Despliegues de microservicios en k8s con Helm

Presentado: Alejandro Morales Gracia

Email: ale95mogra@gmail.com

Repo proyecto: MoralG/Proyecto_Helm

Linkedin/alejandro-morales-gracia

¿Que es Helm?

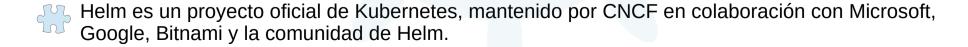
 Administrador de paquetes, llamados charts, utilizado para desplegar escenarios en Kubernetes.

- Instalar automáticamente dependencias de software.
- Interactuar con repositorios de chart, ya sean públicos o privados.
- Empaquetación de aplicaciones.
- Gestionar el ciclo de vida de los despliegues de los charts.
- Proceso de "rollback".
- Facilidad para los desarrolladores.

¿Que es Helm?

Administrador de paquetes, llamados **charts**, utilizado para desplegar escenarios con Kubernetes.

- Instalar automáticamente dependencias de software.
- Interactuar con repositorios de chart, ya sean públicos o privados.
- Empaquetación de aplicaciones.
- Gestionar el ciclo de vida de los despliegues de los charts.
- Proceso de "rollback".
- Facilidad para los desarrolladores.



Instalación de Helm v3

 Descargar la última versión del binario del repositorio oficial de Helm.



- Descomprimimos el fichero.
- Movemos el directorio a la dirección

usr/local/bin/helm

Instalación de Helm v3

 Descargar la última versión del binario del repositorio oficial de Helm.



- Descomprimimos el fichero.
- Movemos el directorio a la dirección

usr/local/bin/helm

Script ejecutable para la instalación de Helm.



Instalación de Helm v3

 Descargar la última versión del binario del repositorio oficial de Helm.



- Descomprimimos el fichero.
- Movemos el directorio a la dirección

usr/local/bin/helm

Script ejecutable para la instalación de Helm.



Para mayor seguridad es recomendable quitar los permisos de lectura a los grupos para el fichero de configuración del cluster. Ejecutamos el siguiente comando:

¿Que podemos hacer con Helm?

- Instalar charts y desplegar aplicaciones.
- Realizar una actualización de charts
- Eliminar los charts.
- Restablecer una release concreta. (rollback)
- Utilización de repositorios.
 - Oficiales.
 - Creados y gestionados por cuenta propia.

Añadir repositorios de charts

Añadir repositorio de chart:

helm repo add [NAME] [URL] [flags]

Ejemplo:

helm repo add stable https://charts.helm.sh/stable

Buscar un chart

helm search repo mysql

Añadir repositorios de charts

Añadir repositorio de chart:

helm repo add [NAME] [URL] [flags]

Ejemplo:

helm repo add stable https://charts.helm.sh/stable

Buscar un chart

helm search repo mysql

NAME	CHART VERSION	APP VERSION	DESCRIPTION
stable/mysql	1.6.3	5.7.28	Fast, reliable, scalable, and easy to use open
stable/mysqldump	2.6.0	2.4.1	A Helm chart to help backup MySQL databases usi
stable/prometheus-mysql-exporter	0.5.2	v0.11.0	A Helm chart for prometheus mysql exporter with
stable/percona	1.2.1	5.7.26	free, fully compatible, enhanced, open source d
stable/percona-xtradb-cluster	1.0.3	5.7.19	free, fully compatible, enhanced, open source d
stable/phpmyadmin	4.3.5	5.0.1	DEPRECATED phpMyAdmin is an mysql administratio
stable/gcloud-sqlproxy	0.6.1	1.11	DEPRECATED Google Cloud SQL Proxy
stable/mariadb	7.3.14	10.3.22	DEPRECATED Fast, reliable, scalable, and easy t

Instalación de charts

Instalando un chart:

helm install [NAME] [CHART] [flags]

Ejemplo:

helm install maria stable/mysql

NAME: maria

LAST DEPLOYED: Thu Apr 23 17:09:13 2020

NAMESPACE: default STATUS: deployed REVISION: 1

NOTES:

MySQL can be accessed via port 3306 on the following DNS name from within your

cluster:

maria-mysql.default.svc.cluster.local

To get your root password run:

MYSQL_ROOT_PASSWORD=\$(kubectl get secret --namespace default maria-mysql -o jsonpath="{.data.mysql-root-password}" | base64 --decode; echo)

