

# Práctica SDLC sobre IaC 2024/2025

## MCSI

### Table of Contents

Normas básicas.....	1
Práctica 2024/25 (puntuá sobre 10).....	1
Parte común: Crear un cluster kubernetes.....	1
Opción 1: FluxCD.....	3
Se debe entregar:.....	4
Opción 2: ArgoCD.....	5
Eliminar la práctica.....	5
Práctica 2023/24 (puntuá sobre 9).....	5

---

### Normas básicas

Este año habrá dos opciones:

1. Repetir la práctica del año pasado, que puntuará sobre 9.
2. Realizar la práctica de este año, que puntuará sobre 10.

Se trata de elegir una de las dos opciones. En ambos casos, 1-2 páginas deberían ser suficiente para obtener la puntuación más alta.

---

## Práctica 2024/25 (puntuación sobre 10)

La práctica consiste en replicar un CD sobre Kubernetes. Se puede optar por usar FluxCD o ArgoCD. No será necesario hacer esta práctica si se opta por la práctica del año pasado (2023/2024).

### Parte común: Crear un cluster kubernetes

1. Descargar kubectl → <https://kubernetes.io/docs/tasks/tools/>
2. Descargar kind → <https://kind.sigs.k8s.io/docs/user/quick-start/#installation>  
Yo personalmente prefiero descargarme el binario directamente de → <https://github.com/kubernetes-sigs/kind/releases>
3. Crear un archivo de configuración para Kind → <https://github.com/magmax/uclm-mcsi-acyds/blob/main/examples/kind.yaml>
4. Ejecutar Kind para crear un cluster con el archivo anterior:  

```
kind create cluster -config kind.yaml
```
5. Comprobar que el nodo está en estado “ready”:  

```
kubectl get nodes
```
6. Guardar evidencia de la creación del cluster y del estado del nodo para el trabajo final.

## Opción 1: FluxCD

No es necesario hacer esta parte si se opta por ArgoCD.

Existen otras formas válidas para instalar Flux, pero se describe aquí una de ellas:

1. Instalar Flux

```
kubectl apply -f
```

```
https://github.com/fluxcd/flux2/releases/latest/download/install.yaml
```

