

Instalación de producto en entornos basados en k8s

Contenido

- 1. Introducción
 - 1.1 ¿Qué es Helm?
 - 1.2 ¿Qué nos aporta HELM?
- 2. Instalación
 - 2.2 Instalación local
- 3. Uso de Helm
 - 3.1 Instalación
 - 3.2 Marcha atrás
 - 3.3 Desinstalación
- 4. Consejos y recomendaciones
 - 4.1 Crear dependencia entre configMap y deployment
- 5. Recursos adicionales
 - 5.1 Plantilla básica de Ejemplo
 - Terminología

Resumen



- **Versión:** 1.0.0
- **Fecha publicación:** 5 ago 2020
- **Entrada en vigor desde:** 5 ago 2020

Histórico de cambios

Los cambios en la normativa vendrán acompañados de un registro de las modificaciones. De este modo se podrá realizar un seguimiento y consultar su evolución. Ordenándose de mas recientes a menos recientes, prestando especial cuidado a las cabeceras de la tablas dónde se indican las fechas de entrada en vigor y versión.

Versión	v01r00	Fecha publicación		Fecha entrada en vigor	
Alcance					
• Primera version de la guía • Establecer requisitos previos • Establecer terminologia basica • Ayudas a la instalacion • Comandos mas utilizados • Consejos • Accesos a recursos de ayuda					

1. Introducción

1.1 ¿Qué es Helm?

Helm es un gestor de paquetes basado en plantillas para Kubernetes y su principal función es definir, instalar y actualizar aplicaciones complejas de Kubernetes

1.2 ¿Qué nos aporta HELM?

- **Sencillez:** El procedimiento de instalación se simplifica, quedando en la ejecución de un comando "helm install".
- **Versionado:** En el desarrollo permite realizar iteraciones rápidas sobre cambios estructurales, añadir o quitar tipos de bases de datos, caches, servicios extra, configuraciones, etc...
- **Pilotable:** Como esta basado en plantillas, se puede personalizar y configurar para que la instalación tenga comportamientos diferentes según entorno.
- **Homegeneización:** Permite tener una forma unificada de instalación en el SAS, simplificando las instalaciones al equipo de Sistemas y replicación en los entornos de desarrollo.
- **Simplicidad:** La gestión de actualización, rollback y configuración se simplifica.
- **Heterogeneidad:** Permite realizar el despliegue en cualquier cluster de kubernetes sin tener que modificar la plantilla.

2. Instalación

Helm 3 es un cliente local, por lo que no será necesario realizar ninguna instalación en el cluster de destino, simplemente tendremos que descargar a local el CLI de Helm y cumplir una serie de requisitos.

2.1 Requisitos previos

- Acceso a un Cluster de Kubernetes de desarrollo
- Cli de HELM en local
- kubectl en local

2.2 Instalación local

La mejor forma de realizar la instalación es seguir la documentación oficial: <https://helm.sh/es/docs/intro/install/>

3. Uso de Helm

En los entornos propios del SAS el equipo de SISTEMAS se encargara de realizar las acciones necesarias con Helm, tales como instalaciones, actualizaciones, marchas atrás y desinstalaciones.

Para que podáis replicar los comando de los operadores en desarrollo, se indicaran aquí ejemplos de los comandos mas habituales.

Si se solicitara, estos comandos son los que deberían de ir incluidos en el PID.

3.1 Instalación

El comando habitual de instalación es "helm install ..." pero en el SAS por simplificar siempre intentaremos usar el siguiente comando para realizar las instalaciones:

Comando de Instalación
helm upgrade [release] [chartPath] --version [tag] -i -f [configPath] --description [description]

Esto comando intentara actualizar la release indicada desde la plantilla local que se indique, se le asignara la versión del tag, y gracias al la flag "-i" si esta release no existiera, realizaría una instalación. También se le indicaría el fichero de configuración, para los valores que debería de introducirse durante la instalación por el operador.

