

Master Cloud Apps

Despliegue continuo

Configuración de un pipeline de despliegue continuo

Se continua la práctica de CI integrando un workflow de despliegue continuo.

Asunciones

- Se crea un sistema de integración continua usando las herramientas de CI **GitHub Actions**. <https://github.com/franco87/fj.franco.2019-CI>
- El código del despliegue se almacena en este repositorio: <https://github.com/franco87/fj.franco.2019-CD>
- El artefacto de entrega es una imagen **Docker** que se almacena en el Registry **DockerHub**. <https://hub.docker.com/r/franco87/practica-ci>
- Todo el ejemplo es funcional y se puede acceder a la aplicación desplegada en <https://practica-cd.mca.ethcero.es>

Entorno de despliegue

Se ha creado un script de Terraform para desplegar un servidor en el cloud de Google donde se instala Microk8s como plataforma de Kuberentes.

El script de Terraform no está integrado dentro del proceso de despliegue continuo. Solamente se ha lanzado de manera manual para inicializar el entorno.

Se asume que se deben extraer las credenciales de Kubernetes para poder acceder al API de k8s ejecutando dentro del servidor el comando `microk8s.config`. estas credenciales se han configurado como **secreto** en el repositorio de CD.

Flujo de despliegue

El proceso de despliegue se inicia cuando se crea una nueva release en el repositorio principal de la aplicación (<https://github.com/franco87/fj.franco.2019-CI>).

El workflow `release-master.yaml` de CI, tras compilar y subir el artefacto a Dockerhub, ejecuta una llamada al API de GitHub creando un evento `dispatch` en el repositorio de CD adjuntando los datos de la nueva imagen generada en el proceso de construcción.

El workflow `deploy-release.yaml` de CD espera un evento `repository_dispatch` del tipo `deploy_application` y comienza el ciclo de despliegue.

NOTE

El API de microk8s no está expuesto a internet. Es necesario ejecutar un tunel ssh para poder interactuar con el API.

Comienza inicializando un tunel ssh entre el Runner de github y el servidor remoto para poder ejecutar llamadas al API de K8s. Previamente se habian introducido la clave pública usada para CD en el servidor de Google y la clave privada como secreto en el repositorio.

Se configura el fichero de configuración `kubeconfig` para poder ejecutar comando con `kubectl` contra el servidor.

Los siguientes pasos son el despliegue propiamente dicho y se dividen en 3 etapas:

- Se despliega la nueva imagen con los datos obtenidos del evento `repository_dispatch`
- Se monitoriza el rollout para verificar que la nueva imagen se ha levantado.
- Si el proceso de rollout falla, se ejecuta un rollback para volver a la versión anterior.