Helm

Overview :

Helm is a package manager for Kubernetes, often referred to as the “Kubernetes Package Manager”. It simplifies the deployment, management, and maintenance of Kubernetes applications by packaging Kubernetes resources like Deployments, Services, and ConfigMaps into Helm Charts.

Key Concepts of Helm :

1. *Helm Charts : A Helm chart is a collection of YAML files that describe a set of Kubernetes resources required to run an application.*
2. *Release : A Helm release is an instance of a chrt running in a Kubernetes cluster.*
3. *Repository : Helm repositories store and distributed charts .*
4. *Tiller (Deprecated) : Prior to Helm v3, Tiller was the server-side component that managed releases.*
5. *Values : Values are customizable parameters that override the default configurationsin Helm charts.*

Why Use Helm in DevOps

Helm plays a critical role in DevOps by addressing the complexities involved in deploying and managing Kubernetes applications , It provides several benefits :

1.Simplifies Kubernetes Deployments
2. Reuseability and Consistency
3. Versioning and Rollbacks
4. Customization with Values
5. Chart Repositories
6. CI/CD Integration

Helm Architecture :

- *Helm CLI*
- *Chart*
- *Release*

Helm v3 Workflow :

1. *Install a chart*

2. *Upgrade a Release*
3. *Rollblock a Release*

Basic Helm Commands :

1. Install Helm on your local machine :

curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 | bash

2. Adding a Helm Repository :

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

3. Searching for Charts

helm search repo stable

4. Installing a Helm Chart

helm install <release-name> stable/<chart-name>

5.Listing Release

helm list

6. Upgrading a Release

helm upgrade <release-name> stable/<chart-name>

7. Uninstalling a Release

helm uninstall <release-name>

8. Viewing Release History

helm history <release-name>

9. Rolling Back a Release

helm rollback <release-name> <revision-number>