Desafío 15

Nombre: Matias O. Riedel

Para este desafío empezaremos creando nuestros archivos de nuestra app, para eso crearemos los siguientes archivos:

Este archivo/manifiesto describe un Deployment de Kubernetes. Utiliza variables Helm ({{ .Values.\* }}) para permitir la personalización del despliegue. Gestiona las réplicas, la estrategia de actualización, los contenedores, y las variables de entorno.

Texto

Descripción generada automáticamente

Este archivo/manifiesto define un Service de Kubernetes. Exponen el deployment myapp a través de un LoadBalancer. Usa variables Helm para personalizar el tipo de servicio y puerto.

Captura de pantalla de un celular

Descripción generada automáticamente

Describimos un ConfigMap de Kubernetes. Proporciona variables de entorno configuradas (FLASK\_ENV) a los contenedores de myapp. Las variables pueden ser personalizadas a través del values.yaml.

Captura de pantalla de un celular

Descripción generada automáticamente

Defininimos la información del Chart de Helm. Incluye datos como el nombre del chart, descripción, versión del chart y versión de la aplicación.

Texto

Descripción generada automáticamente

Proporcionamos valores predeterminados o configurables para las plantillas de Helm. Con esto permitimos la personalizacion y reutilizar los recursos de Kubernetes con facilidad Texto

Descripción generada automáticamente

Definimos una aplicación de ArgoCD que implementa el patrón "App of Apps". Los sub-aplicaciones serán gestionadas desde aquí, con la sincronización automatizaciones.

Texto

Descripción generada automáticamente

Aplicamos la configuracion de ArgoCD para desplegar el Helm chart myapp. La sincronización se realiza automáticamente y se permite la auto-sanación de los recursos.

Texto

Descripción generada automáticamente

Definimos la aplicación de ArgoCD para desplegar los manifiestos Kubernetes sin Helm.

Texto

Descripción generada automáticamente

Luego de crear los archivos necesarios procedemos a crear nuestro entorno con kubectl con los comandos:

kubectl create namespace argocd

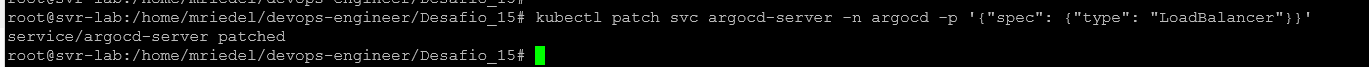
kubectl apply -n argocd -f https://raw.githubusercontent.com/argoproj/argo-cd/stable/manifests/install.yaml

Texto

Descripción generada automáticamente

Instalamos el CLI de nuestros ArgoCD con el comando de instalación para tener nuestro entorno de administración.

Luego de eso procedemos a configurar el servicio para exponer el servidor de ArgoCD



Procedemos a iniciar el servicio de nuestro servidor argocd con el comando:



Luego de iniciar procedemos a chequear que tengamos el pod de nuestro argocd activo para poder obtener los datos de acceso:



Verificaremos los datos de acceso a nuestro entorno con:



Procedemos a chequear nuestra conexión a nuestro entorno con el comando:   
  
argocd login <https://127.0.0.1:30080>

Problemas Y Soluciones:

Error al intertar instalar el chart de Helm donde recibí un error de failed to download. (Solución se verifico la correcta url del repositorio )

Error al sincronizar la aplicación de ArgoCD, por un tema de permisos no se pudo ejecutar correctamente. (Solución, ajustar permisos de los recursos utilizados por Kubernetes y cambie el nombre del namespace)