Helm

yongping.ren@ygomi.com

What is Helm?

Helm is a tool for managing Kubernetes packages

What is chart?

Helm uses a packaging format called charts. A chart is a collection of files that describe a related set of Kubernetes resources

Helm -> charts==apt-get -> deb

What can Helm do?

- Create new charts from scratch
- Package charts into chart archive (tgz) files
- Interact with chart repositories where charts are stored
- Install and uninstall charts into an existing Kubernetes cluster
- Manage the release cycle of charts that have been installed with Helm

A simple Helm chart

```
[➜  kubernetes tree nginx
nginx
├── Chart.yaml
├── charts
│   └── common-0.0.5.tgz
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   └── service.yaml
└── values.yaml

2 directories, 7 files
```

Walk into Helm chart

- Chart.yaml
- Template
- Built-in Objects
- Values
- Template Functions and Pipelines
- Named Templates
- NOTES.txt

Chart.yaml

A YAML file containing information about the chart

```yaml
# The chart API version (required)
apiVersion: v2

# The name of the chart (required)
name: nginx

description: A Helm chart for Kubernetes

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged into versioned archives
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer. They're included as
# a dependency of application charts to inject those utilities and functions into the rendering
# pipeline. Library charts do not define any templates and therefore cannot be deployed.
type: application

# This is the chart version. This version number should be incremented each time you make changes
# to the chart and its templates, including the app version.
# A SemVer 2 version (required)
version: 0.1.0

# This is the version number of the application being deployed. This version number should be
# incremented each time you make changes to the application.
appVersion: 1.16.0
```

Dependency

- Directly put charts into folder ./mychart/charts/
- Define Dependency in Chart.yaml
  - name
  - version
  - repository

```
➜ kubernetes tree nginx
nginx
├── Chart.yaml
├── charts
│   └── common-0.0.5.tgz
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   └── service.yaml
└── values.yaml

2 directories, 7 files
```

```
dependencies: # A list of the chart requirements (optional)
  - name: The name of the chart (nginx)
    version: The version of the chart ("1.2.3")
    repository: The repository URL ("https://example.com/charts")
```

Template

The templates/ directory is for template files. When Helm evaluates a chart, it will send all of the files in the templates/ directory through the template rendering engine. It then collects the results of those templates and sends them on to Kubernetes.

```
➜  nginx tree templates
templates
├── NOTES.txt
├── _helpers.tpl
├── deployment.yaml
└── service.yaml

0 directories, 4 files
```

- NOTES.txt: The “help text” for your chart. This will be displayed to your users when they run helm install.
- deployment.yaml: A basic manifest for creating a kubernetes deployment
- service.yaml: A basic manifest for creating a service endpoint for your deployment
- _helpers.tpl: A place to put template helpers that you can re-use throughout the chart

Debug: helm install --debug --dry-run ./mychart --generate-name

Built-in Objects

- Release
- Values
- Files
- Capabilities
- Template

Values Files

Its contents come from four sources：

- The values.yaml file in the chart
- If this is a subchart, the values.yaml file of a parent chart
- A values file if passed into helm install or helm upgrade with the -f flag (helm install -f myvals.yaml ./mychart)
- Individual parameters passed with --set (such as helm install --set foo=bar ./mychart)

The list above is in order of specificity: values.yaml is the default, which can be overridden by a parent chart’s values.yaml, which can in turn be overridden by a user-supplied values file, which can in turn be overridden by --set parameters.

Template Functions and Pipelines

functionName arg1 arg2...

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hello World"
  drink: {{ quote .Values.favorite.drink }}
  food: {{ quote .Values.favorite.food }}
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
data:
  myvalue: "Hello World"
  drink: {{ .Values.favorite.drink | quote }}
  food: {{ .Values.favorite.food | quote }}
```

Named Templates

An important detail to keep in mind when naming templates: template names are global. If you declare two templates with the same name, whichever one is loaded last will be the one used. Because templates in subcharts are compiled together with top-level templates, you should be careful to name your templates with chart-specific names.

```
{{- define "mychart.labels" }}
  labels:
    generator: helm
    date: {{ now | htmlDate }}
{{- end }}
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: {{ .Release.Name }}-configmap
  {{- template "mychart.labels" }}
data:
  myvalue: "Hello World"
  {{- range $key, $val := .Values.favorite }}
  {{ $key }}: {{ $val | quote }}
  {{- end }}
```

NOTES.txt

providing instructions to your chart users. At the end of a chart install or chart upgrade, Helm can print out a block of helpful information for users.

