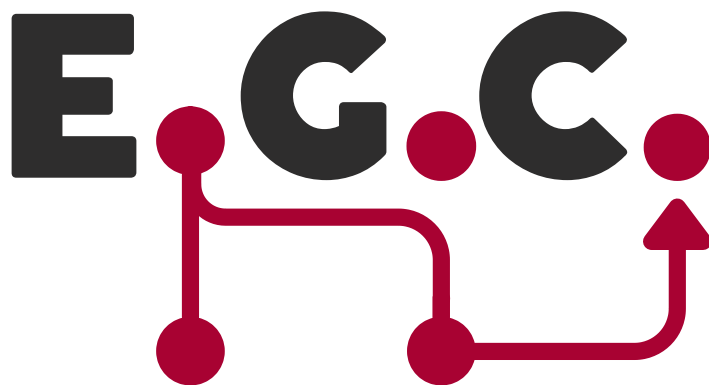




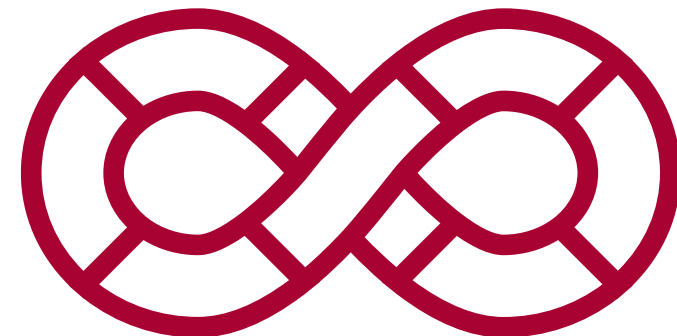
Grado en Ingeniería Informática – Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática

Práctica 2 Integración y despliegue continuos



1. **Introducción a CI/CD**
 2. **GitHub Actions**
 3. **Integración Continua (CI) en uvlhub.io**
 4. **Despliegue Continuo (CD) en uvlhub.io**
 5. **Ejercicio práctico CI: Codacy**
 6. **Ejercicio práctico CD: Render**
 7. **Y yo, ¿qué puedo hacer en el proyecto?**
- 

- 1. Introducción a CI/CD**
 - 2. GitHub Actions**
 - 3. Integración Continua (CI) en uvlhub.io**
 - 4. Despliegue Continuo (CD) en uvlhub.io**
 - 5. Ejercicio práctico CI: Codacy**
 - 6. Ejercicio práctico CD: Render**
 - 7. Y yo, ¿qué puedo hacer en el proyecto?**
- 

1. Introducción a CI/CD

INTEGRACIÓN CONTINUA (CI)

- **Integramos** el código en un repositorio compartido de forma **frecuente**
- **Detectamos** errores rápidamente
- **Automatizamos pruebas** para garantizar que el nuevo código no rompa el sistema

DESPLIEGUE CONTINUO (CD)

- **Automatizamos** el proceso de lanzamiento de las aplicaciones
- Los **cambios que pasan las pruebas** en CI **se despliegan** en producción
- **Entregas rápidas y confiables** de nuevas versiones del software.

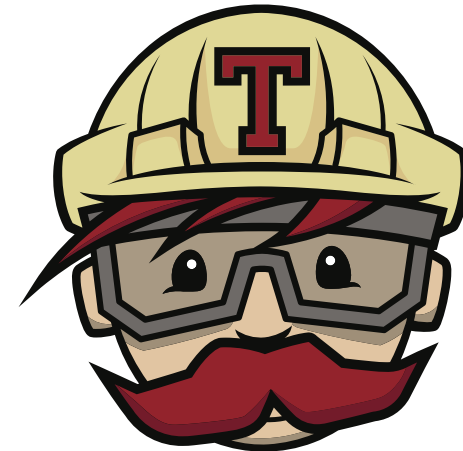
1. Introducción a CI/CD

Algunas herramientas CI



Jenkins

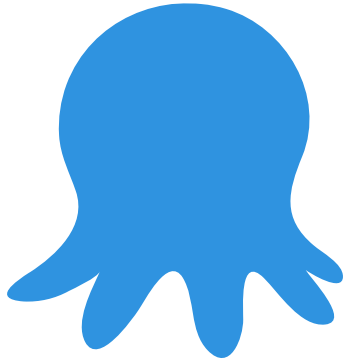
GitHub
actions



travis

1. Introducción a CI/CD

Algunas herramientas CD



Octopus Deploy



Argo CD

GitHub
actions



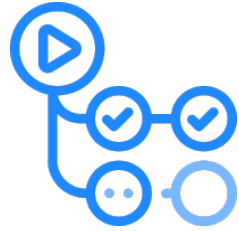
Google Cloud Build

1. Introducción a CI/CD
 - 2. GitHub Actions**
 3. Integración Continua (CI) en uvlhub.io
 4. Despliegue Continuo (CD) en uvlhub.io
 5. Ejercicio práctico CI: Codacy
 6. Ejercicio práctico CD: Render
 7. Y yo, ¿qué puedo hacer en el proyecto?
- 

2. GitHub Actions

¿Qué es?

GitHub
actions



Herramienta de automatización que permite ejecutar tareas de manera automática dentro de un proyecto en GitHub

Facilita procesos como la **integración continua (CI)** y el **despliegue continuo (CD)**, sin intervención manual

docs.github.com/es/actions



2. GitHub Actions | elementos

Workflows

workflow_1.yml

workflow_2.yml

...

