

Desafío 14 - Bootcamp Devops Engineer

Alumno: Ignacio Peretti

Objetivo:

El objetivo de este desafío es poner en práctica el despliegue del Helm Chart desarrollado en el desafío #12 utilizando ArgoCD.

Se debe automatizar la gestión del despliegue de la aplicación y su base de datos MongoDB, siguiendo los principios de GitOps.

Requisitos:

1. Desplegar ArgoCD en el entorno de Kubernetes:

- Si no está desplegado aún, sigue los pasos necesarios para instalar ArgoCD en Kubernetes (usando Minikube o un entorno similar).

2. Configurar ArgoCD:

- Conectar ArgoCD a un repositorio Git que contenga el Helm Chart desarrollado en el desafío #12.
- Configurar una aplicación en ArgoCD para gestionar el despliegue de dicho Helm Chart.

3. Automatización de despliegues:

- Implementar la sincronización automática de ArgoCD para que los cambios realizados en el repositorio Git se reflejen automáticamente en el clúster.

4. Verificación del despliegue:

- Validar que el Helm Chart se despliega correctamente utilizando ArgoCD y que la aplicación y MongoDB están

Guía paso a paso para desplegar nuestro Helm Chart utilizando ArgoCD, siguiendo los principios de GitOps.

Paso 1: Desplegar ArgoCD en Kubernetes

Si no tenemos ArgoCD instalado, podemos hacerlo con el siguiente comando:

```
kubectl create namespace argocd
```

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

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

Acceder a la interfaz de ArgoCD: Para acceder a la interfaz de usuario, debemos usar el siguiente comando para exponer el servicio:

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

Luego, nos dirigimos a <http://localhost:8080>.

Iniciar sesión en ArgoCD:

Obtenemos la contraseña del usuario **admin** para acceder a ArgoCD:

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

Paso 2: Configurar ArgoCD

Conectamos ArgoCD a nuestro repositorio Git:

En la interfaz de ArgoCD, vamos a "Settings" > "Repositories" y hacemos clic en "CONNECT REPO" para agregar nuestro repositorio que contiene el Helm Chart.

Via HTTPS

Tipo: Git.

URL: URL del nuestro repositorio Git.

Usuario y contraseña en caso de necesitar credenciales.

CONNECTSAVE AS CREDENTIALS TEMPLATECANCEL

Choose your connection method:
VIA SSH ▼




CONNECT REPO USING SSH

Name (mandatory for Helm)
desafio13-educacionit

Project
default

Repository URL
https://github.com/IgnacioPeretti/desafio12.git

SSH private key data

	TYPE	NAME	PROJECT	REPOSITORY	CONNECTION STATUS
	git		default	https://github.com/IgnacioPeretti/desafio12.git	 Successful 

Creamos una aplicación en ArgoCD:

Nos dirigimos a "Applications" y damos clic en "NEW APP".

Nombre de la aplicación: elegimos un nombre para tu aplicación.

Proyecto: default.

Repositorio: seleccionamos el repositorio que acabamos de conectar.

Ruta: especificamos la ruta donde se encuentra nuestro Helm Chart (por ejemplo, /educacionit-chart).

Cluster: <https://kubernetes.default.svc>.

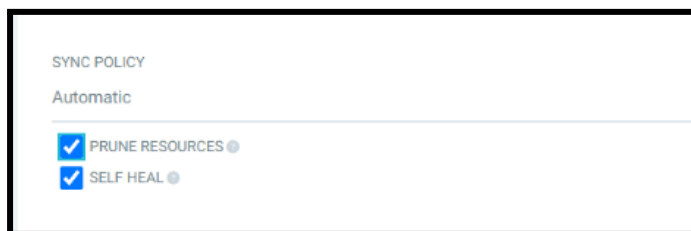
Namespace: elegimos el namespace donde deseamos desplegar la aplicación.

Paso 3: Automatización de despliegues

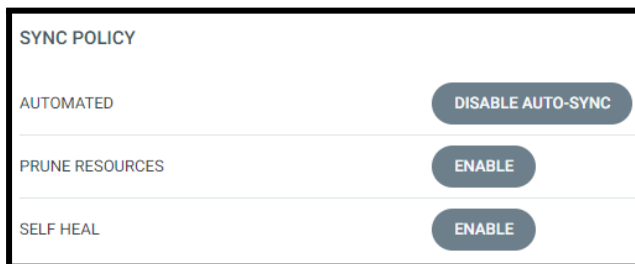
Habilitar la sincronización automática:

En la configuración de la aplicación en ArgoCD, buscamos la opción de Sync Policy y habilitamos la opción de sincronización automática (Auto-sync).

Esto asegurará que cualquier cambio en nuestro repositorio Git se refleje automáticamente en el clúster de Kubernetes.



