



kubernetes

Kubernetes: HELM Intro

KUBERNETES : Package & Deploy

- **Helm** is a Kubernetes-based package installer.
- Helm is the best way to find, share, and use software built for Kubernetes.
- Helm helps user to maintain the Application.
- Helm is managed by **CNCF - Cloud Native Computing Foundation**.
- Helm manages Kubernetes “charts”, which are “preconfigured packages of Kubernetes resources.”
- Helm enables you to easily install packages, make revisions, and even roll back complex changes.

KUBERNETES : Administration

Why we Need Helm

- **Helm** lets you fetch, deploy and manage the lifecycle of applications, both 3rd party products and your own.
- With **helm**, there is a structure and a convention for a software package that defines a layer of **YAML templates** and another layer that changes the templates called **values**.
- **Values** are injected into templates, thus allowing a **separation of configuration**, and defines where changes are allowed.
- This whole package is called a **Helm Chart**.
- Essentially you create structured application packages that contain everything they need to run on a Kubernetes cluster; including **dependencies** the application requires.

KUBERNETES : Administration

Helm CLI or Tiller

- **Helm** is a CLI tool that interacts with its backend server called “Tiller”.
- Tiller is typically installed by sending the command **helm-init** and lives in the **kube-system** namespace.
- When a Chart is installed, Tiller creates a “Release” and starts tracking it for changes.
- This way Helm not only takes part in installation but is an actual deploy tool that manages the lifecycle of applications in a cluster using **Chart Releases** and **their revisions**.

KUBERNETES : Administration

Helm Charts or Packages

- **Helm** uses Charts to pack all the required K8S components for an application to deploy, run and scale.
- Helm chart is simply a **collection of YAML** template files organized into a specific directory structure.
- Chart is also where dependencies are defined, and configurations are updated and maintained.
- A chart root has to have only one file named **Chart.yml** by convention.
- **Templates** are an optional subdirectory in a chart that combines the K8S components of it, e.g. Services, ReplicaSet, Deployment etc.

KUBERNETES : Administration

Helm Charts or Packages

- **Values** are described in the **values.yml** file which is necessarily a YAML structure holding values to match the templates.
- Values basically define the Deployment values like name, labels.
- **Subcharts** also named dependencies, are required charts for the current one.
- Another way of using **Subcharts** is considering it as an inheritance mechanism which allows fetching a standard chart with templates and uses it as a **Subcharts** in multiple parent charts that would provide the values.

KUBERNETES : Administration

Helm concepts | Repository

- **Repositories** are where helm charts are held and maintained.
- Sometimes HELM packed repository as a **.tar.gz** file.

Helm concepts | Release

- **Release** is a mechanism to track installed applications on a K8S cluster; when an application is installed by **Helm**, a release is being created.
- Releases can be tracked with **helm ls**, each would have a “**revision**” which is the Helm release versioning terminology; if a specific release is updated, e.g., adding more memory to the Redis release, the revision would be incremented.
- Helm allows rolling back to a particular revision, making it virtually the manager for deployments and production status handler.

Will see you in Next Lecture...

Thank you!

A close-up photograph of a hand holding a black marker, writing the words 'Thank you!' in a cursive script on a white surface. The hand is positioned on the right side of the frame, with the marker tip touching the paper. The background is plain white.

See you in next lecture ...