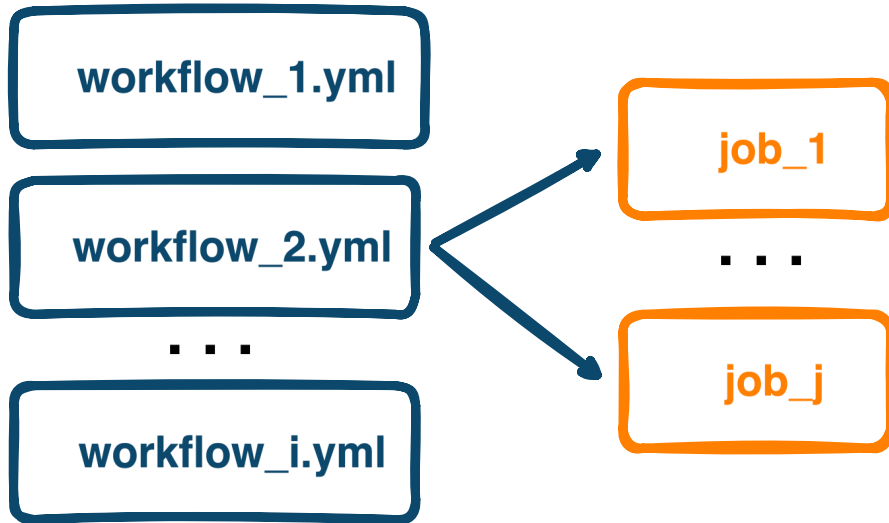
workflow_i.yml

Un **workflow** es un conjunto de tareas automatizadas que GitHub ejecuta cuando se cumplen ciertas condiciones

Los workflows se definen en archivos con formato **.yml** dentro del repositorio (*/.github/workflows*)

2. GitHub Actions | elementos

Jobs

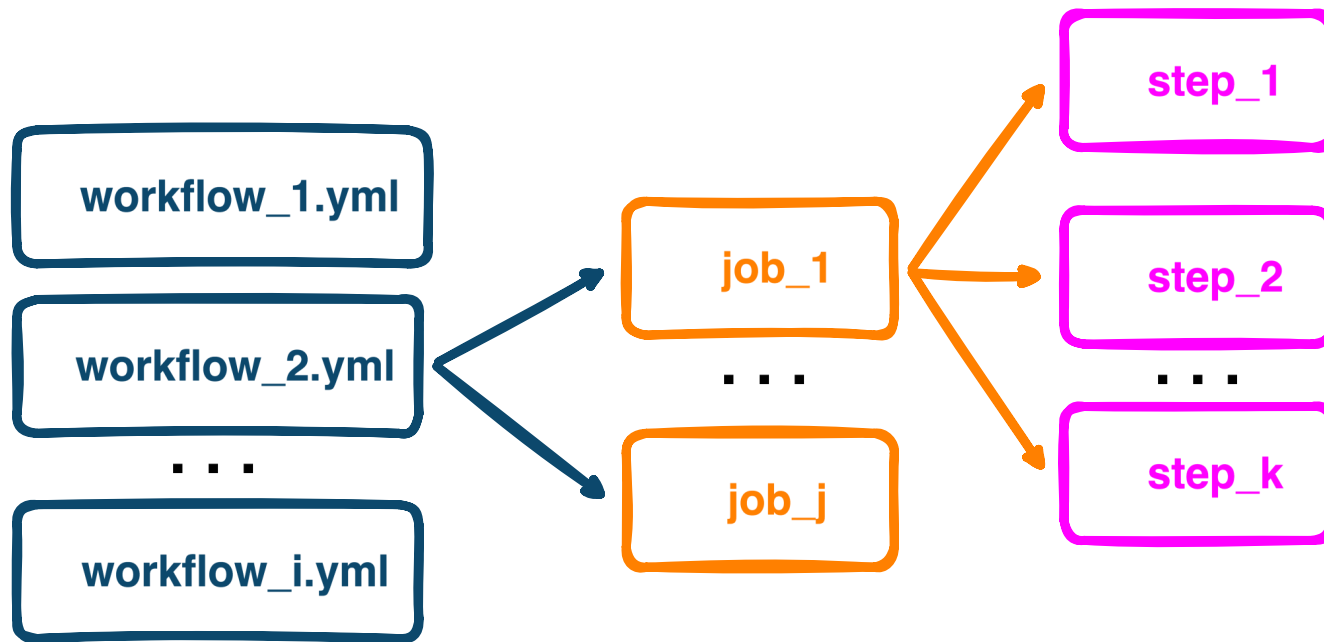


Un **job** es una unidad de trabajo dentro del workflow

Cada workflow puede tener uno o varios jobs, que se ejecutan de **manera paralela o secuencial**

2. GitHub Actions | elementos

Steps

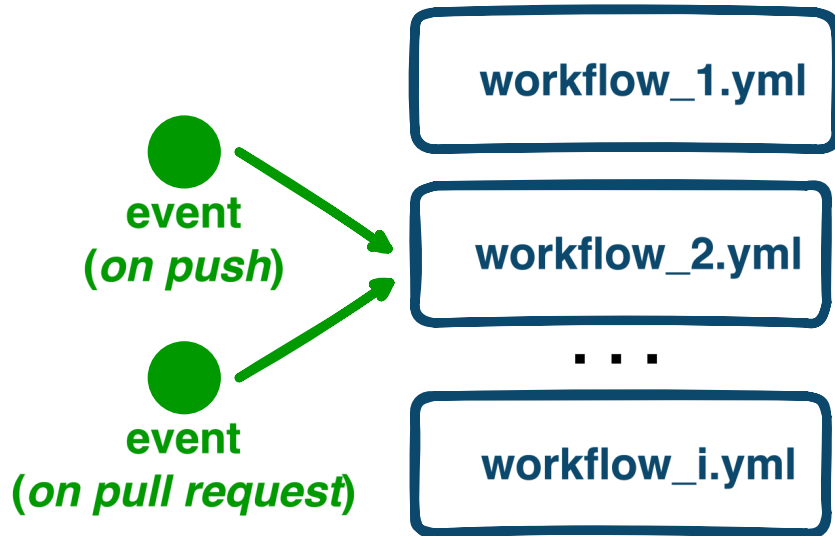


Cada job está compuesto por **steps** o pasos, que **describen acciones específicas** que deben realizarse, como instalar dependencias o ejecutar comandos

