Helm Crash Course

Agenda

- Resources
- Helm Chart Structure
- Adding Values
- Named Templates
- Dependencies
- Hooks

Resources

The starting set of (6) resources for the demo:

- One Namespace definition
- Two Deployment definitions
- Two Service definitions
- One Ingress (rule) definition

> Question
>
> Would you want to `kubectl apply -f {{file}}` every file, every time?

Helm Chart Structure

Create a `Chart.yaml` and smash your resources into a `templates` sub-directory.

Directory Structure

- Project root directory/
  - Chart.yaml
  - templates/
    - Resource definitions go here

Chart.yaml

```yaml
apiVersion: v2
name: microapps # chart name
version: 0.1.0 # chart version
appVersion: "0.0.1" # application version
description: Demo apps for Helm crash course # chart description
type: application
```

Values

Values are YAML key/value pairs that can be referenced within our resource templates.

Values.yaml

```yaml
image:
  registry: docker.io # container image registry
  tag: 0.0.1 # container image tag
```

Deployment.yaml (Template)

```yaml
...
spec:
  containers:
    - name: example
      image: "{{ .Values.image.registry }}/imagename:{{
.Values.image.tag }}"
...
```

There are additional "values" that you can use in your templates, which originate from other places. Common examples are:

- `{{ .Release.Namespace }}`
  - This represents the namespace that the Helm chart is being deployed to
- `{{ .Chart.AppVersion }}`
  - The version of the application, which is sourced from the `Chart.yaml` file

Named Templates

Named templates allow you to define reusable blocks of YAML, which can be used within (m)any of your templates.

Start by creating a file in your templates directory that begins with the `_` character and has the `tpl` extension. `_helpers.tpl`, for example.

Named Template Definition

```
{{- define "do.the.thing" }}
annotations:
    nginx.ingress.kubernetes.io/enable-opentelemetry: "
true"
    nginx.ingress.kubernetes.io/opentelemetry-trust-
incoming-span: "true"
{{- end }}
```

Usage in Resource Template

```
apiVersion: networking.k8s.io/v1
kind: Ingress
metadata:
  name: microapp
  {{- template "do.the.thing" }}
...
```

We're going to do something a little funky though...

Dependencies

In your `Chart.yaml` file, you can define dependencies (a.k.a subcharts).

- Additional/external charts
- Pulled from a Helm repository
- Deployed alongside your chart's resources

Chart.yaml

```yaml
apiVersion: v2
name: microapps
version: 0.1.0 # chart version
appVersion: "0.0.1" # application version
description: Demo apps for Helm crash course # chart description
type: application
dependencies:
  - name: ingress-nginx # Chart name
    version: 4.12.2 # Chart version (**not** app version)
    repository: "https://kubernetes.github.io/ingress-nginx" # Chart repository
    condition: ingress.enabled
```

Values.yaml

```yaml
...
ingress-nginx: # Chart name of the dependency
  controller:
    metrics:
      enabled: true
    podAnnotations:
      prometheus.io/port: "10254"
      prometheus.io/scrape: "true"
    replicaCount: 1
...
```

Dependencies

You can also use aliases for your dependencies:

Chart.yaml

```yaml
...
dependencies:
  - name: ingress-nginx # chart name
    version: 4.12.2 # chart version (**not** app version)
    repository: "https://kubernetes.github.io/ingress-nginx" # chart repository
    condition: ingress.enabled
    alias: ingresscontroller # we can name this whatever we like
```

Values.yaml

```yaml
...
ingressController: # Using the alias
  controller:
    metrics:
      enabled: true
    podAnnotations:
      prometheus.io/port: "10254"
      prometheus.io/scrape: "true"
    replicaCount: 1
...
```

Hooks

Hooks can be used to run Jobs at a defined point in the Helm deployment lifecycle.

Definition

```yaml
apiVersion: batch/v1
kind: Job
metadata:
  name: "{{ .Release.Name }}"
  annotations:
    "helm.sh/hook": post-install,post-upgrade
    "helm.sh/hook-weight": "1"
    "helm.sh/hook-delete-policy": hook-succeeded
```

Common Hook Stages

Stage	Description
pre-install	Runs before resources are installed
pre-upgrade	Runs before resources are upgrades
post-install	Runs after resources are installed
post-upgrade	Runs after resources are upgraded
post-delete	Runs after resources are deleted

Additional Info

This Demo

- Git Repository (https://github.com/iambenzo/helm-crash)

Helm Documentation

- Chart.yaml Documentation (https://helm.sh/docs/topics/charts/#the-chartyaml-file)
- Additional "Values" (https://helm.sh/docs/chart_template_guide/builtin_objects/)
- Named Templates (https://helm.sh/docs/chart_template_guide/named_templates/)
- Hooks (https://helm.sh/docs/topics/charts_hooks/)