To connect to your database:

1. Run an Ubuntu pod that you can use as a client:

kubectl run -i --tty ubuntu --image=ubuntu:16.04 --restart=Never -- bash -il

• • •

Instalación de charts

Instalando un chart:

helm install [NAME] [CHART] [flags]

Ejemplo:

helm install maria stable/mysql

Listar charts instalados

helm list

NAME: maria

LAST DEPLOYED: Thu Apr 23 17:09:13 2020

NAMESPACE: default STATUS: deployed REVISION: 1

NOTES:

 $\ensuremath{\mathsf{MySQL}}$ can be accessed via port 3306 on the following DNS name from within your

cluster:

maria-mysql.default.svc.cluster.local

To get your root password run:

MYSQL_ROOT_PASSWORD=\$(kubectl get secret --namespace default maria-mysql -o jsonpath="{.data.mysql-root-password}" | base64 --decode; echo)

To connect to your database:

1. Run an Ubuntu pod that you can use as a client:

kubectl run -i --tty ubuntu --image=ubuntu:16.04 --restart=Never -- bash -il

• • •

NAME NAMESPACE REVISION UPDATED STATUS CHART APP VERSION maria default 1 2020-12-14 13:39:0... deployed mysql-1.6.9 5.7.30

Actualización de charts

Actualizar un chart:

```
helm upgrade [RELEASE] [CHART] [flags]
```

Ejemplo:

helm upgrade -f values.yaml maria stable/mariadb



Actualización de charts

Actualizar un chart:

```
helm upgrade [RELEASE] [CHART] [flags]
```

Ejemplo:

helm upgrade -f values.yaml maria stable/mariadb

mariadbUser: usuario1

Ver los valores agregados

helm get values maria

USER-SUPPLIED VALUES: mariadbUser: usuario1

Actualización de charts

Actualizar un chart:

```
helm upgrade [RELEASE] [CHART] [flags]
```

Ejemplo:

helm upgrade -f values.yaml maria stable/mariadb

mariadbUser: usuario1

Ver los valores agregados

helm get values maria

USER-SUPPLIED VALUES: mariadbUser: usuario1

Listar charts instalados

helm list

REVISION	UPDATED	STATUS	CHART	APP VERSION	DESCRIPTION
1	Mon Dec 14 13:39:0	superseded	mysql-1.6.9	5.7.30	Install complete
2	Mon Dec 14 13:41:4	deployed	mariadb-7.3.14	10.3.22	Upgrade complete

Desinstalación de charts

• Desisnstalar un chart:

```
helm uninstall RELEASE_NAME [...] [flags]
```

Ejemplo:

helm uninstall maria --keep-history



Desinstalación de charts

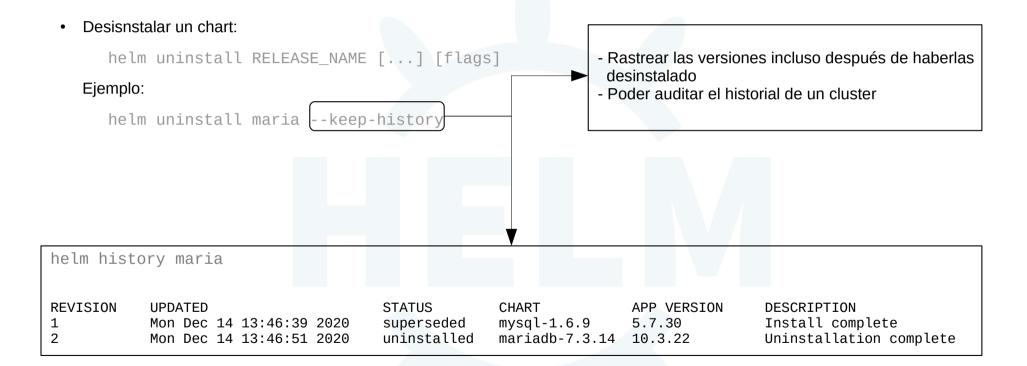
• Desisnstalar un chart:

helm uninstall RELEASE_NAME [...] [flags]

Ejemplo:
helm uninstall maria --keep-history

- Rastrear las versiones incluso después de haberlas desinstalado
- Poder auditar el historial de un cluster

Desinstalación de charts



Rollback de charts

• Realizar un rollback de un chart:

```
helm rollback <RELEASE> [REVISION] [flags]
```