2. GitHub Actions | elementos

Eventos disparadores

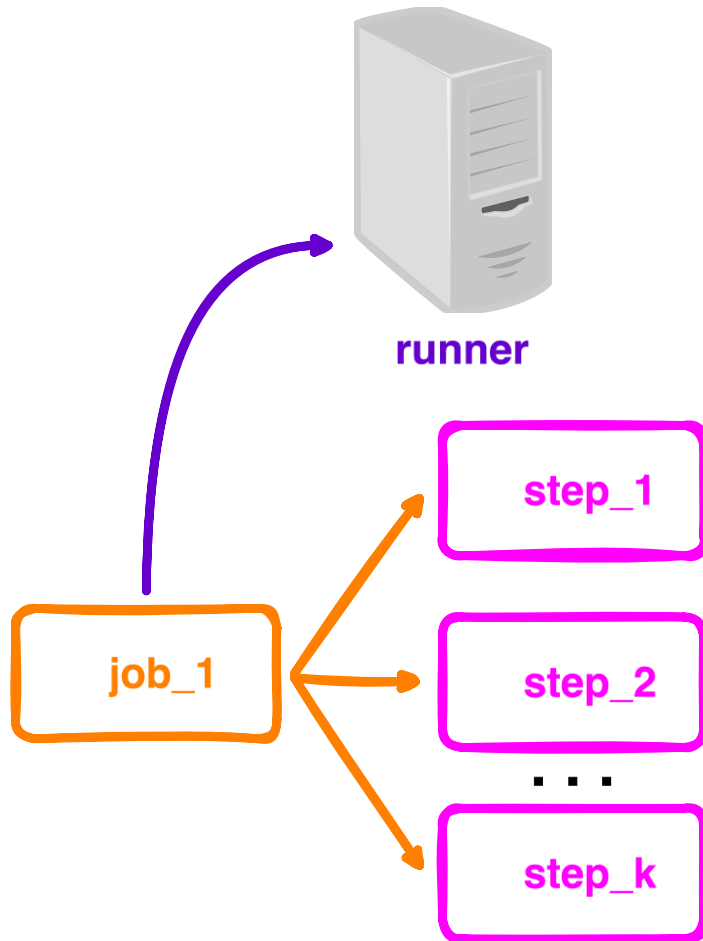
Un **evento disparador** es la acción que inicia la ejecución de un workflow



- **push**
- **pull_request**
- **workflow_run**
- **workflow_dispatch**
- **schedule**
- **issue_comment**

2. GitHub Actions | elementos

Runners

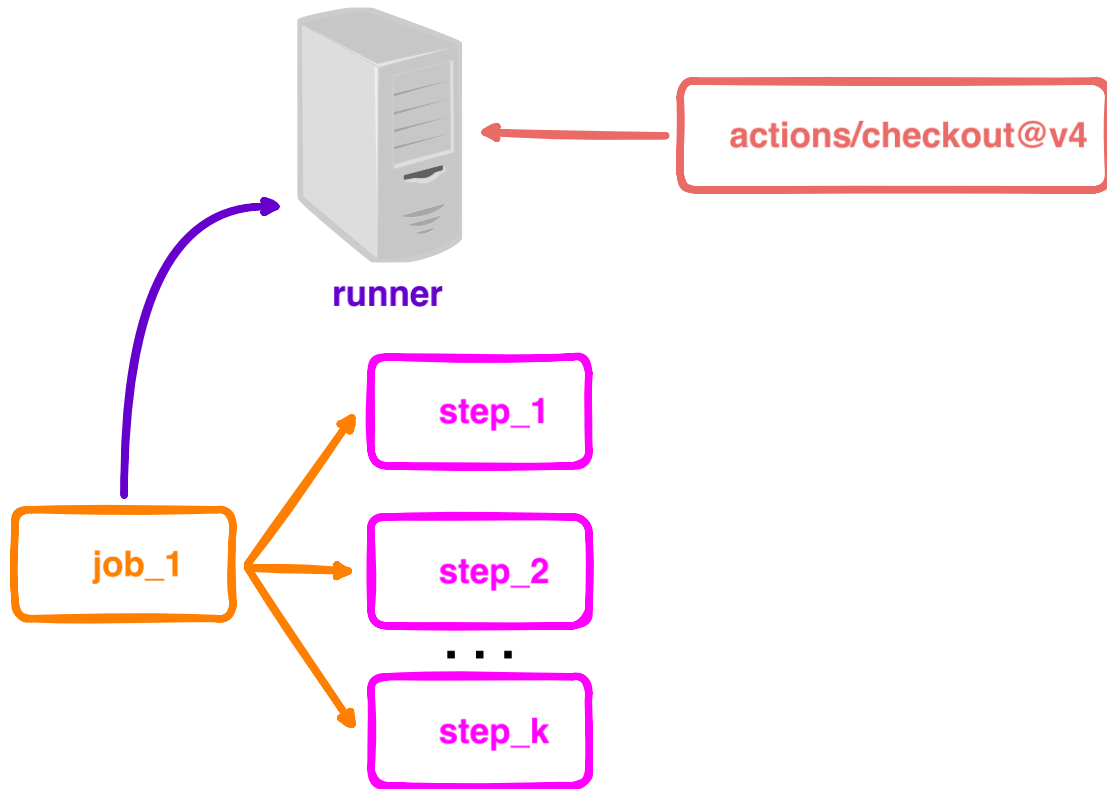


Un **runner** es el entorno o máquina donde se ejecutan los jobs de un workflow

- Cada **job** dentro del workflow se ejecuta en su **propio runner**
- El runner es responsable de ejecutar los steps del job, como instalar dependencias, correr pruebas...

