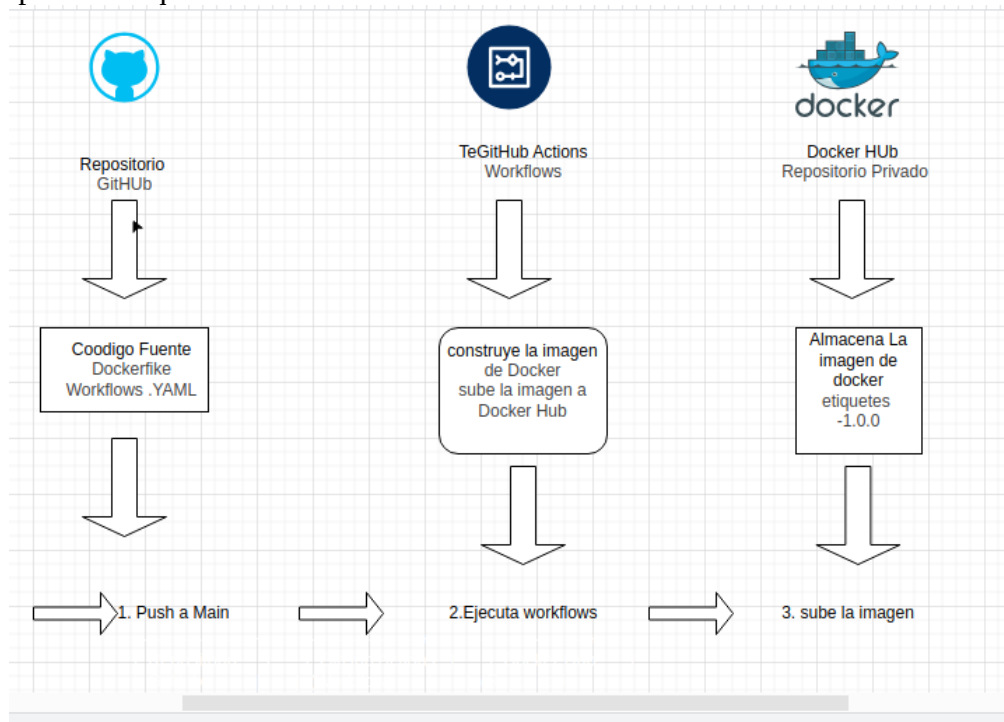


Informe sobre la Implementación del Desafío N°9

Responsable: Acosta Hernán Andrés

Diagrama de Alto Nivel y Explicación

Para abordar el Desafío N°9, se elaboró un diagrama de alto nivel que proporciona una visión general del proceso requerido.



Componentes y Flujo de Trabajo

Repositorio GitHub

Contiene el código fuente de la aplicación, incluyendo archivos esenciales como `app.js` y `package.json`.

Incluye un `Dockerfile` para la construcción de la imagen de Docker.

Se configura un workflow de GitHub Actions en `.github/workflows/docker-build-push.yml` para la automatización del proceso.

GitHub Actions

Se activa al realizar un push en la rama `main`.

Construye la imagen de Docker utilizando el `Dockerfile`.

Sube la imagen a Docker Hub con las etiquetas `1.0.0` y `latest`.

Docker Hub

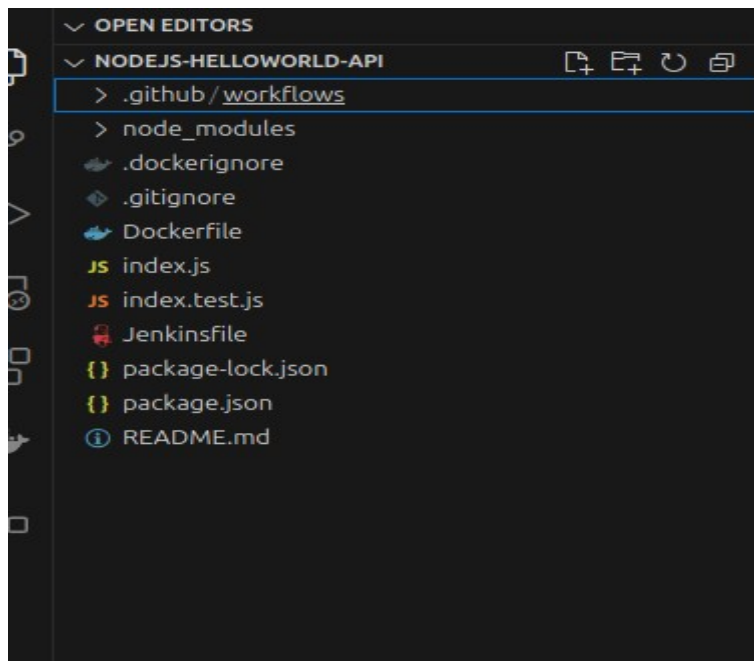
Almacena la imagen de Docker en un repositorio privado.

