

Capacitación Kubernetes

Servicio de capacitación y entrenamiento ARIU

Clase 6

Despliegue de aplicación

Agenda

- Práctico Wordpress
- Kustomize
- Expedientes electrónicos integrados

- Archivos Yaml necesarios:
 - Base de datos
 - Declaración de PVC
 - Deploy con imagen de mysql oficial
 - Service de conexión con puerto 3306
 - Wordpress
 - Declaración de PVC
 - Deploy con imagen wordpress oficial
 - Service de conexión con puerto 80

Conexión Base de datos

- Variable de entorno con usuario y nombre de base
- Secret con contraseña

- Archivos Yaml necesarios:
 - Acceso externo con ingress

Límite en tamaño de archivos

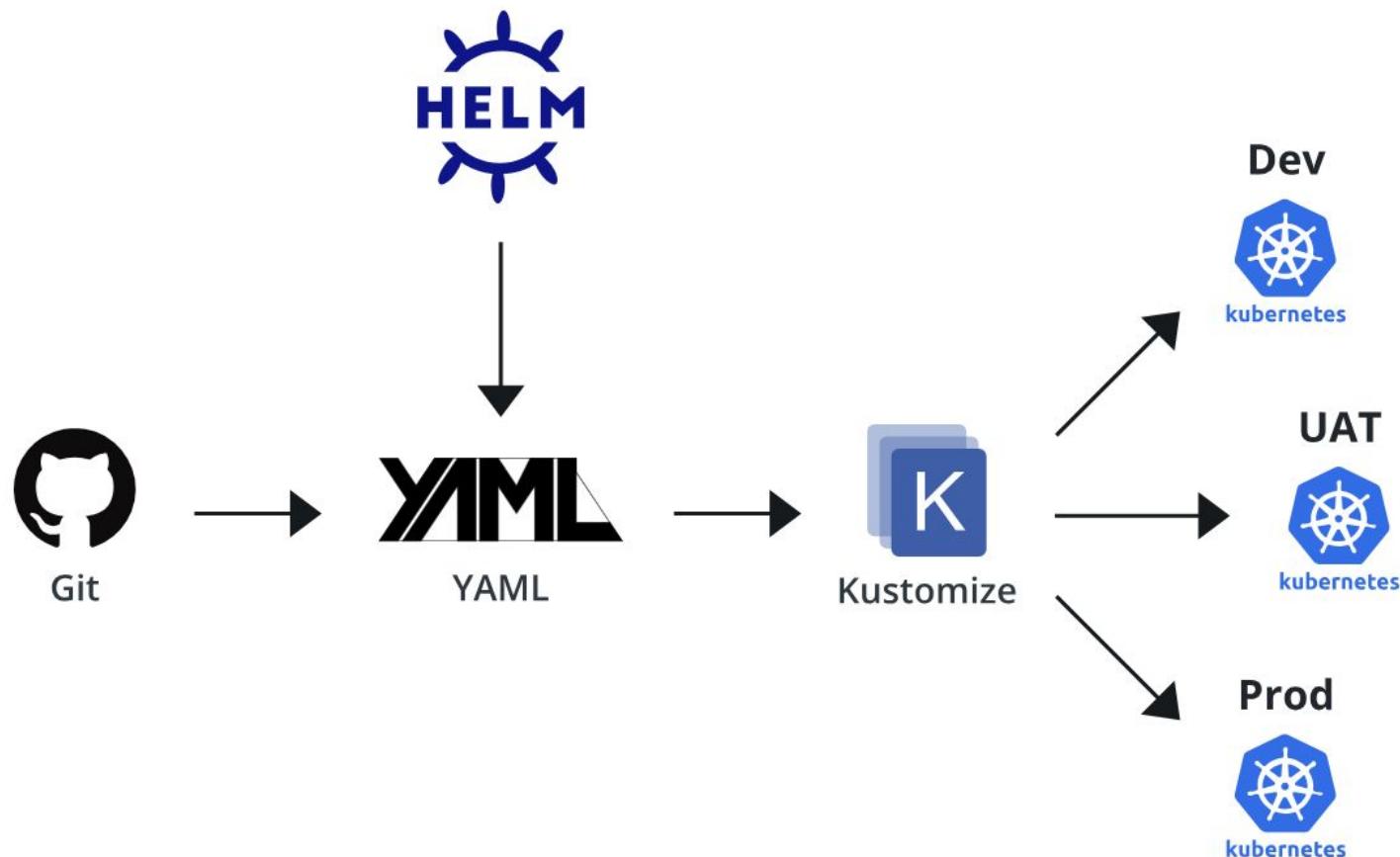
- La configuración dentro de Wordpress está en el archivo php.ini
- Dado que el php.ini se encuentra como parte de la imagen es necesario crear un configmap para reemplazarlo.
- Además hay que configurar mediante una anotación en el ingress (nginx) que maneja el límite de subida permitido.