Si ya tenemos una aplicación funcionando podemos activarla desde el apartado "Details"

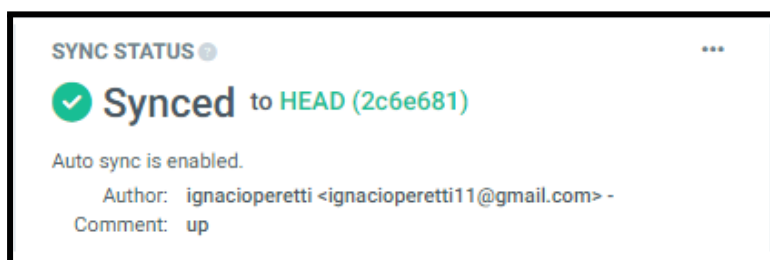


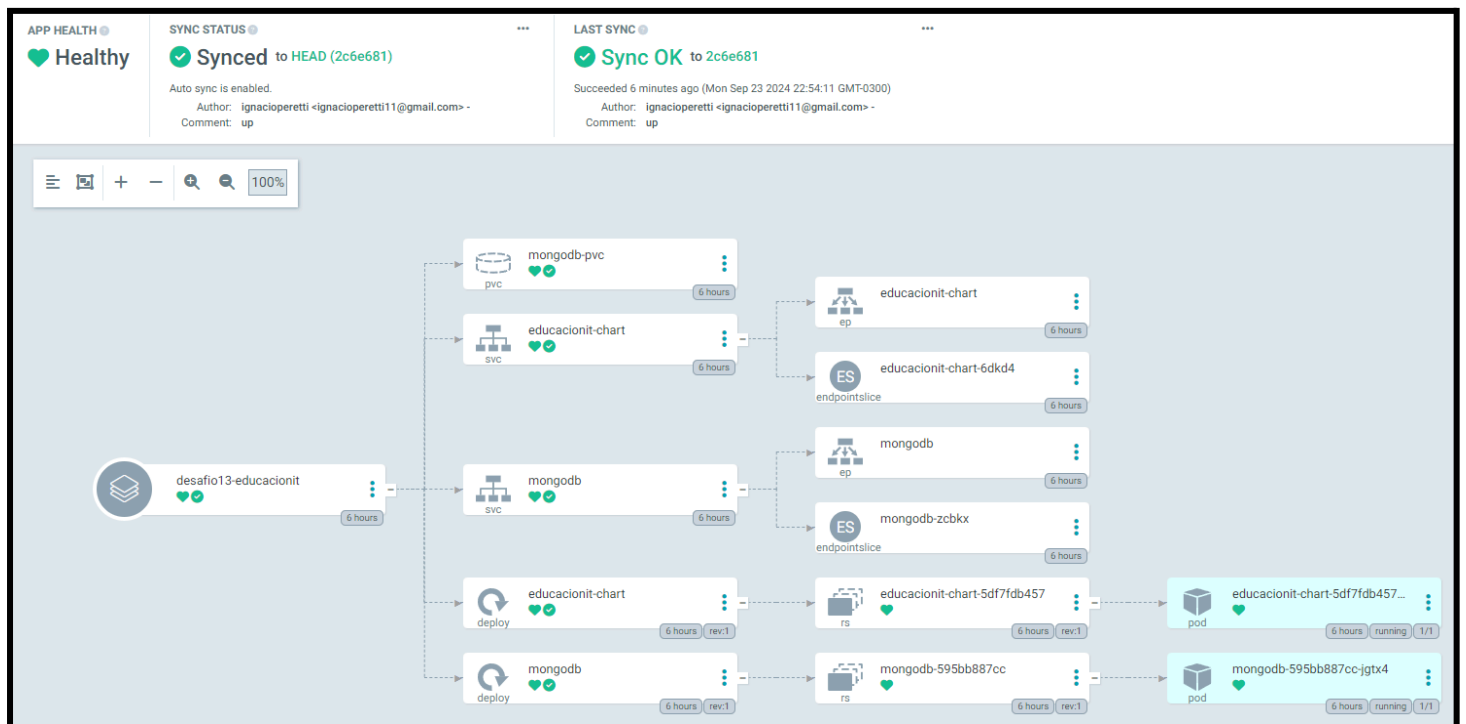
Paso 4: Verificación del despliegue

Verificar el estado de la aplicación:

Regresamos a la interfaz de ArgoCD y verificamos que nuestra aplicación esté en estado Synced y que no haya errores.

Podemos ver que el Auto Sync está habilitado.





Comprobar que los pods están corriendo:

`kubectl get pods -n <namespace_de_tu_aplicacion>`

```
PS C:\Users\CPU\Desktop\devops\devops-bootcamp\educacionitdevops\desafio12> kubectl get pods -n default
NAME                                READY   STATUS    RESTARTS   AGE
educacionit-chart-5df7fdb457-vqmcx  1/1     Running   0           5h32m
mongodb-595bb887cc-jgtx4            1/1     Running   0           5h32m
PS C:\Users\CPU\Desktop\devops\devops-bootcamp\educacionitdevops\desafio12>
```

En conclusión, para finalizar.

Con estos pasos, hemos configurado ArgoCD para automatizar el despliegue de nuestra aplicación y base de datos MongoDB utilizando el Helm Chart desarrollado previamente.

Cualquier cambio en el repositorio Git se reflejará automáticamente en nuestro clúster de Kubernetes, facilitando así la gestión y el despliegue continuo.

Fin de la guía.