```
NOTES:
1. Get the application URL by running these commands:

Use the expose command with NodePort to create a new service and expose it to external traffic

sudo kubectl get deployment

sudo kubectl expose deployment/${deployment-name} --type="NodePort" --port 80

sudo kubectl get service

sudo minikube service ${service-name} --url
```

Create a simple helm chart

```
➜  kubernetes helm create nginx
Creating nginx
➜  kubernetes tree nginx
nginx
├── Chart.yaml
├── charts
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   ├── ingress.yaml
│   ├── service.yaml
│   ├── serviceaccount.yaml
│   └── tests
│       └── test-connection.yaml
└── values.yaml

3 directories, 9 files
```

Delete serviceaccount, ingress and test

```
[➜  kubernetes rm -rf nginx/templates/ingress.yaml
[➜  kubernetes rm -rf nginx/templates/serviceaccount.yaml
[➜  kubernetes rm -rf nginx/templates/tests
[➜  kubernetes tree nginx
nginx
├── Chart.yaml
├── charts
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   └── service.yaml
└── values.yaml

2 directories, 6 files
```

Modify Chart.yaml

```yaml
# The chart API version (required)
apiVersion: v2

# The name of the chart (required)
name: nginx

description: A Helm chart for Kubernetes

# A chart can be either an 'application' or a 'library' chart.
#
# Application charts are a collection of templates that can be packaged into versioned archives
# to be deployed.
#
# Library charts provide useful utilities or functions for the chart developer. They're included as
# a dependency of application charts to inject those utilities and functions into the rendering
# pipeline. Library charts do not define any templates and therefore cannot be deployed.
type: application

# This is the chart version. This version number should be incremented each time you make changes
# to the chart and its templates, including the app version.
# A SemVer 2 version (required)
version: 0.1.0

# This is the version number of the application being deployed. This version number should be
# incremented each time you make changes to the application.
appVersion: 1.16.0
```

Modify values.yaml

```yaml
# Default values for nginx.
# This is a YAML-formatted file.
# Declare variables to be passed into your templates.

replicaCount: 1

image:
  repository: nginx
  pullPolicy: IfNotPresent

service:
  port: 80
  type: NodePort
```

Modify deployment.yaml

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ include "nginx.fullname" . }}
  labels:
    {{- include "nginx.labels" . | nindent 4 }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      {{- include "nginx.selectorLabels" . | nindent 6 }}
  template:
    metadata:
      labels:
        {{- include "nginx.selectorLabels" . | nindent 8 }}
    spec:
      containers:
      - name: {{ .Chart.Name }}
        image: "{{ .Values.image.repository }}:{{ .Chart.AppVersion }}"
        ports:
        - containerPort: 80
```

Add library chart

```
➜ kubernetes tree nginx
nginx
├── Chart.yaml
├── charts
│   └── common-0.0.5.tgz
├── templates
│   ├── NOTES.txt
│   ├── _helpers.tpl
│   ├── deployment.yaml
│   └── service.yaml
└── values.yaml

2 directories, 7 files
```

Modify service.yaml

```
{{ template "common.service" (list . "mychart.nginx.service") -}}
{{- define "mychart.nginx.service" -}}
metadata:
  name: {{ template "common.fullname" . }}  # overrides the default name to add a suffix
  labels:                                   # appended to the labels section
    protocol: TCP
spec:
  ports:                                    # composes the `ports` section of the service
  - name: www
    port: 80
    targetPort: 80
    nodePort: 30001
{{- end -}}
```

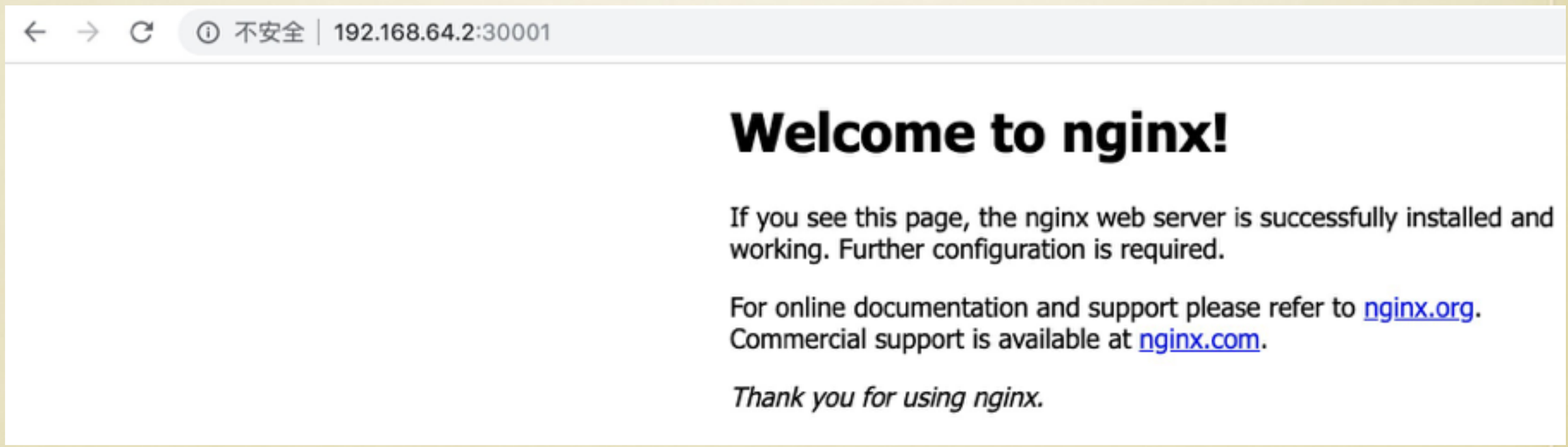
Note: If you do not want to declare a service , you can also use kubectl expose deployment/${deployment-name} --type="NodePort" --port 80 to manually expose a service

