

# (Helm in Kubernetes)

## Helm Explained :-

- In this document, we are going to discuss on the overview of the "Helm" and Helm-Chart.

### Overview :-

- (\*) What is Helm?
- (\*) What are Helm Charts?
- (\*) How to use them?
- (\*) When to use them?
- (\*) What is Tiller?

#### ① What is Helm?

- So, Helm is the package manager in K8s.
- Helm combined our (.yaml) file and put it in a Repository.
- And the Helm repository, may be a public repositories or a private repos.
- So, in this way, that helps us to - reduce to write .yaml conf. file each time for our every file.
- And it helps us to, provide a no. of Repositories library in a "Helm hub".
- Only thing we take that repositories and use it as per our requirement.

Example :-

- We have a kubernetes cluster.
- And in that cluster, we have run some application.
- And for tracking and monitoring of the cluster, we use "ELASTIC STACK" for logs collection inside in our cluster.
- So, in order to use "Elastic Stack" for logs inside in a K8s cluster, we need some configuration in our K8S cluster.

(a) Stateful Set :-

This is used to manage and take all details of our replicaset of the "data base".

(b) Config Map :-

This is helps to manage all of our external configuration.

(c) Secrets :-

Secrets helps in maintain and manage all our secrets and passworded credential of our all services.

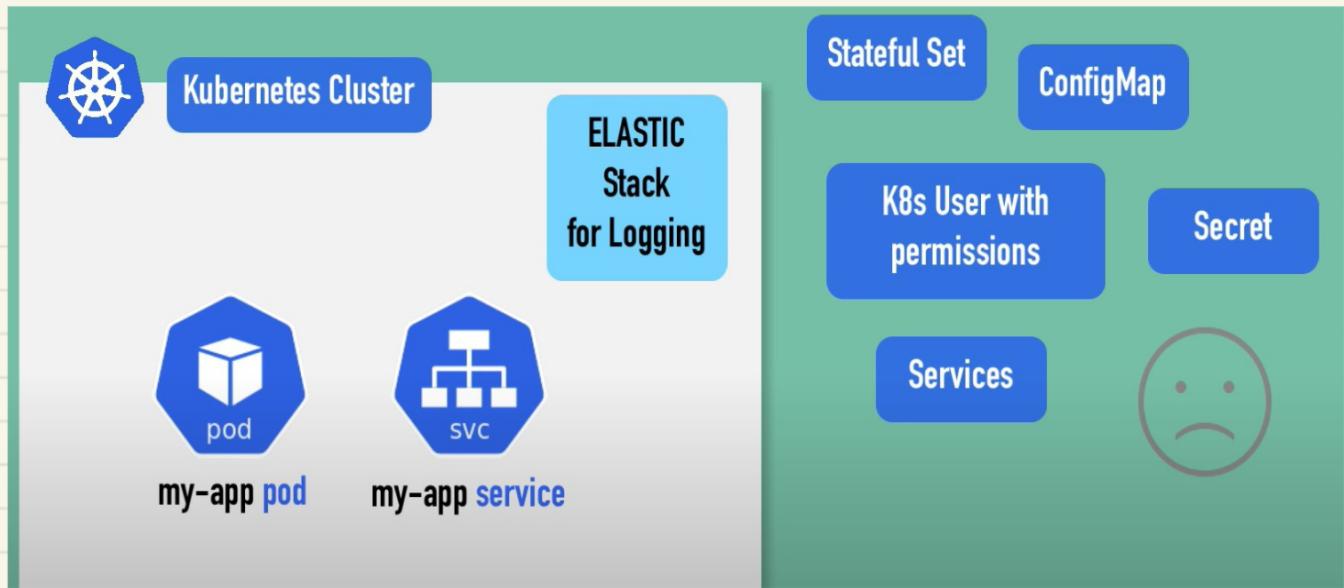
(d) K8s users with permission :-

This is helps to add our user and manage their permission.

## (c) Services :-

Also we need some couple of services file.