- Revisar los yaml generados
- Crear namespaces
 - k create ns wordpress
- Generar configmap
 - kubectl create configmap phpini --from-file=php.ini
- Aplicar yaml
 - k apply -f .

Agenda

- Práctico Wordpress
- **Kustomize**
- Expedientes electrónicos integrados

- Kustomize es una herramienta que permite personalizar y gestionar recursos YAML de manera declarativa.
- Proporciona una forma estructurada y eficiente de definir y aplicar modificaciones a recursos sin modificar directamente los archivos de definición.
- Promueve la reutilización, la organización y la escalabilidad de las configuraciones.



- **Composición de Recursos:** Kustomize permite componer recursos a partir de bases (definiciones de recursos comunes) y aplicar ajustes específicos en los overlays.
- **Separación de Configuraciones:** Con kustomize, puedes separar la configuración de la lógica de la aplicación, lo que facilita la gestión y la reutilización de configuraciones.
- **Personalización Declarativa:** Kustomize se basa en declaraciones, lo que significa que puedes definir personalizaciones de manera declarativa en lugar de realizar ediciones manuales en los archivos YAML.

Un proyecto kustomize típico tiene la siguiente estructura:

- **Bases:** Carpeta que contiene definiciones de recursos comunes utilizados por múltiples aplicaciones.
- **Overlays:** Carpeta que contiene ajustes específicos para diferentes entornos o aplicaciones.
- **kustomization.yaml:** Archivo que define cómo kustomize ensambla los recursos y configura el comportamiento de personalización.

Una de las capacidades más poderosas de kustomize es la personalización de recursos de manera programática.

- Puedes cambiar imágenes de contenedor.
- añadir o modificar etiquetas.
- ajustar límites de recursos y más.

Todo ello a través de declaraciones en el archivo `kustomization.yaml`.

Functionality	Kustomize	Native Helm	Native Kubectl
Templating	No templating	Complex templating	No templating
Setup	No separate setup	Needs setup	No separate setup
Configuration	Manage multiple configurations with one base file	Manage multiple configurations with one base file	Should have separate files for each different configuration
Ease of Use	Easy learning curve	More difficult compared to the other two	Easy learning curve

Agenda

- Práctico Wordpress
- Kustomize
- Expedientes electrónicos integrados

- URL Kustomize
<https://www.densify.com/kubernetes-tools/kustomize/>
- Borrar todo lo anterior
 - k delete namespace eei-prueba
 - rm temp/*.yaml

Inicialización

- Hacer una copia de registry-secrets.env.dist y editar los valores necesarios:
 - cp registry-secrets.env.dist registry-secrets.env
 - nano registry-secrets.env
- Luego, ejecutando el script de inicialización se creará el namespace y los secretos para los registros de docker privados:
 - ./init-config.sh eei-prueba
- Probar
 - k ns eei-prueba
 - k get secrets
- Modificar INGRESS para que acepte snippet
 - k ns ingress-nginx
 - helm upgrade --install ingress-nginx ingress-nginx --repo https://kubernetes.github.io/ingress-nginx --namespace ingress-nginx --create-namespace --set controller.replicaCount=1,controller.service.loadBalancerIP=170.210.5.160 --set controller.allowSnippetAnnotations=true
 - k ns eei-prueba

Despliegue de una personalización

- Overlay "eei-prueba"
- Copiar overlay
 - cp -r overlays/siu-prueba overlays/eei-prueba
- Modificar dominios en archivos de overlay propio
 - sed -i "s/dominio\nciudad/eei\.k8s\.riu\.edu\.ar/g" overlays/eei-prueba/*/*
- Revisar kustomation.yaml principal del overlay propio y **modificar namespace**
 - nano overlays/eei-prueba/kustomization.yaml
 - ./secrets-config.sh eei-prueba
 - kubectl apply -k overlays/eei-prueba

Entrar a:

- eei.k8s.riu.edu.ar
- user: admin; pass: admin1234
- Cambiar en Arai-Usuarios correo de usuario `admin` por: admin@sudocu.edu.ar
- Crear cuenta de sudocu para el usuario `admin`. Cuenta: adminsudocu

Práctica para la casa

- Mirar video “Iceberg”
- Completar en sus cluster la instalación de Expedientes electrónicos Integrados



ARIU

Asociación Redes de Interconexión Universitaria