Helm

By Karthik Veeragoni

Agenda

1. Introduction to Helm
2. Helm Chart Layout
3. Helm Architecture
   - V2
   - V3
4. Helm v3 Updates
5. Helm Charts Customization
6. Demo

Senior Development Engineer at
Pramati Technologies
https://www.linkedin.com/in/karthikvee/

Introduction to Helm

Helm Terminology	Helm Analogies
What is Helm?	Ubuntu/CentOS — Kubernetes Cluster
What is Helm Chart?	apt, yum — Helm CLI
What is a Release?	.deb, .rpm — Helm Chart
What is Helm Repository?	Debian Repository (packages.ubuntu.com) — Helm Charts Repository (hub.helm.sh)
Why use Helm?	
Helm Installation	
Helm CLI Usage	

kubernetes

Helm Chart Layout

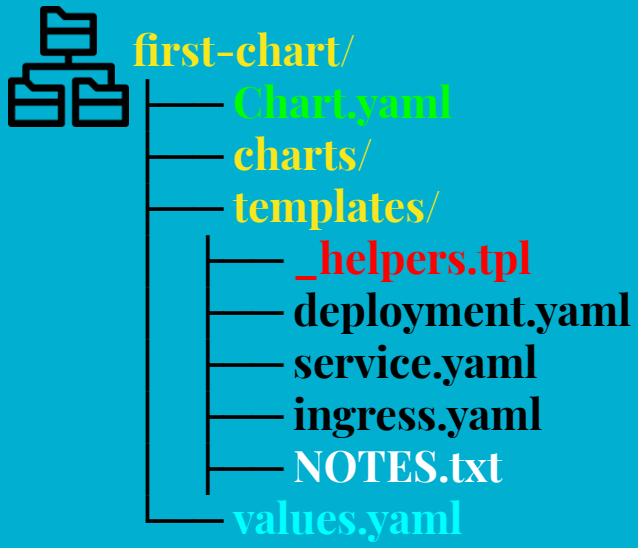


Chart.yaml: This is where you put all the information about the chart you are packaging.

Charts: Dependent charts are placed in this directory

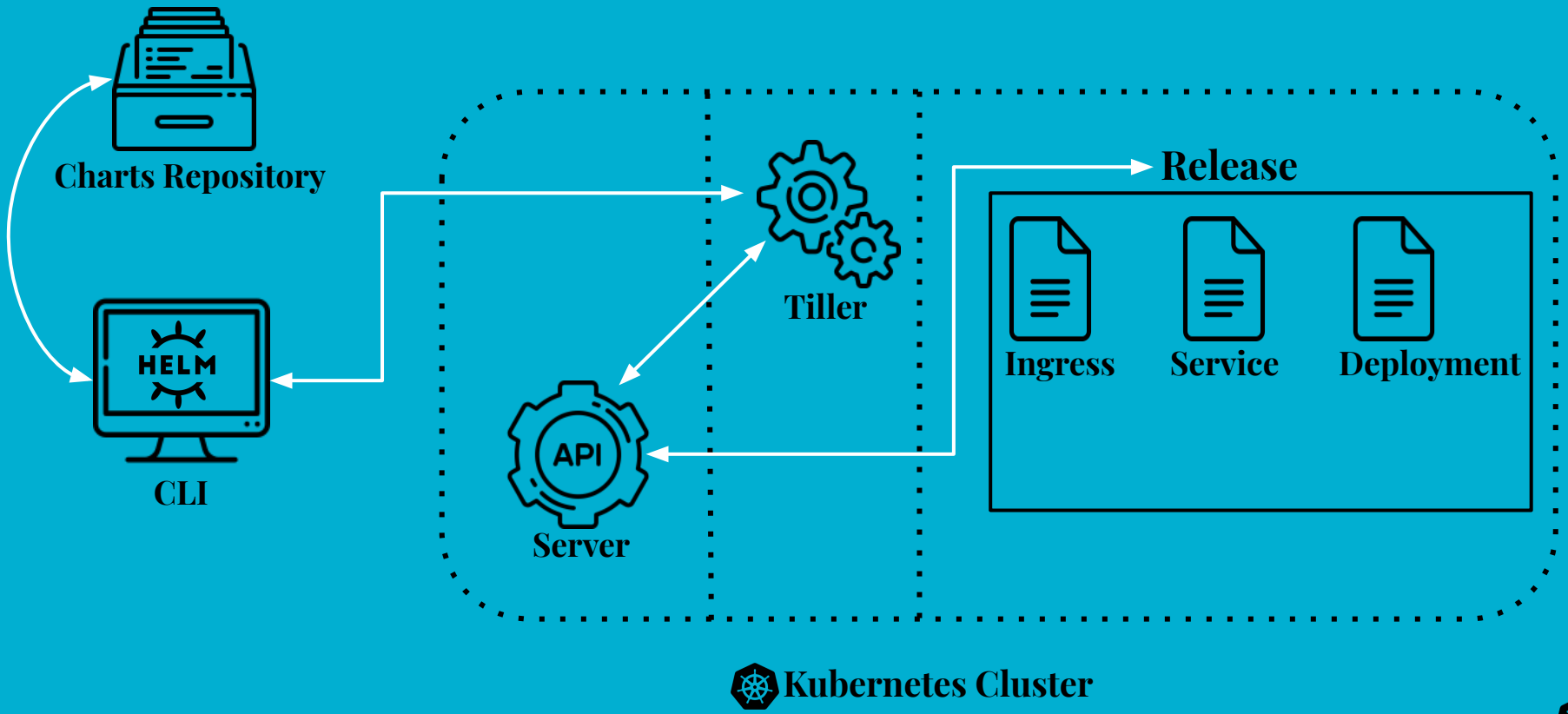
Templates: A directory of templates that when combined with values, will generate valid Kubernetes manifest files