2. Esperar a que se instalen todas las aplicaciones:

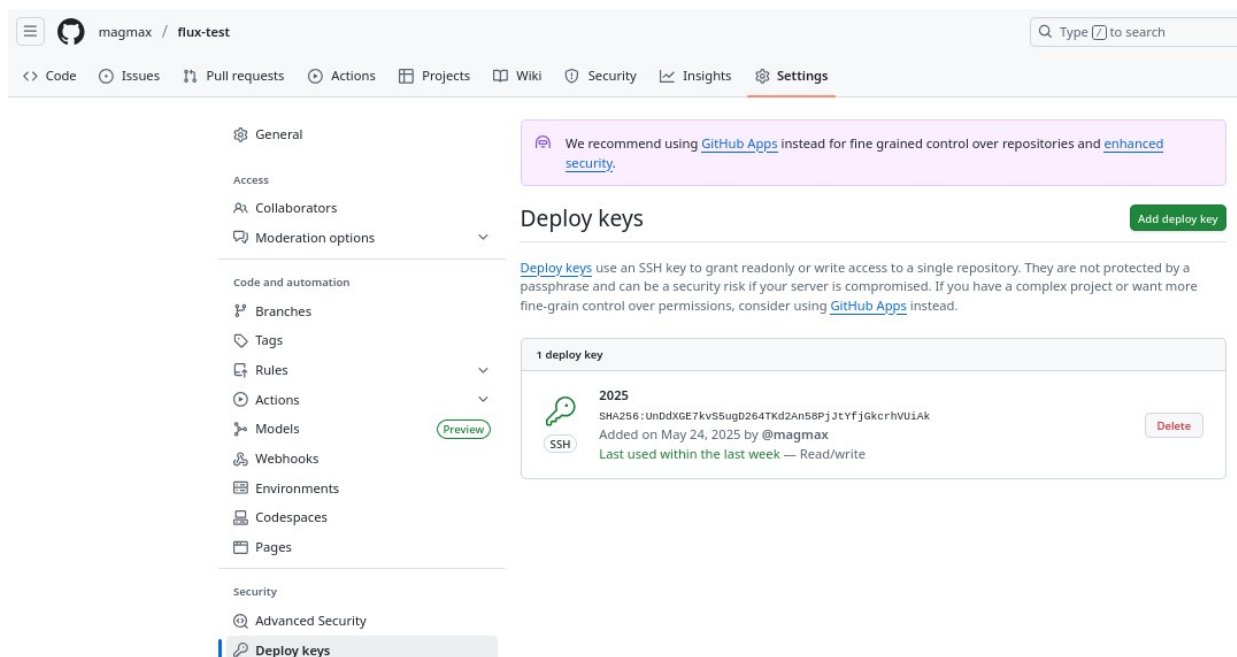
```
kubectl get pods -n flux-system
```

3. Generar un par de claves ssh sin contraseña (generará los archivos `key` y `key.pub` en el directorio en el que se ejecute):

```
ssh-keygen -f key -t ecdsa
```

4. Crear un repositorio público en GitHub, GitLab o algún sistema de control de versiones similar. El resto de la práctica se explicará con GitHub, pero el resto también son válidos.

5. Añadir la clave pública (archivo `key.pub`) como “Deployment keys” con permiso de escritura (en las “settings” del repositorio):



6. Configurar el repositorio remoto mediante la orden:

```
flux bootstrap git -url=<REPO> --branch=main --private-key-file=key -path=<PATH>
```

Donde “<REPO>” es la URL del repositorio y “<PATH>” es el directorio dentro del repositorio donde se encontrará el código que queréis desplegar. Es posible que este path deba existir para que la orden funcione correctamente. Ejemplo:

```
flux bootstrap git -url=ssh://git@github.com/magmax/flux-test.git --branch=main --private-key-file=key -path=clusters/my-cluster
```

7. Elegir el nombre del namespace que se utilizará para la práctica. En los ejemplos se usa el namespace: “example”, pero se sugiere cambiarlo por el nombre del alumno o algo que diferencie la práctica de la de los demás para evitar confusiones.
8. Mostrar el contenido del namespace (este comando puede fallar; mostrar el fallo):  
`kubectl get deploy,pod,svc,ingress,cm -n <NAMESPACE>`
9. Crear el Namespace y contenido dentro usando el repositorio configurado. Se puede utilizar esta plantilla → <https://github.com/magmax/flux-test/blob/main/clusters/my-cluster/example/example.yaml>
10. Probar que una modificación en el repositorio se refleja en el contenido del Namespace. Se sugiere cambiar el número de réplicas, que debería alterar el número de Pods.

### Qué se debe entregar:

- Evidencia de la creación del cluster y de que el nodo está “ready” (1 punto)
  - Evidencia de los servicios de flux corriendo en el namespace apropiado ( 1 punto)
  - Evidencia de que el namespace inicial está vacío o no existe (1 punto)
  - URL del repositorio público utilizado para la gestión de la configuración con fluxCD (hasta 2 puntos)
  - Evidencia de la creación del namespace y del contenido dentro, creados usando el repositorio configurado (hasta 2 puntos)
  - Evidencia de la alteración del contenido al modificarlo en el repositorio (hasta 2 puntos)
  - Explicar dónde se podrían establecer medidas de seguridad en el proceso (hasta 1 punto)
-

## Opción 2: ArgoCD

No es necesario hacer esta parte si se opta por FluxCD.