2. GitHub Actions | elementos

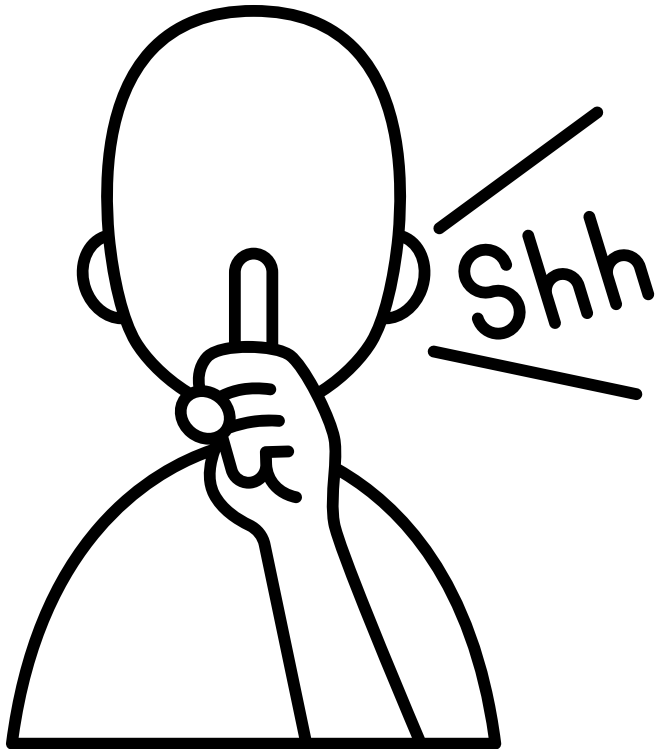
Acción checkout@v1/2/3/4/5...



El **checkout** es una acción predefinida de GitHub Actions que se utiliza para **clonar** el código de tu repositorio en el entorno del runner antes de ejecutar cualquier job

2. GitHub Actions | elementos

Secretos



Los **secretos** en GitHub Actions son **variables de entorno sensibles**, como claves API o credenciales, tokens...

Se almacenan de manera segura y se usan en los flujos de trabajo (**workflows**)

```
${{ secrets.INFO_SECRETATA }}
```