Values.yaml: This is where you define all the values you want to inject into your templates.

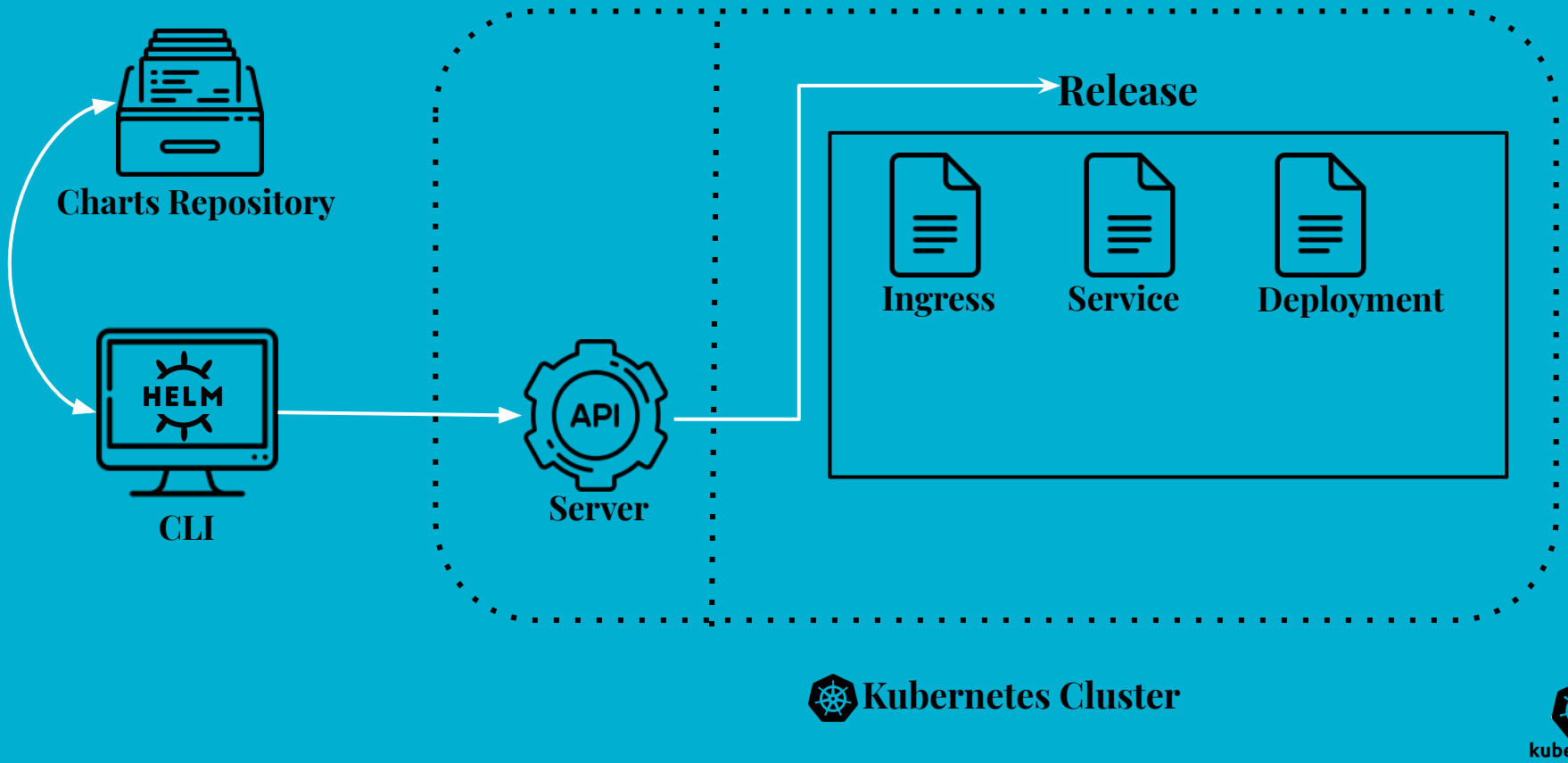
NOTES.txt: The "help text" for your chart. This will be displayed to your users when they run helm install