Use debug mode to check rendering

```
# Source: nginx/templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  labels:
    app: nginx
    chart: nginx-0.1.0
    heritage: Helm
    protocol: TCP
    release: nginx-1577347819
  name: nginx-1577347819-nginx
spec:
  ports:
  - name: www
    nodePort: 30001
    port: 80
    targetPort: 80
  selector:
    app: nginx
    release: nginx-1577347819
  type: NodePort
---
# Source: nginx/templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-1577347819
  labels:
    helm.sh/chart: nginx-0.1.0
    app: nginx
    release: nginx-1577347819
    app.kubernetes.io/version: "1.16.0"
    app.kubernetes.io/managed-by: Helm
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
      release: nginx-1577347819
  template:
    metadata:
      labels:
        app: nginx
        release: nginx-1577347819
    spec:
      containers:
      - name: nginx
        image: "nginx:1.16.0"
        ports:
        - containerPort: 80
```

Package, install and access service

```
→ kubernetes helm package nginx
Successfully packaged chart and saved it to: /Users/ypren/kubernetes/nginx-0.1.0.tgz
→ kubernetes sudo helm install my-nginx nginx-0.1.0.tgz
NAME: my-nginx
LAST DEPLOYED: Thu Dec 26 17:03:39 2019
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
1. Get the application URL by running these commands:

Use the expose command with NodePort to create a new service and expose it to external traffic

sudo kubectl get deployment

sudo kubectl expose deployment/${deployment-name} --type="NodePort" --port 80

sudo kubectl get service

sudo minikube service ${service-name} --url
→ kubernetes sudo kubectl get service
NAME                          TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                    ClusterIP   10.96.0.1       <none>        443/TCP        2d14h
my-nginx-nginx                NodePort    10.96.189.214   <none>        80:30001/TCP   20s
redis-1576747274-headless     ClusterIP   None            <none>        6379/TCP       2d1h
redis-1576747274-master       ClusterIP   10.96.173.108   <none>        6379/TCP       2d1h
redis-1576747274-slave        ClusterIP   10.96.37.193    <none>        6379/TCP       2d1h
→ kubernetes sudo minikube service my-nginx-nginx --url
http://192.168.64.2:30001
```



Helm ls and uninstall

```
➜ kubernetes sudo helm ls
NAME                    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART           APP VERSION
nginx-0-1577348927      default         1               2019-12-26 16:28:48.327978 +0800 CST    deployed        nginx-0.1.0     1.16.0
redis-1576747274        default         1               2019-12-19 17:21:17.028653 +0800 CST    deployed        redis-10.2.1    5.0.7
➜ kubernetes sudo kubectl get service
NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)        AGE
kubernetes                  ClusterIP   10.96.0.1       <none>        443/TCP        2d13h
nginx-0-1577348927-nginx    NodePort    10.96.193.118   <none>        80:30001/TCP   13s
redis-1576747274-headless   ClusterIP   None            <none>        6379/TCP       2d
redis-1576747274-master     ClusterIP   10.96.173.108   <none>        6379/TCP       2d
redis-1576747274-slave      ClusterIP   10.96.37.193    <none>        6379/TCP       2d
➜ kubernetes sudo helm uninstall nginx-0-1577348927
release "nginx-0-1577348927" uninstalled
➜ kubernetes sudo helm ls
NAME                    NAMESPACE       REVISION        UPDATED                                 STATUS          CHART           APP VERSION
redis-1576747274        default         1               2019-12-19 17:21:17.028653 +0800 CST    deployed        redis-10.2.1    5.0.7
➜ kubernetes sudo kubectl get service
NAME                        TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)    AGE
kubernetes                  ClusterIP   10.96.0.1       <none>        443/TCP    2d13h
redis-1576747274-headless   ClusterIP   None            <none>        6379/TCP   2d
redis-1576747274-master     ClusterIP   10.96.173.108   <none>        6379/TCP   2d
redis-1576747274-slave      ClusterIP   10.96.37.193    <none>        6379/TCP   2d
```

Thank you