Permite el versionado de la imagen con etiquetas.

Pasos Realizados

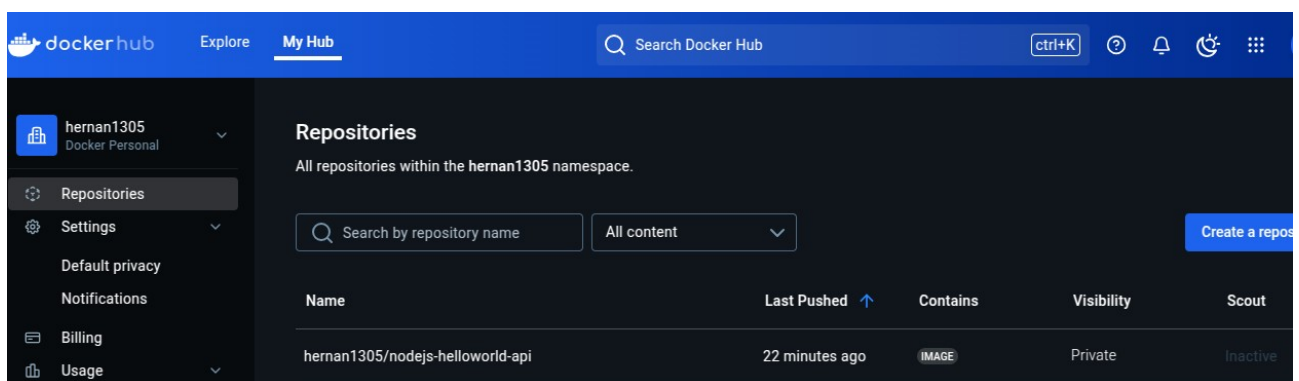
Creación del Repositorio en GitHub

Se generó un nuevo repositorio basado en el template `nodejs-helloworld-api`.



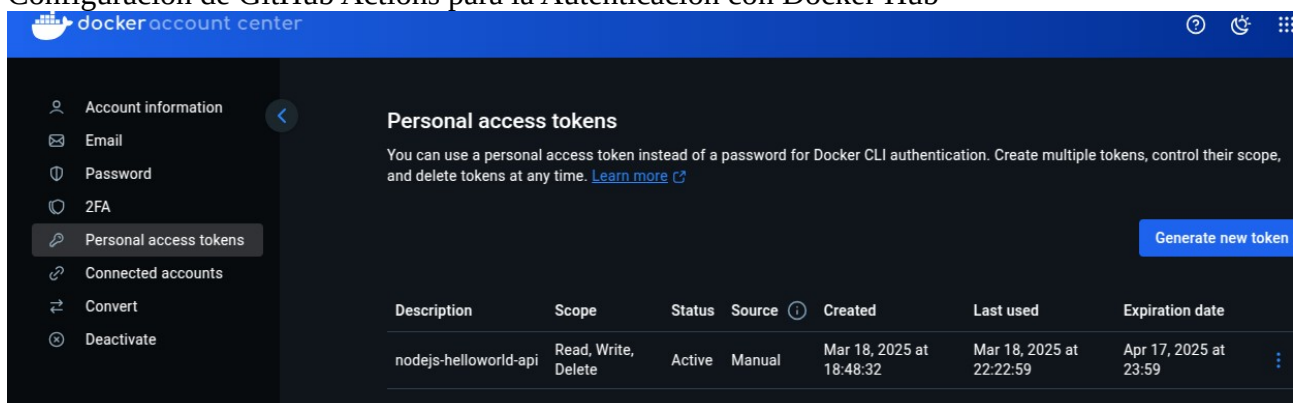
Creación del Repositorio Privado en Docker Hub

Se estableció un repositorio privado con el nombre herman1305/nodejs-helloworld-api.