1. Crear el Namespace para ArgoCD:

```
kubectl create namespace argocd
```

2. Instalar ArgoCD:

```
kubectl apply -n argocd -f
```

```
https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml
```

3. Esperar a que todo esté instalado:

```
kubectl get pods -n argocd
```

4. Obtener la clave de administrador:

```
kubectl get secret -n argocd argocd-initial-admin-secret -o  
jsonpath="{.data.password}" | base64 -d -
```

5. En una ventana que dejaremos por ahí en segundo plano, ejecutar un port-forwarding para mapear un puerto local, como el 8080, al puerto 443 del servicio de ArgoCD:

```
kubectl port-forward svc/argocd-server -n argocd 8080:443
```

6. Acceder a la interfaz gráfica, probablemente en localhost:8080, usando el usuario “admin” y la contraseña obtenida en el paso 4.

7. Elegir el nombre del namespace que se utilizará para la práctica. En los ejemplos se usa el namespace: “example”, pero se sugiere cambiarlo por el nombre del alumno o algo que diferencie la práctica de la de los demás para evitar confusiones.

8. Crear un repositorio público en GitHub, GitLab o algún sistema de control de versiones similar. El resto de la práctica se explicará con GitHub, pero el resto también son válidos.

9. Copiar el contenido de este directorio a vuestro repositorio →

<https://github.com/argoproj/argocd-example-apps/tree/master/guestbook>

10. Crear una nueva Application. Este paso se puede hacer con la interfaz o en línea de órdenes, configurado para instalar lo que se encuentre en el repositorio creado en el paso 8.

La Application debe crear el namespace elegido en el paso 7 y desplegar algo dentro.

Aquí teneis una plantilla ->

<https://github.com/magmax/uclm-mcsi-acyds/blob/main/examples/app-guestbook.yaml>

Un detalle importante: aunque el contenido de la aplicación se despliegue en vuestro namespace, la Application se debe instalar en el Namespace “argocd”. Es decir: en el yaml que define la Application, el valor de `metadata.namespace` debe ser `argocd` y sólo tendréis que cambiar el contenido de `spec`.

11. Obtener evidencias del contenido del namespace:

```
kubectl get deploy,pod,svc,ingress,cm -n <NAMESPACE>
```

12. Realizar modificaciones en vuestro repositorio y observad cómo cambian en vuestro despliegue. [Se sugiere cambiar el número de réplicas](#), lo que alterará el número de PODs desplegados.

## Qué se debe entregar:

- Evidencia de la creación del cluster y de que el nodo está “ready” (1 punto)
  - Evidencia de los servicios de ArgoCD en ejecución y ready (1 punto)
  - Evidencia de que el namespace inicial está vacío o no existe (1 punto)
  - URL del repositorio público utilizado para la gestión de la configuración con ArgoCD (hasta 2 puntos)
  - Evidencia de la creación del namespace y del contenido dentro, creados usando el repositorio configurado, junto con imágenes de la Application (hasta 2 puntos)
  - Evidencia de la alteración del contenido al modificarlo en el repositorio (hasta 2 puntos)
  - Explicar dónde se podrían establecer medidas de seguridad en el proceso (hasta 1 punto)
-

## Eliminar la práctica

Para eliminar la práctica y todos los recursos creados bastará con ejecutar:

```
kind delete cluster
```

## Práctica 2023/24 (puntúa sobre 9)

Ésta es la segunda opción, realizar la práctica del año pasado, aunque puntuará menos que la opción nueva. No la hagáis si optáis por la práctica 2024/2025 (fluxcd/argocd)

1. <https://gist.github.com/magmax/c8915d28e9fd002f580f45da77af3386> (3 puntos)
2. <https://gist.github.com/magmax/5107e463a0b4b812f2b6732c519c26e7> (3 puntos)
3. Describir las medidas de seguridad a establecer para gestionar un CI/CD para los dos proyectos anteriores. (3 puntos)

1-2 páginas debería ser suficiente para obtener la nota más alta.