Ejemplo:

helm rollback maria 1



Rollback de charts

Realizar un rollback de un chart:

```
helm rollback <RELEASE> [REVISION] [flags]

Ejemplo:
helm rollback maria 1
```

helm history maria

REVISION UPDATED STATUS CHART APP VERSION
1 Mon Dec 14 13:46:39 2020 superseded mysql-1.6.9 5.7.30
2 Mon Dec 14 13:46:51 2020 uninstalled mariadb-7.3.14 10.3.22

DESCRIPTION
Install complete
Uninstallation complete

Rollback de charts

Realizar un rollback de un chart:

```
helm rollback <RELEASE> [REVISION] [flags]
```

Ejemplo:

DEV/TSTON

helm rollback maria

helm history maria

INCATOTOM	OLDVIED	
1	Mon Dec 14 13	3:46:39 202
2	Mon Dec 14 13	3:46:51 202

HDDATED

Mon Dec 14 13:46:51 2020

STATUS superseded uninstalled CHART mysql-1.6.9 mariadb-7.3.14

APP VERSION 5.7.30 10.3.22

DESCRIPTION Install complete Uninstallation complete

Rollback was a success! Happy Helming!

helm history maria

REVISION	UPDATED	STATUS	CHART	APP VERSION	DESCRIPTION
1	Mon Dec 14 13:46:39 2020	superseded	mysql-1.6.9	5.7.30	Install complete
2	Mon Dec 14 13:46:51 2020	uninstalled	mariadb-7.3.14	10.3.22	Uninstallation complete
3	Mon Dec 14 13:49:02 2020	deployed	mysql-1.6.9	5.7.30	Rollback to 1

Creación de charts

Helm trabaja con un lenguaje de plantillas Go

Ejemplo:

```
apiVersion: v1
kind: ConfigMap
metadata:
   name: {{ .Release.Name }}-configmap
data:
   myvalue: "Hola Mundo"
   drink: {{ quote .Values.favorite.drink }}
food: {{ quote .Values.favorite.food }}
```

- Estructura: functionName arg1 arg2 ...
- Código Go entre {{ }}
- Multitud de funciones helm/functionGo

Creación de charts

Helm trabaja con un lenguaje de plantillas Go

Ejemplo:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hola Mundo"
  drink: {{ quote .Values.favorite.drink }}
food: {{ quote .Values.favorite.food }}
```

- Estructura: functionName arg1 arg2 ...
- Código Go entre {{ }}
- Multitud de funciones helm/functionGo

Para crear chart:

helm create NameChart

Creación de charts

Helm trabaja con un lenguaje de plantillas Go

Ejemplo:

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hola Mundo"
  drink: {{ quote .Values.favorite.drink }}
food: {{ quote .Values.favorite.food }}
```

- Estructura: functionName arg1 arg2 ...
- Código Go entre {{ }}
- Multitud de funciones helm/functionGo

Para crear chart:

helm create NameChart

```
app-crud/
    charts
    Chart.yaml
    templates
       deployment.yaml
       _helpers.tpl
       ingress.yaml
       NOTES, txt
       service.yaml
        tests
        test-connection.yaml
    values.vaml
3 directories, 8 files
```

Fichero Chart.yaml

Ejemplo:

Contiene:

Metadatos del chart.

Dependencias

```
apiVersion: v1
appVersion: "1.0.0"
description: Aplicacion CRUD en Helm chart con express.js y mongodb.
name: app-crud
version: 0.1.0
sources:
- https://github.com/MoralG/appCRUD
maintainers:
- name: moralq
  email: ale95mogra@gmail.com
dependencies:
- name: mongodb
version: latest
```

repository: https://kubernetes-charts.storage.googleapis.com/

Fichero values.yaml

Ejemplo:

 Contiene los valores que señalamos en las plantillas con el atributo. Values

```
initContainerImage: "alpine:3.6"
imagePullSecrets:
replicaCount: 1
image:
  repository: jainishshah17/express-mongo-crud
  taq: 1.0.1
  pullPolicy: IfNotPresent
mongodb:
  enabled: true
  image:
    taq: 3.6.3
    pullPolicy: IfNotPresent
  mongodbExtraFlags:
  mongodbUsername: admin
  mongodbDatabase: test
service:
  name: http
  type: LoadBalancer
  internalPort: 3000
  externalPort: 80
```

Despliegues de microservicios en k8s con Helm

Muchas Gracias