

# GITHUB ACTIONS — CONCEPTOS EN UNA HOJA

CI y CD se orquestan con workflows: eventos Git → jobs en runners. Esta hoja separa bien **integrar vs llevar código a un entorno**.

## CI VS CD (QUÉ SUELE ENTRAR EN ENTREVISTAS)

Sigla	Significa	Idea práctica
CI	Continuous Integration	Cada cambio integrado al repo debe <b>build + test</b> automatizado (rápido feedback en PR).
CD (Delivery)	Continuous Delivery	Trunk/main <b>siempre desplegable</b> ; el último clic hacia prod puede ser <b>manual</b> pero el pipeline ya es confiable.
CD (Deployment)	Continuous Deployment	Tras CI verde en <code>main</code> , el <b>deploy a producción corre solo</b> — solo con guardrails fuertes (tests, revisión previa por PR, flags, etc.).

En GitHub Actions el **CD** es otro workflow o **otro job** que corre **cuando CI pasó**, en otro branch/tag o con `workflow_run`, y usa credenciales hacia Kubernetes, ECS, `terraform apply`, función serverless, etc.

## PALABRAS QUE DEBES DOMINAR

Término	Qué es
Workflow	Archivo YAML en <code>.github/workflows/</code>
Event	Disparador: <code>push</code> , <code>pull_request</code> , <code>schedule</code> , <code>workflow_dispatch</code> , ...
Job	Grupo de <b>steps</b> que corren en el mismo runner
Step	Comando o acción ( <code>uses: actions/checkout</code> )
Runner	Máquina virtual donde corre el job (hosted o self-hosted)
Secret	Variable sensible ( <code>Settings → Secrets and variables</code> )
Artifact	Salida guardada entre jobs ( <code>actions/upload-artifact</code> ), p. ej. binario para desplegar
Environment	Nombre tipo <code>staging</code> / <code>production</code> con <b>secrets</b> propios y, si quieres, <b>approval</b> antes del deploy

## PATRÓN TÍPICO DE CI (INTEGRACIÓN)




- Checkout del código
- Instalar runtime (Node, Python, Go...)
- Lint / `npm test` / `pytest` / `terraform validate`
- Build (artefacto o imagen) y, si toca, **push** solo a registry con tag de ese commit

## PATRÓN TÍPICO DE CD (ENTREGA)

- Gatillo sólo cuando toca — ej. `push` a `main` con CI ya verde, `release` o job previo terminado bien
- Recuperar **artifact** o **imagen** del paso CI (mismo digest/tag inmutable)
- Autenticar contra el destino (OIDC a la nube, kubeconfig, API del proveedor)
- Deploy**: `kubectl apply`, `helm upgrade`, `aws ecs update-service`, `terraform apply` en workspace `prod`, etc.
- En **producción**: **Environment** con secretos distintos, **protección de rama** y, a menudo, **aprobación manual** o ventana de deploy

## BUENAS PRÁCTICAS RÁPIDAS

- ✓ CI en PR: fallar si tests o quality gate no pasan

-  **CD**: credenciales con **OIDC** o secrets de **Environment**, nunca hardcodedas en YAML
-  Jobs cortos y con **caché** de dependencias cuando duela el tiempo
-  Separar permisos: el job de deploy no necesita los mismos secretos que el de build de un fork






#### ★ COMUNIDAD SKOOL — SIGUIENTE PASO



### ¿LISTO PARA PRACTICARLO DE VERDAD?

Esta guía es el mapa; la diferencia suele estar en **hacer labs sin quedarte atrapado ante un error** durante tres días. En mi comunidad de Skool aprendes con **cursos** dentro de la plataforma, **sesiones en vivo** y **soporte técnico** cuando tu entorno no levanta o tu despliegue falla.

 **Entra en la comunidad — [skool.com/jemjaf](https://skool.com/jemjaf)** 

Qué hay dentro (según la oferta vigente en Skool):

-  **Cursos** grabados para ir a tu ritmo, más **lives** temáticos (cloud y DevOps)
-  **Sandboxes temporales en AWS, Azure y GCP**: practica en nube **real** sin el miedo a una factura sorpresa
-  **Soporte directo** para destrabar laboratorios, código y dudas puntuales
-  **Career Center** para revisión de CV y LinkedIn con mirada técnica
-  Materiales prácticos y guías que acompañan lo que enseño en clase

 **Únete en [skool.com/jemjaf](https://skool.com/jemjaf)**  — Es formación hands-on y acompañamiento; sin prometer empleos automáticos. El objetivo es que **lleves práctica defendible**.