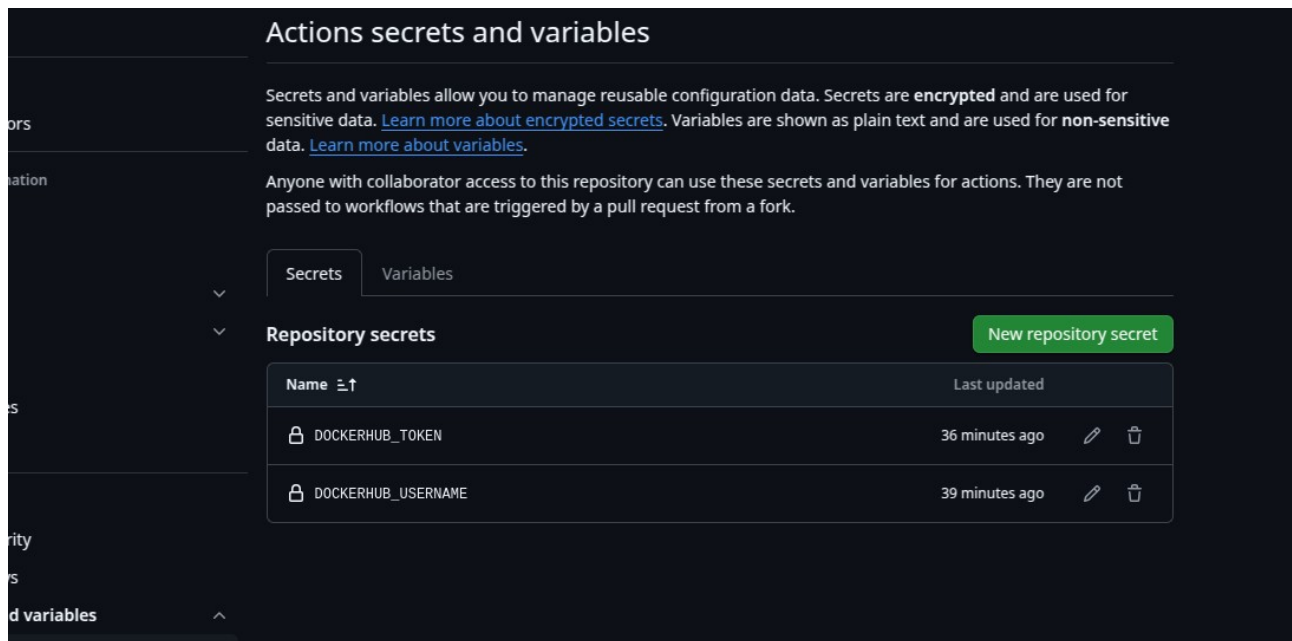


Generación del Token en Docker Hub

Se generó un Token de Acceso Personal (PAT) con permisos de lectura y escritura. Configuración de GitHub Actions para la Autenticación con Docker Hub



Se añadieron los secrets DOCKERHUB_USERNAME y DOCKERHUB_TOKEN en GitHub.



Se incluyó el siguiente Dockerfile en la raíz del repositorio:

Creación del Dockerfile

dockerfile

```
# Usar una imagen base de Node.js
```

```
FROM node:16
```

```
# Establecer el directorio de trabajo dentro del contenedor
```

```
WORKDIR /app
```

```
# Copiar los archivos de dependencias
```

```
COPY package*.json ./
```

```
# Instalar las dependencias
```

```
RUN npm install
```

```
# Copiar el resto de los archivos de la aplicación
```

```
COPY . .
```

```
# Exponer el puerto de la aplicación
```

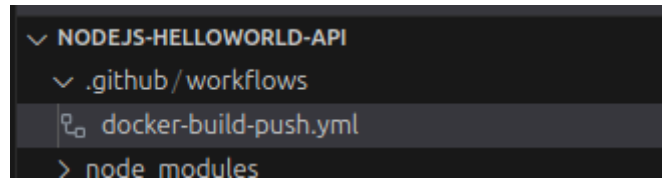
```
EXPOSE 3000
```

```
# Comando para ejecutar la aplicación
```

```
CMD ["npm", "start"]
```

Implementación del Workflow en GitHub Actions

Se configuró el archivo `.github/workflows/docker-build-push.yml` para construir y subir la imagen en múltiples plataformas (linux/amd64 y linux/arm64).



name: Docker Build and Push

on:

push:

branches:

- main

jobs:

build-and-push:

runs-on: ubuntu-latest

steps:

- name: Checkout repository

uses: actions/checkout@v3

- name: Log in to Docker Hub

uses: docker/login-action@v2

with:

username: \${{ secrets.DOCKERHUB_USERNAME }}

password: \${{ secrets.DOCKERHUB_TOKEN }}

- name: Set up Docker Buildx

uses: docker/setup-buildx-action@v2

- name: Build and push

uses: docker/build-push-action@v4

with:

context: .