Eventos disparadores

*push, pull_request,
workflow_run,
workflow_dispatch, schedule,
issue_comment*

Workflow

*Conjunto de tareas
automatizadas
definidas en un
.yml*

Jobs

*Unidad de trabajo
dentro de un
workflow.
Ejecución paralela
o secuencial*

Steps

*Unidad de trabajo
dentro de un job.
Son acciones
específicas*

Runner

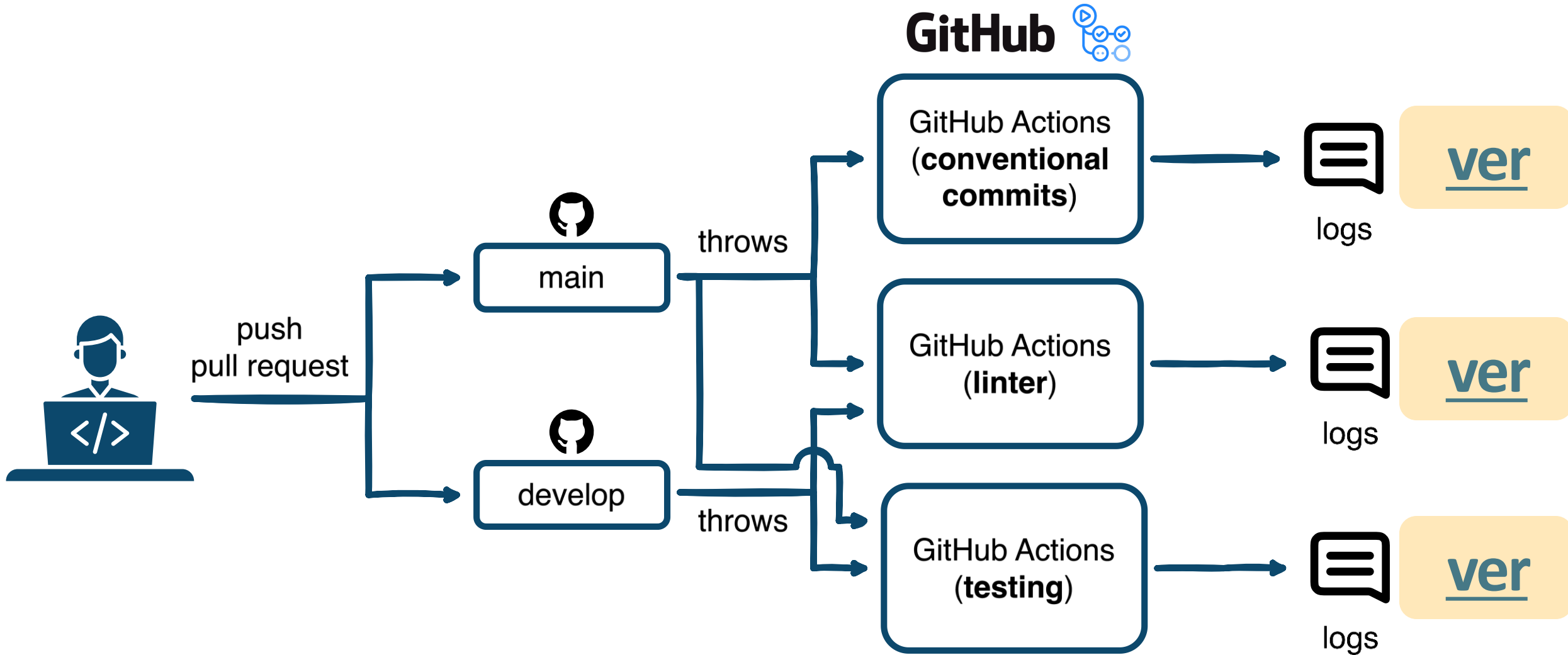
Entorno de ejecución de jobs

Secrets

Variables de entorno sensibles

1. Introducción a CI/CD
2. GitHub Actions
3. Integración Continua (CI) en uvlhub.io
4. Despliegue Continuo (CD) en uvlhub.io
5. Ejercicio práctico CI: Codacy
6. Ejercicio práctico CD: Render
7. Y yo, ¿qué puedo hacer en el proyecto?

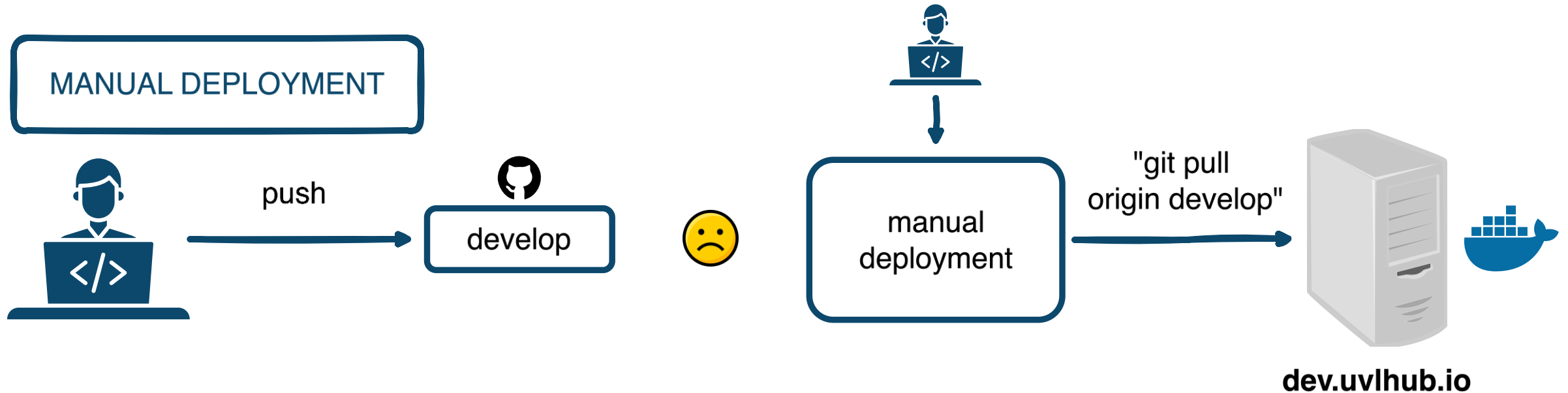
3. Integración Continua (CI) en uvlhub.io



1. Introducción a CI/CD
2. GitHub Actions
3. Integración Continua (CI) en uvlhub.io
4. **Despliegue Continuo (CD) en uvlhub.io**
5. Ejercicio práctico CI: Codacy
6. Ejercicio práctico CD: Render
7. Y yo, ¿qué puedo hacer en el proyecto?

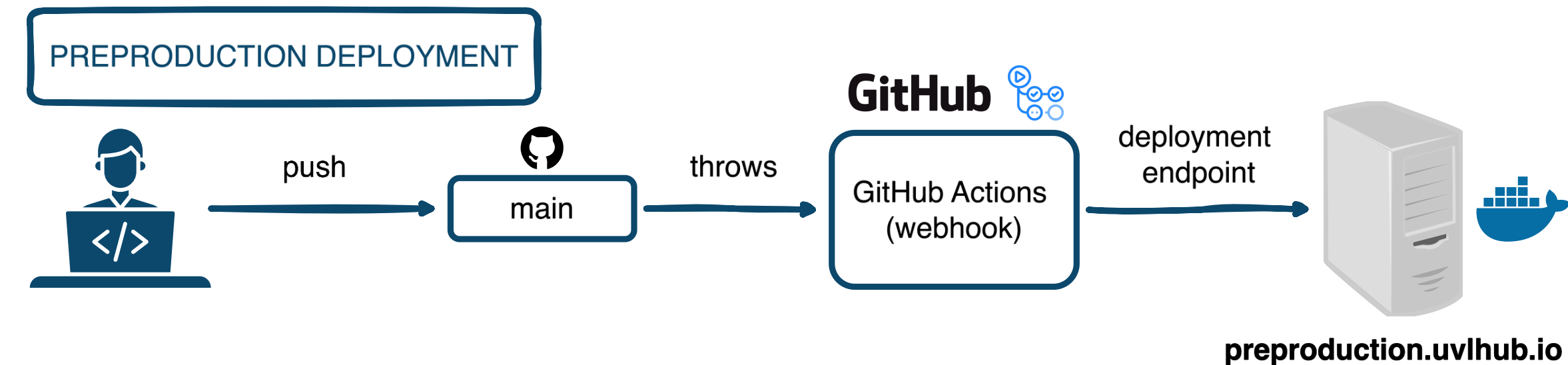
4. Despliegue Continuo (CD) en uvlhub.io

