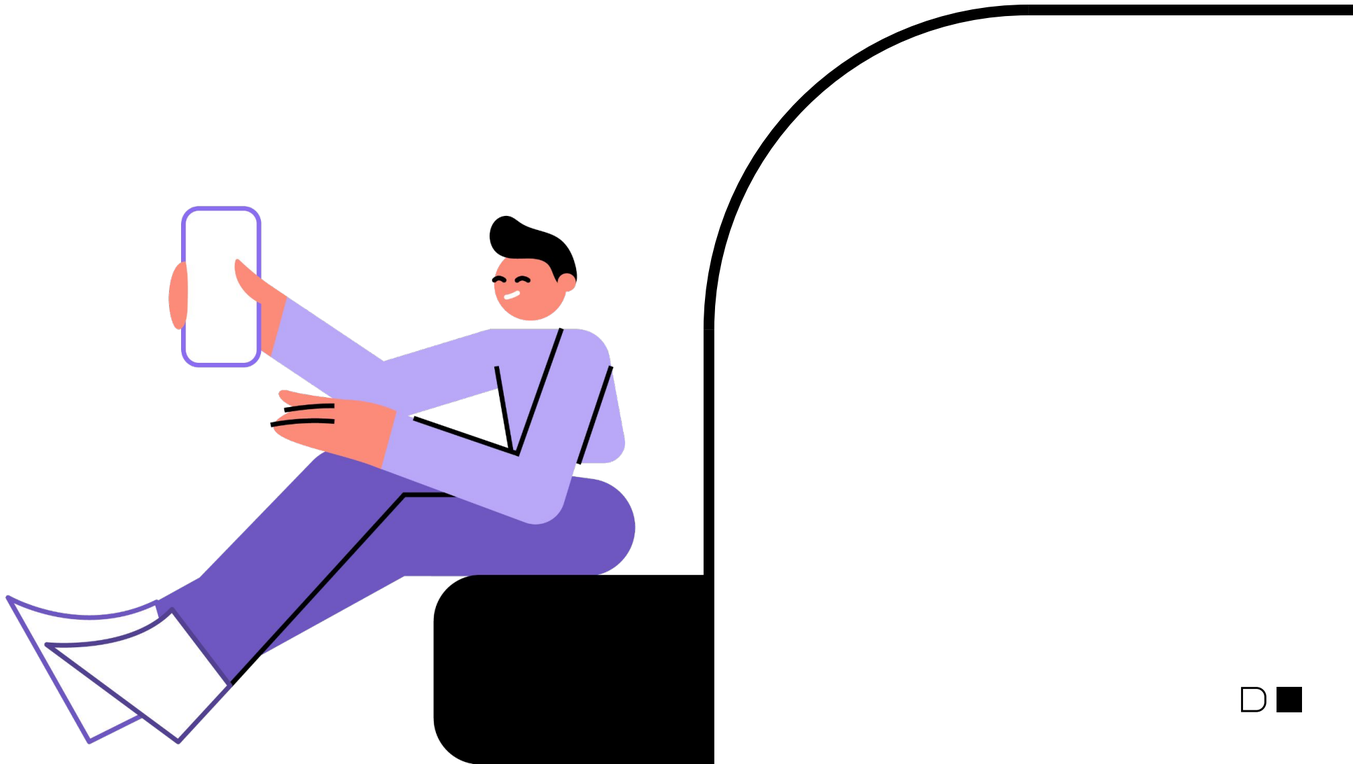
DIGITALHOUSE

Helm

Certified Tech Developer

The Ultimate Degree

Índice



01 ¿Qué es Helm?