platforms: linux/amd64, linux/arm64

push: true

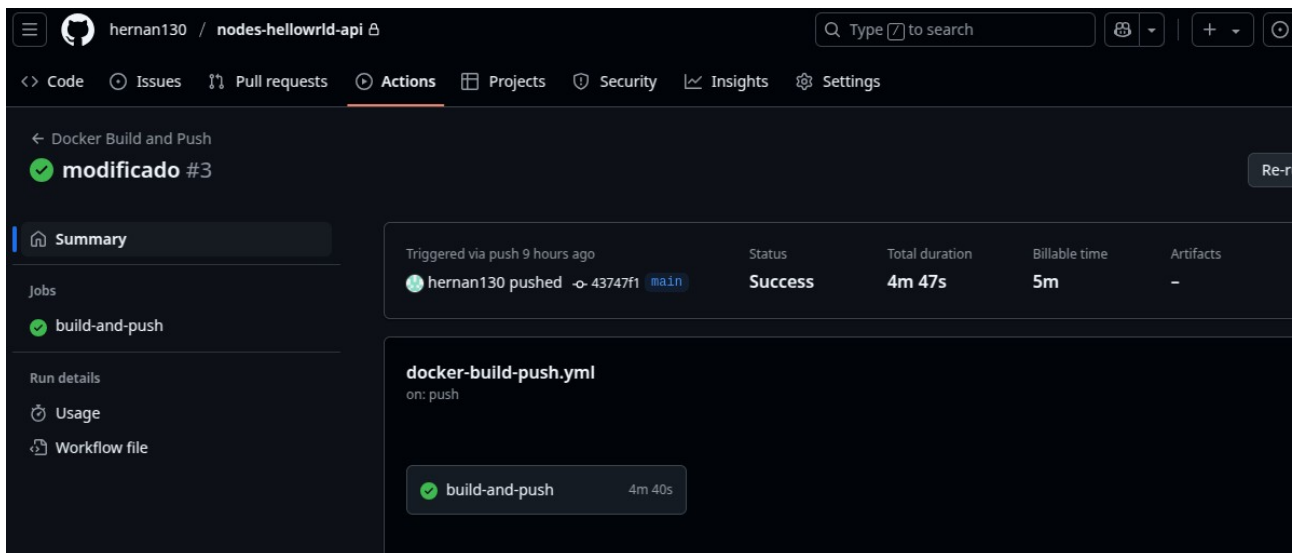
tags: |

hernan1305/nodejs-helloworld-api:1.0.0

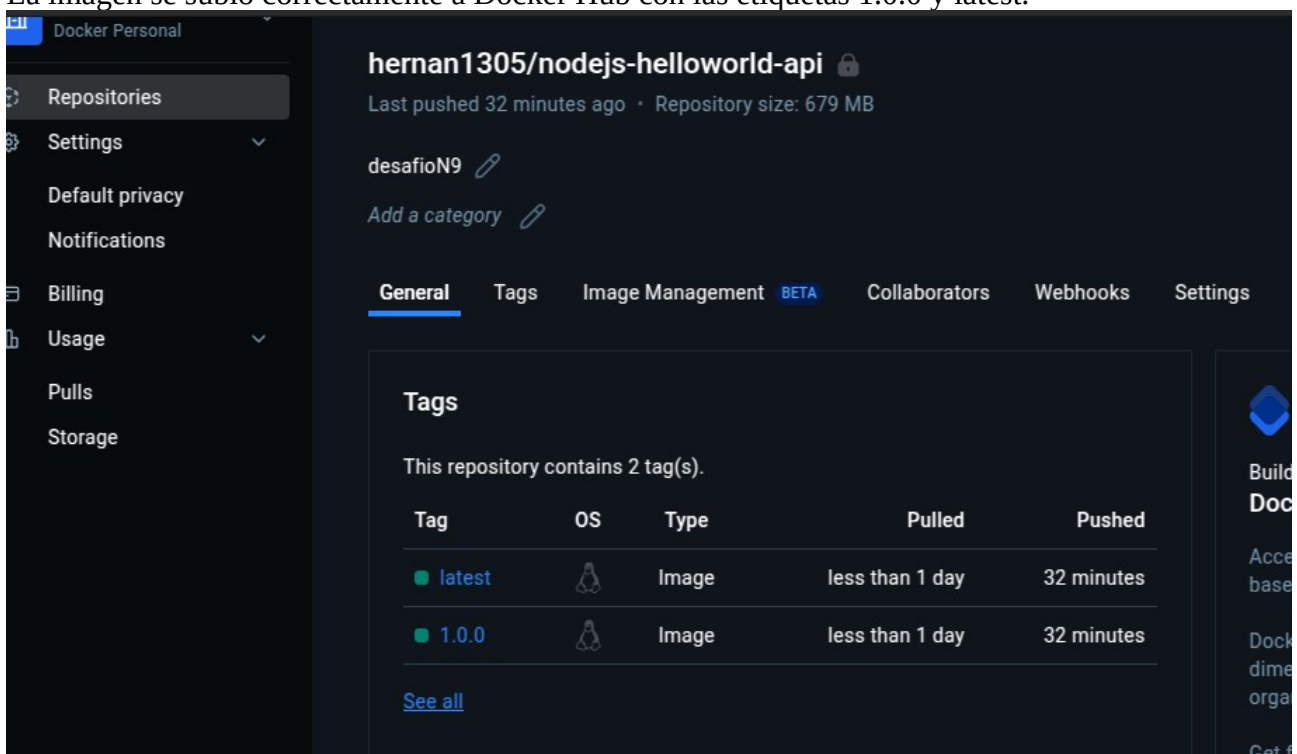
hernan1305/nodejs-helloworld-api:latest

Prueba del Workflow

Se realizó un push a la rama main y el workflow se ejecutó exitosamente.



La imagen se subió correctamente a Docker Hub con las etiquetas 1.0.0 y latest.



Verificación Adicional

Prueba Local de la Imagen de Docker

Construcción de la imagen localmente:

`docker build -t hernan1305/nodejs-helloworld-api:1.0.0 .`

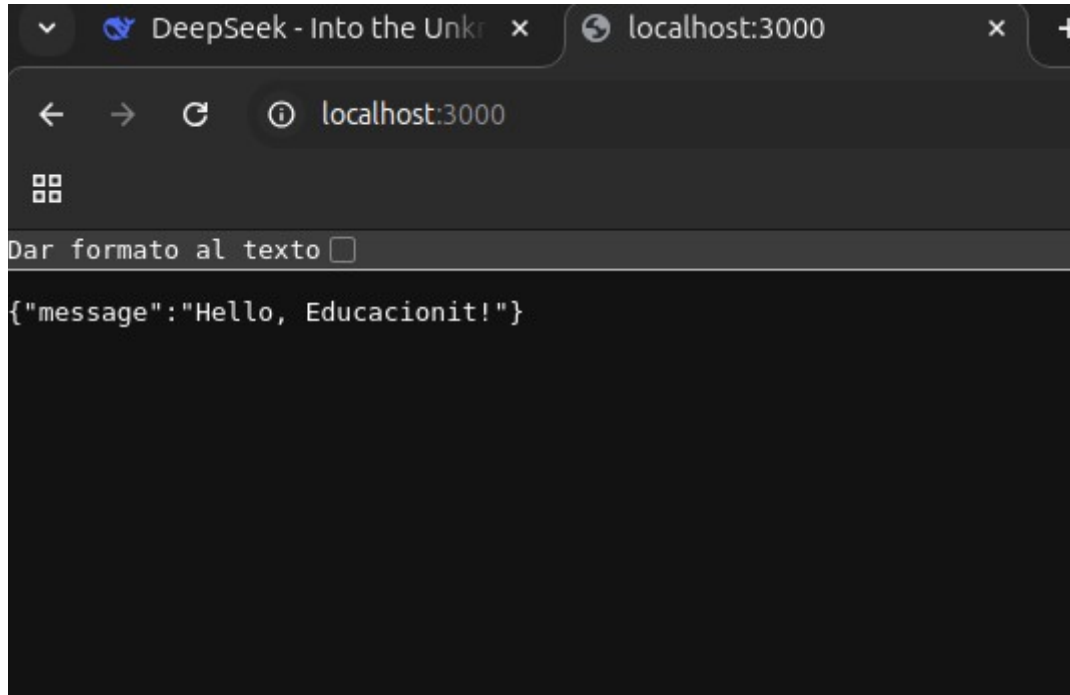
Ejecución del contenedor:

`docker run -p 3000:3000 hernan1305/nodejs-helloworld-api:1.0.0`

Verificación del funcionamiento:

Se accedió a la aplicación en `http://localhost:3000`, confirmando que respondió correctamente con el mensaje esperado.

```
hernan@andres:~/nodejs-helloworld-api$ docker run -p 3000:3000 hernan1305/nodejs-helloworld-api:1.0.0
> nodejs-helloworld-api@1.0.0 start
> node index.js
Server is listening on port 3000
```



Conclusión

El proceso de configuración y automatización del despliegue de la imagen Docker se llevó a cabo con éxito. La integración de GitHub Actions permitió un flujo de trabajo eficiente, garantizando la generación y publicación automática de la imagen en Docker Hub cada vez que se realice un push en la rama main. Las pruebas locales confirmaron el correcto funcionamiento de la aplicación dentro del contenedor.