### Diagram (Ex.)



## Helm charts :-

- So every time create those files and use it for your cluster, it's a headache.
- So, what should we do, we take all those files once and bundle it and put it in a private or public repositories.
- And in this way we can create a no. of repo's libraries with different API's conf. only we should pull a repo's as per our requirement from "helm hub" and use it.
- And this helm packages are known as - "Helm chart".

## Overview :-

- Bundle of YAML file.
- Create your own Helm Charts with Helm.
- We can push our Helm package to Helm repositories.
- Then we can download and use existing one.

## Common uses of Helm Chart :-

- we use Helm chart, for common applications deployment.

like :-

### ① Database Apps

\* MongoDB

\* Elasticsearch

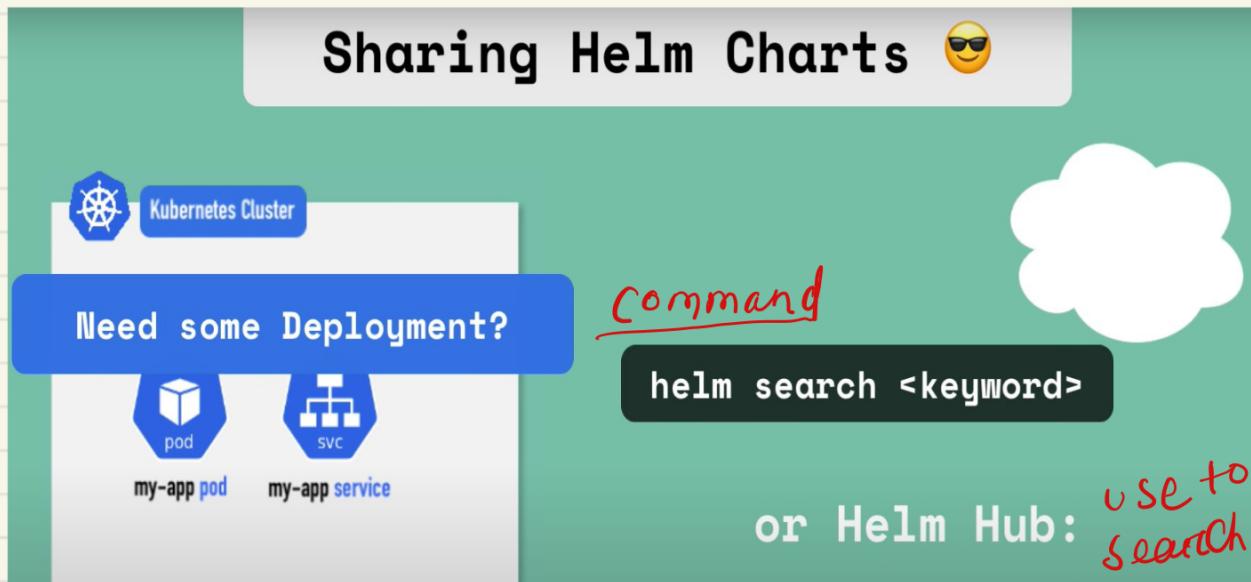
\* MySQL

### ② Monitoring Apps

\* Prometheus

\* Loki

# Sharing Helm Charts



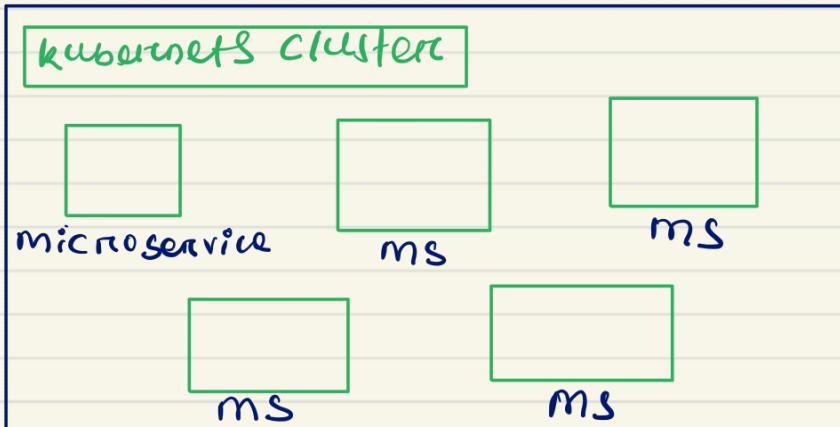
→ Again we have public and private repositories.