Despliegue manual



4. Despliegue Continuo (CD) en uvlhub.io

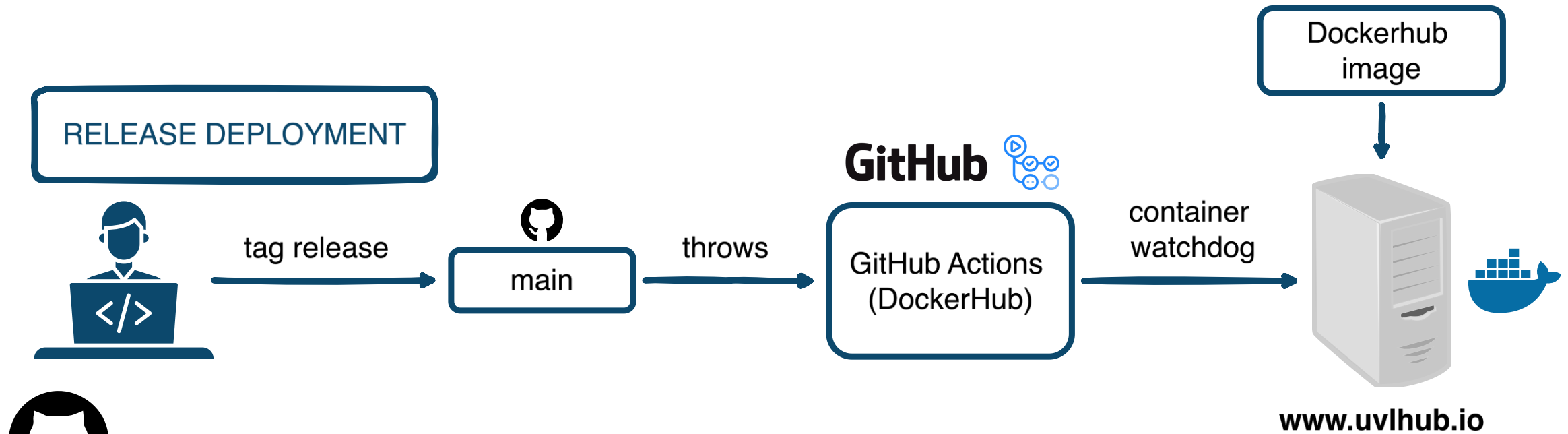
Despliegue en pre-producción (webhook)



[Ver código de ejemplo](#)

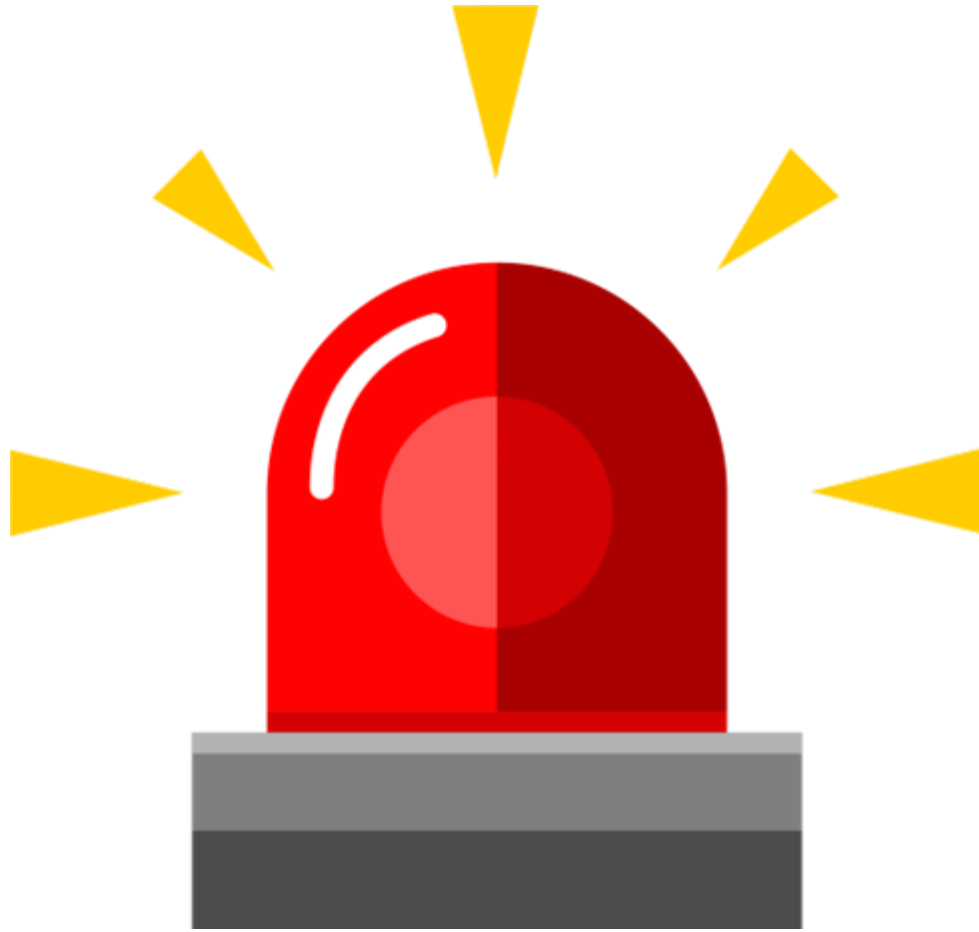
4. Despliegue Continuo (CD) en uvlhub.io