Tres grandes conceptos

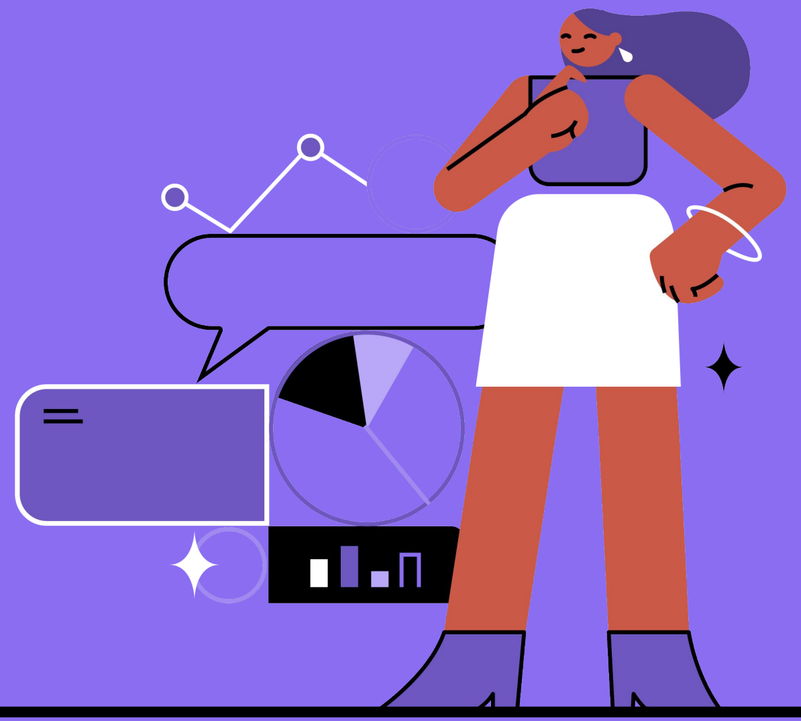
Un **chart** es un paquete de Helm. Contiene todas las definiciones de recursos necesarias para ejecutar una aplicación, herramienta o servicio dentro de un cluster de Kubernetes. Pensemos en ello como el equivalente de Kubernetes de una fórmula Homebrew, un archivo Apt dpkg o Yum RPM.

Un **repositorio** es el lugar donde se pueden recopilar y compartir charts. Es como el archivo CPAN de Perl o la base de datos de paquetes de Fedora, pero para paquetes de Kubernetes.

Una **release** es una instancia de un chart que se ejecuta en un cluster de Kubernetes. A menudo, un chart se puede instalar muchas veces en el mismo grupo. Y cada vez que se instala, se crea una nueva release. Consideremos un chart de MySQL. Si deseamos que se ejecuten dos bases de datos en un cluster, podemos instalar ese chart dos veces. Cada uno tendrá su propia release, que a su vez tendrá su propio nombre de release.

Con estos conceptos en mente, ahora podemos explicar Helm así:

Helm instala charts en Kubernetes y crea una nueva release para cada instalación. Y para encontrar nuevos charts, puede buscar en los repositorios de charts de Helm.



02

¿Cómo se usa?

Uso básico

helm init # deja que helm configure ambos archivos de datos locales e instale su componente de servidor.
helm search # buscar charts disponibles (usamos helm search <repo-name>/ para buscar solo un repositorio en particular).
helm install <chart-name> # instalar un chart (usamos values para especificar un archivo de valores personalizado).
helm inspect values<chart-name> # obtenemos el archivo de valores base de un chart para personalizarlo.
helm list # lista de charts instalados ('releases').
helm delete # eliminamos una versión (usamos --purge para eliminarla por completo).

03 Estructura de un chart

Estructura de un chart

charts.yaml

Contiene los metadatos del chart.

values.yaml

Contiene la configuración de chart predeterminada.

templates/

Contiene la esencia del chart, todos los archivos yaml que describen objetos de kubernetes.

templates/_helpers.tpl

Archivo opcional que puede contener código de ayuda para completar las plantillas.

04

¿Cómo crear un repositorio?

Crear un repositorio de charts

Un repositorio es simplemente un directorio que contiene un archivo index y charts empaquetados como tarballs que se sirven a través de HTTP(S).

helm package <chart-directory> # empaqueta un chart en el directorio actual.
helm repo index. # (re)construye el archivo index del directorio actual.
helm repo add <repo-name> <repo-addr> # agrega un repositorio no oficial.

Nota: es posible instalar un chart local sin pasar por un repositorio, lo cual es muy útil para el desarrollo, solo usamos: **helm install** <chart-directory>

DIGITALHOUSE

¡Muchas gracias!