→ So some charts we can use free and others takes charges.

## Second Feature in Helm

### Templating Engine

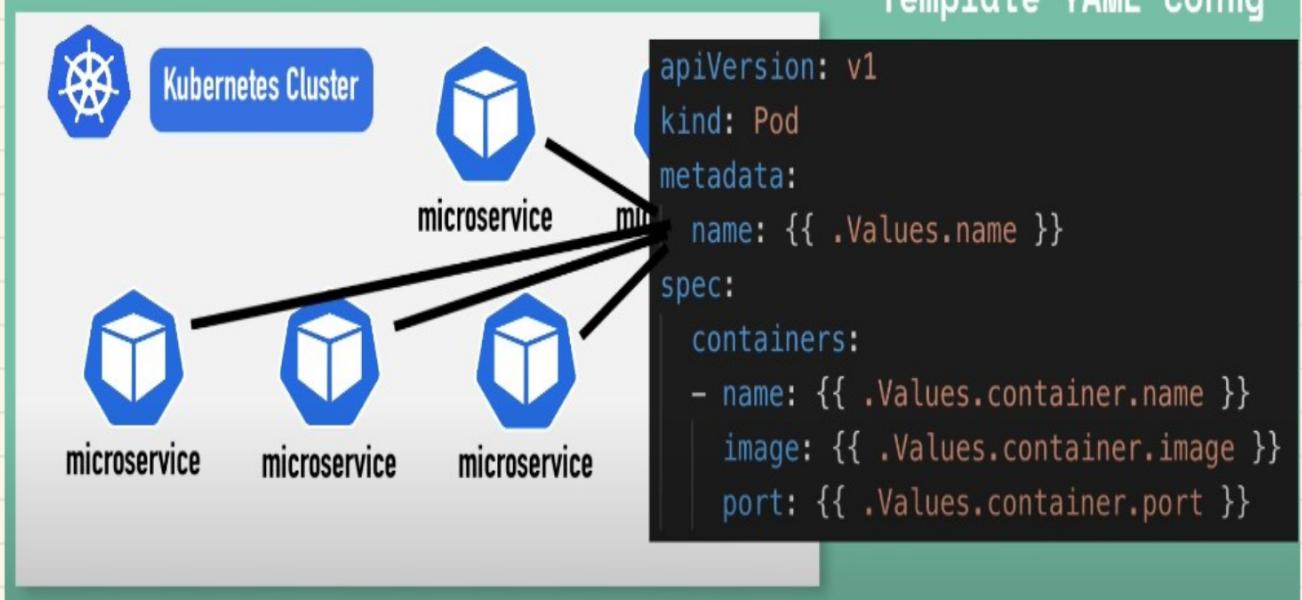
→ If you have a k8s cluster and inside a node, you have running multiple "microservices" pod running.



- And for each microservices, we have separate (.yaml) config. file.
  - But there if you see the .yaml file, you find ONLY some common changes in the values.
  - So, instead of creating multiple .yaml file for each microservices.
  - We should create a "common blueprint".
  - And "Dynamicity" values are replaced by placeholders.
  - So we can pass those values in a variable.
- Example :-
- ```
apiVersion: v1
kind: pod
metadata:
  name: {{.Values.name}}
spec:
  containers:
    - name: {{.values.containers.name}}
      image: {{.values.containers.image}}
      port: {{.values.containers.port}}
```

# Templating Engine

Template YAML config



Note Prepared By  
Savan Kcenarc Rocet  
25.10.2024  
(Bharat)