kubernetes

Helm-V2 Architecture



Helm-V3 Architecture



What new in Helm-V3?

- ➔ Release information stored in the cluster as secrets
- ➔ The apiVersion field in Chart.yaml is changed to v2
- ➔ Chart dependencies are declared in Chart.yaml
- ➔ Chart values validation using values.schema.json
- ➔ Chart repository information is to be added manually
- ➔ Upgrade strategy: Three-way strategic merge patch

Helm Charts Customization

Built-in Objects	Template Functions	Flow Control/ Control Structures	Named Templates
Release	quote	if / else -⤑ conditions	define
Chart	default	with -⤑ scope	template/include
Values	trimSuffix	range -⤑ for loop	block
Files	b64enc	handling white spaces	
Template	upper	variables	
Capabilities	**Syntax:-** functionName arg1 arg2...		

kubernetes

Built-in Objects: Values

```
apiVersion: v1
kind: service
metadata:
    name: {{ .Values.service.name }}
```

mychart/templates/service.yaml

```
service:
  type: NodePort
  name: nginx-service
  port: 80
```

values.yaml

Rendered Outcome of:
templates/service.yaml

```
apiVersion: v1
kind: service
metadata:
    name: nginx-service
```

kubernetes

Built-in Objects: Chart

```
apiVersion: v1
kind: service
metadata:
    name: {{ .Chart.Name }}
```

mychart/templates/service.yaml

Rendered Outcome of:
templates/service.yaml

```
apiVersion: v1
kind: service
metadata:
    name: nginx
```

```
version: 1.0.0
name: nginx
appVersion: "2.0"
```

Chart.yaml

kubernetes

Built-in Objects: Release

```
apiVersion: v1
kind: service
metadata:
    name: {{ .Release.Name }}
```

mychart/templates/service.yaml

Rendered Outcome of:
templates/service.yaml

```
apiVersion: v1
kind: service
metadata:
    name: nginx-app
```

```
Release.Name
Release.Namespace
Release.Service
Release.Revision
```

Release Runtime Data

$ helm install nginx-app chart-directory ➳ nginx-app is release name

kubernetes

References:

https://helm.sh/

Ibm developer blog articles