Despliegue en producción (Dockerhub)



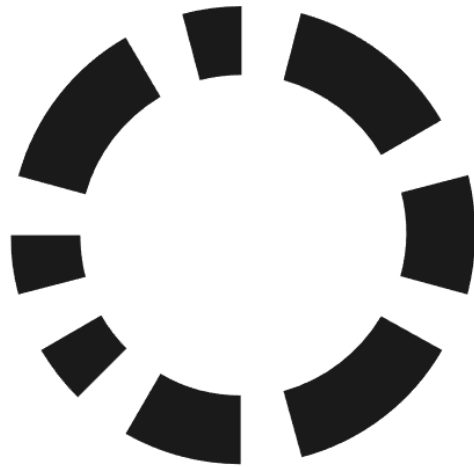
[Ver código de ejemplo](#)

1. Introducción a CI/CD
2. GitHub Actions
3. Integración Continua (CI) en uvlhub.io
4. Despliegue Continuo (CD) en uvlhub.io
5. **Ejercicio práctico CI: Codacy**
6. Ejercicio práctico CD: Render
7. Y yo, ¿qué puedo hacer en el proyecto?



Antes de continuar, borre **TODOS** los archivos **.yml**
de ***.github/workflows***

5. Ejercicio práctico CI: Codacy

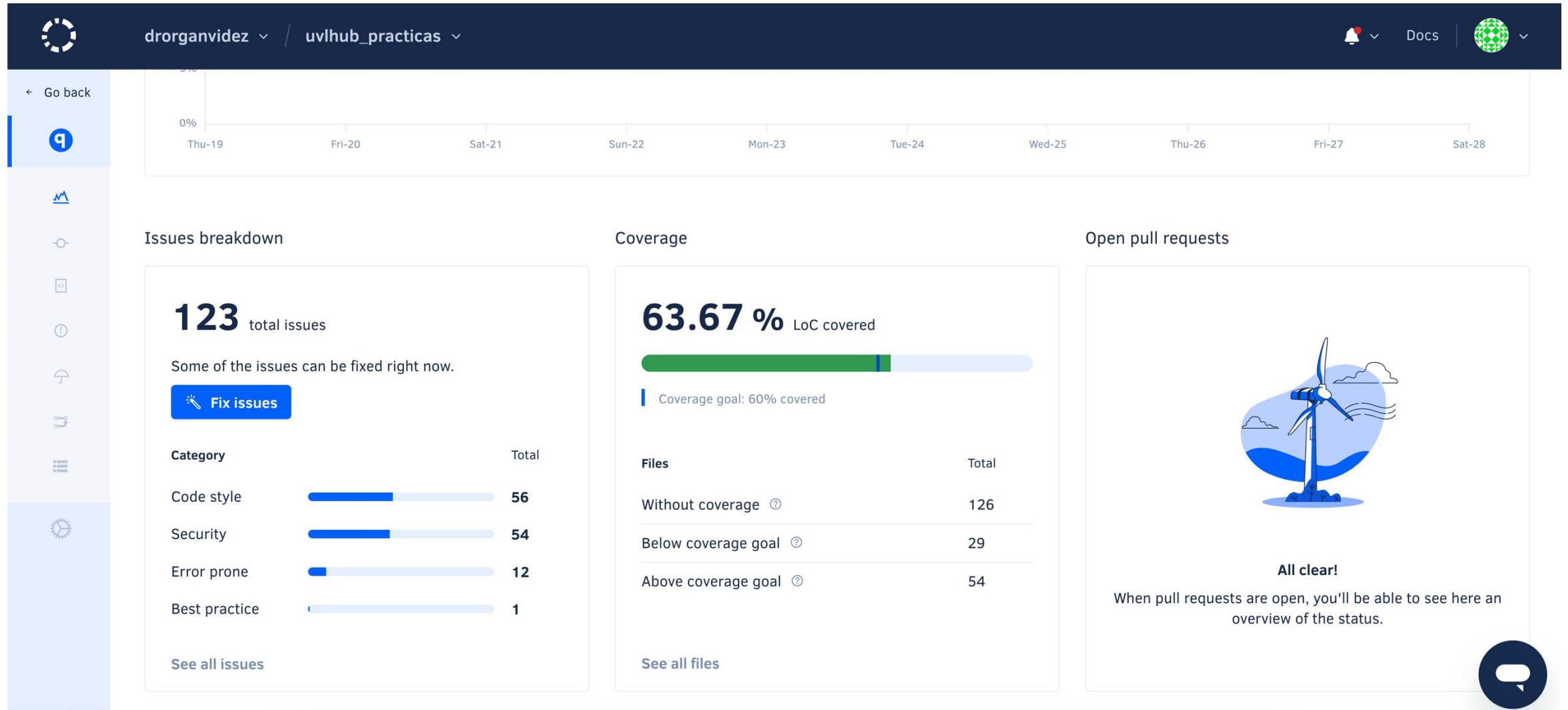
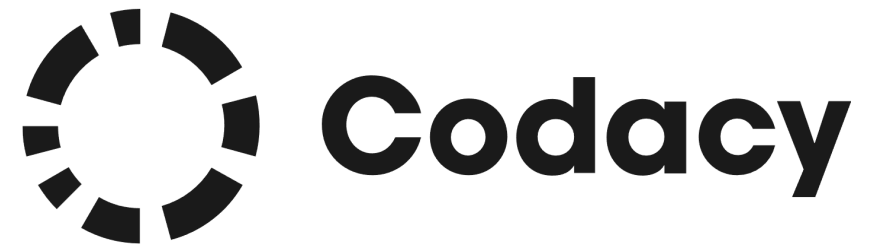