3.2 Marcha atrás

Para volver a una revisión anterior de la reléase se deberá de ejecutar el siguiente comando:

Comando de marcha atrás
helm rollback [release] [revision]

Hay que tener en cuenta que en Helm, las releases no solo tienen versiones, si no que también tienen revisiones, cada vez que se hace cualquier cambio en una release, se crea una nueva revisión, por ejemplo:

instalar la versión 1.0.0 por primera vez creara la primera revisión "1" si después hacemos una actualización solo de la configuración, la versión se mantendría pero generaría una nueva revisión "2". Esto hay que tenerlo en cuenta al realizar una marcha atras porque el comando tiene en cuenta las revisiones y no las versiones.

Para poder ver las revisiones podeis ejecutar el siguiente comando:



Leyenda:

- **[chartPath]:** Ruta local del Chart a instalar o actualizar
- **[configPath]:** Fichero de configuración (este es el fichero que rellenara y custodiara CTI)
- **[release]:** El nombre del Aplicativo
- **[revision]:** El numero de revisión
- **[description]:** Texto descriptivo de la operación que se pretende realizar, útil cuando se consulta el histórico

Comando para consultar el histórico

```
helm history[release]
```

Este comando mostrara una tabla en la que se podrá comprobar la información de cada una de las revisiones.

Ejemplo de comando de histórico

```
helm history angry-bird
REVISION    UPDATED              STATUS
CHART       APP VERSION         DESCRIPTION
1           Mon Oct 3 10:15:13 2016    superseded    alpine-
0.1.0       1.0                 Initial install
2           Mon Oct 3 10:15:13 2016    superseded    alpine-
0.2.0       2.0                 Upgraded successfully
3           Mon Oct 3 10:15:13 2016    superseded    alpine-
0.1.0       1.0                 Rolled back to 1
4           Mon Oct 3 10:15:13 2016    deployed      alpine-
0.1.0       1.0                 Upgraded config successfully
```

3.3 Desinstalación

Si se quiere realizar una desinstalación se puede usar el siguiente comando. Se recomienda la flag "--keep-history" por si se quisiera hacer marcha atrás de la desinstalación.

Comando para desinstalar

```
helm uninstall [release] --keep-history
```

4. Consejos y recomendaciones

4.1 Crear dependencia entre configMap y deployment

Helm detecta los cambios en los recursos que han cambiado y redespiega solo los necesarios. Pero por defecto no detecta si ha cambiado la configuración de un recurso que este en un configMap o una secret.

Para que Helm entienda que tiene que realizar un redespigue de un recurso que depende de otro, lo mas facil es incluir una anotacion que apunte al hash de el fichero del que depende.

Por ejemplo:

Ejemplo de anotación

```
kind: Deployment
spec:
  template:
    metadata:
      annotations:
        checksum/config: {{ include (print $.Template.BasePath "/configmap.yaml") . | sha256sum }}
    ...
```

En este ejemplo de Deployment, estamos incluyendo una nueva anotación que contiene el checksum o hash del fichero del que depende este deployment, de esta manera cada vez que la configuración cambien, la anotación también, y esto provocara que Helm detecte el cambio del deployment y lo redespigue.

5. Recursos adicionales

En el SAS disponemos de una pequeña colección de recursos útiles para el desarrollo.

5.1 Plantilla básica de Ejemplo

Se ha creado una plantilla sencilla para los proyectos que no tengan mucha complejidad en sus servicios, donde con mínimas adaptaciones y configuración se podría tener lista para entregar.

La plantilla esta alojada en el repositorio de código del SAS: <http://git.sas.junta-andalucia.es/gobernanza/ArchivosDeDesarrollo/basic-helm>

Si no se tuvieran permisos para acceder, solicitar a la OCA que incluyan a vuestro usuario en el grupo "[team-publico](#)" de Gitlab

En el [Readme.md](#) de este repositorio se amplia la documentación de como debe de ser usada.

Terminología

- **Release:** Una [Release] es una instancia de un Chart que se ejecuta en un clúster de Kubernetes. A menudo, un Chart se puede instalar muchas veces en el mismo clúster. Y cada vez que se instala, se crea un nuevo release. Considere un Chart MySQL. Si desea que se ejecuten dos bases de datos en su clúster, puede instalar ese Chart dos veces. Cada uno tendrá su propio release, que a su vez tendrá su propio nombre de release.
- **Chart:** La plantilla Helm es el conjunto específico de plantillas y otros recursos que se usa para instalar el producto.