edc42b8c956ab5fd33c
pedrofernandez@MacBook-Air-de-Pedro ~ %

```

- Paso 6: Navegar a <http://localhost:8080/weatherforecast>

```

[{"date": "2023-09-21", "temperature": -13, "summary": "Chilly"}, {"date": "2023-09-22", "temperature": 33, "summary": "Sweating"}, {"date": "2023-09-23", "temperature": -3, "summary": "Warm"}, {"date": "2023-09-24", "temperature": -19, "summary": "Chilly"}, {"date": "2023-09-25", "temperature": -2, "summary": "Warm"}]

```

## 4- Crear una GitHub Action que genere los artefactos para el proyecto React

- En GitHub Actions generar una acción que genere los artefactos para el Ejercicio 2 del TP 5

react-app / .github / workflows / build-react-app.yml

pedrofernandezmz Create build-react-app.yml

Code Blame 34 lines (26 loc) · 926 Bytes Code 55% faster with GitHub Copilot







```

1  name: Build React App
2
3  on:
4    push:
5      branches:
6        - main # Cambia "main" por la rama en la que deseas que se ejecute la acción.
7
8  jobs:
9    build:
10     runs-on: ubuntu-latest
11
12     steps:
13       - name: Checkout código fuente
14         uses: actions/checkout@v2
15
16       - name: Configurar Node.js
17         uses: actions/setup-node@v2
18         with:
19           node-version: '14' # Cambia a la versión de Node.js que estás utilizando.
20
21       - name: Instalar dependencias

```

## 5- Crear una GitHub Action que genere una imagen de Docker para el proyecto React y lo suba a DockerHub


- En GitHub Actions generar una acción que genere una imagen de Docker para el Ejercicio 2 del TP 5 y la suba a DockerHub

Repository secrets		
	DOCKERHUB_PASSWORD	Updated now  
	DOCKERHUB_USERNAME	Updated now  

### Workflow file for this run

.github/workflows/main.yml at ba1f591

```
1  name: Build React App
2
3  on:
4    push:
5      branches:
6        - main # Cambia "main" por la rama en la que deseas que se ejecute el workflow.
7
8  jobs:
9    build:
10     runs-on: ubuntu-latest
11
12     steps:
13       - name: Checkout código fuente
14         uses: actions/checkout@v2
15
16       - name: Configurar Node.js
17         uses: actions/setup-node@v2
18         with:
19           node-version: '14' # Cambia a la versión de Node.js que estás utilizando.
20
21       - name: Instalar dependencias
22         run: npm install
23
```

 dockerhub

Search Docker Hub

Explore Repositories Organizations Help

piterfmz

Search by repository name

All Content

Create repository

piterfmz / react-app

Contains: No content | Last pushed: a few seconds ago

Inactive ☆ 0 📄 0 🌐 Public