Codacy

herramienta automatizada para analizar la
calidad del código

codacy.com

5. Ejercicio práctico CI: Codacy



5. Ejercicio práctico CI: Codacy



docs.uvlhub.io/tutorials/codacy

Docs / Tutorials / CI: Codacy tutorial



5. Ejercicio práctico CI: Codacy

Modificaciones propuestas



- > **1.-** Añadir un paso que verifique si hay nuevas versiones de las dependencias instaladas
- > **2.-** Verificar la compatibilidad del código con las versiones 3.10, 3.11 y 3.12 de Python
- > **3.-** Realizar un análisis de seguridad de las dependencias instaladas con la librería *pip-audit*

1. Introducción a CI/CD
 2. GitHub Actions
 3. Integración Continua (CI) en uvlhub.io
 4. Despliegue Continuo (CD) en uvlhub.io
 5. Ejercicio práctico CI: Codacy
 - 6. Ejercicio práctico CD: Render**
 7. Y yo, ¿qué puedo hacer en el proyecto?
- 

6. Ejercicio práctico CD: Render



plataforma de **despliegue**
de aplicaciones web

render.com



filess.io

hosting de **bases**
de datos

filess.io

6. Ejercicio práctico CD: Render



uvlhub / Production / uvlhub_drorganvidez

WEB SERVICE

uvlhub_drorganvidez

Docker

Free

Upgrade your instance

Connect

Manual Deploy

drorganvidez / uvlhub_practicas

main

https://uvlhub-drorganvidez.onrender.com

- Events
- Logs
- Disks
- Environment
- Shell
- Previews
- Jobs
- Metrics
- Scaling
- Settings

Your free instance will spin down with inactivity, which can delay requests by 50 seconds or more.

Upgrade now

✓	Deploy live for 687ef1d: feat: Addd Render CD September 24, 2024 at 9:15 PM	
⬆	Deploy started for 687ef1d: feat: Addd Render CD Triggered via Deploy Hook September 24, 2024 at 9:14 PM	
✓	Deploy live for 687ef1d: feat: Addd Render CD September 24, 2024 at 9:01 PM	Rollback
⬆	Deploy started for 687ef1d: feat: Addd Render CD Manually triggered by you via Dashboard	

6. Ejercicio práctico CD: Render



filess.io

filess.io



☆ Smart Query

Databases

Standard Tier

Billing

<> API

✓ Status



Get more?

Get more storage space and more features.

Upgrade Now



MariaDB



MariaDB

v10.7.4

Status

Up



Web Client



Code

Size

0.58 MB / 10 MB



Request Backup



Delete Uvhub



Backups

uvlhub

Host

mnr.h.filess.io

Database

uvlhub_signalboy

User

uvlhub_signalboy

Port

3305

Do not drop this database. It won't be included in the backup if it does not exist.

Password

.....

Connection formats

MariaDB URI

mariadb://uvlhub_signalboy:.....@mnr.h.filess.io:3305/uvlhub_signalboy

MariaDB login command

mariadb -u uvlhub_signalboy -P 3305 -p..... -h mnr.h.filess.io uvlhub_signalboy



6. Ejercicio práctico CD: Render

1ª parte



docs.uvlhub.io/deployment/render

Docs / Deployment / Render



6. Ejercicio práctico CD: Render

2ª parte



docs.uvlhub.io/tutorials/render_tutorial



Docs / Tutorials / CD: Render tutorial

6. Ejercicio práctico CD: Render

Modificaciones propuestas



- > **1.-** Permitir que se realice el despliegue solo si un *pull request* ha sido aprobado y mergeado
- > **2.-** Permitir que se realice el despliegue solo si todas las pruebas han pasado
- > **3.-** Realizar un despliegue solo cuando se crea un *tag* en GitHub

1. **Introducción a CI/CD**
 2. **GitHub Actions**
 3. **Integración Continua (CI) en uvlhub.io**
 4. **Despliegue Continuo (CD) en uvlhub.io**
 5. **Ejercicio práctico CI: Codacy**
 6. **Ejercicio práctico CD: Render**
 7. **Y yo, ¿qué puedo hacer en el proyecto?**
- 

7. Y yo, ¿qué puedo hacer en el proyecto?

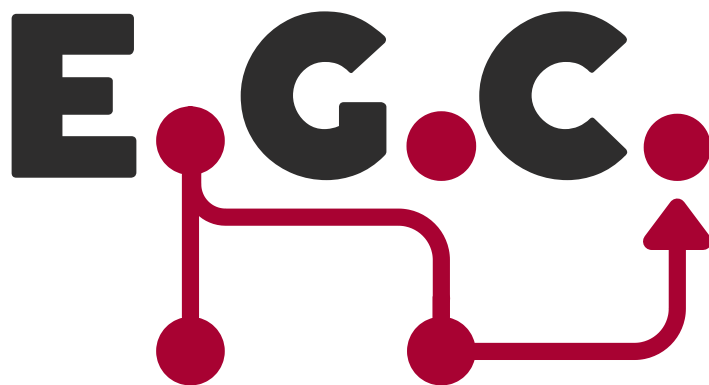


**¡Diseña tu propio
pipeline de CI / CD!**

Integrar workflows
actuales (**¡configurarlos!**)

Diseñar **otros** workflows
y/o **extender** los actuales

Usar otras herramientas
de CI / CD no vistas



Grado en Ingeniería Informática – Ingeniería del Software

Evolución y Gestión de la Configuración



Escuela Técnica Superior de
Ingeniería Informática

¡Gracias!

“Cuando alguien afirma que algo no puede hacerse, generalmente están equivocados.”

- Isaac Asimov