



SIGHUP


Kustomize vs Helm(v3)

2/10/19



kubernetes

I work as a DevOps Engineer at SIGHUP

- Milan based consulting company
- Specialized on Kubernetes and Cloud-Native infrastructures
- Battle Tested Kubernetes and Cloud Native stack with [Kubernetes Fury](#)
- [Distrubution](#)  **FURY**
- Kubernetes certified service provider
- CI/CD, DevOps, Automation, Infrastructure As Code



We're hiring!

<https://sighup.io>

<https://github.com/sighupio>

>> jobs@sighup.io <<



That question.. which tool is better?

Should i use **Kustomize** or **Helm** ?

...answer of all questions in computer science is:

That question.. which tool is better?

Should i use **Kustomize** or **Helm** ?

...answer of all questions in computer science is:

“it depends!”

Helm and kustomize are not strictly alternatives to each other

- Different focus - different problems
- Abstraction/separation of concerns vs Raw complexity
- Application config vs Runtime config

....It's possible even using them combined!

Helm

- A package manager for Kubernetes
- Abstract, self-contained

Kustomize

- A tool to customize raw, template-free YAML files
- No abstraction, fine grained control

What is Helm?

- **Chart** -> Packaging format, a collection of templated kubernetes manifests to deploy an application
- **Config** -> configuration information to be merged into a packaged chart to create a releasable object
- **Release** -> running instance of a *chart*, combined with a specific *config*



What is Kustomize?

- Customize raw, template-free YAML files for multiple purposes
- Leaving the original YAML untouched and usable as is
- It doesn't create any intermediary layer, just spits out "patched" YAMLs
- integrated to kubectl (kubernetes \geq [v1.14.7](#))

Helm charts

mychart/

- **Chart.yaml** # A YAML file containing information about the chart
- **values.yaml** # The default configuration values for this chart
- **charts/** # A directory containing any charts upon which this chart depends.
- **crds/** # Custom Resource Definitions
- **templates/** # A directory of templates that, when combined with values will generate valid

Kubernetes manifest files.

Chart repo -> where your chart code is hosted (e.g. github)

Chart registry -> where your chart is packed as an archive and pushed (e.g. docker registry)

Deploying with Helm

```
$ helm install my-mysql stable/mysql
```

```
NAME: mydb
```

```
LAST DEPLOYED: 2019-10-01 22:08:41.736214 +0200 CEST m=+2.435158573
```

```
NAMESPACE: default
```

```
STATUS: deployed
```

```
NOTES:
```

```
...
```

```
$ helm ls
```

NAME	VERSION	UPDATED	STATUS	CHART
my-mysql	1	Wed Sep 28 12:59:46 2016	DEPLOYED	mysql-0.1.0



Kustomize

kustomize build ~/someApp | kubectl apply -f -
kubectl apply -f ~/someApp

~/someApp

├─ deployment.yaml

├─ kustomization.yaml

└─ service.yaml

base: kustomization + resources

kustomization.yaml

```
commonLabels:  
  app: myWord  
resources:  
- deployment.yaml  
- service.yaml  
configMapGenerator:  
- name: wordpress-map  
  files:  
  - env.startup.txt
```

deployment.yaml

```
apiVersion: v1  
kind: Deployment  
metadata:  
  name: wordpress  
  labels:  
    app: wordpress  
spec:  
  replicas: 1  
  selector:  
    matchLabels:  
      app: wordpress  
  template: ...
```

service.yaml

```
apiVersion: v1  
kind: Service  
metadata:  
  name: wordpress  
spec:  
  ports:  
  - port: 389  
  selector:  
    app: wordpress
```

Kustomize overlays

kustomize build ~/someApp/production | kubectl apply -f -

~/someApp

```
├── base
│   ├── deployment.yaml
│   ├── kustomization.yaml
│   └── service.yaml
├── development
│   ├── cpu_count.yaml
│   ├── kustomization.yaml
│   └── replica_count.yaml
└── production
    ├── cpu_count.yaml
    ├── kustomization.yaml
    └── replica_count.yaml
```

overlay: **kustomization** + **patches** + more resources
(referencing a base)

kustomization.yaml

```
namePrefix: prod-
commonLabels:
  variant: prod
commonAnnotations:
  note: Hello, I am production!
bases:
- ../../base
patches:
- replica_count.yaml
- cpu_count.yaml
```

replica_count.yaml

```
apiVersion: v1
kind: Deployment
metadata:
  name: wordpress
spec:
  replicas: 80
```

cpu_count.yaml

```
apiVersion: v1
kind: Deployment
metadata:
  name: wordpress
spec:
  template:
    spec:
      containers:
      - name: my-container
        resources:
          limits:
            cpu: 7000m
```


Helm Advantages

More tools, more control over deployment life-cycle

- **helm lint** : Check if chart is well-formed
- **helm verify**: Verify that the given chart has a valid provenance file (so it's not been tampered with)
- **helm package --sign**: Generate a signed package
- **Chart Hooks**: allow chart developers to intervene at certain points in a release's life cycle:
 - Pre-install, Post-install, pre-delete, post-delete, pre-upgrade, post-upgrade etc
- **Chart Test**: Pod definition that specifies a container with a given command to run

What happens when you want to customize a Helm Chart?

...You FORK it! And then just try to keep up with upstream chart...

What happens when you want to customize third party manifests with kustomize?

...No fork! You can just REBASE it since kustomize doesn't touch to original files

What happens when you want to customize a Helm Chart?

What happens when you want to customize a Helm Chart?

...You

Helm Chart Example: stable/prometheus

129 total commits to Chart

Started on Nov 14, 2016 (2 years ago) - Averaging 1 commit per week

100 "make this configurable", 19 "updates", 10 "helm chores"

Started with 415 lines in Values.yaml

Now has 1116 lines in Values.yaml

The chart has 10k+ lines of TEMPLATED yaml

And this chart was in Incubator before!

...No fork! You c

(stolen slide)

What is your use case?

- What kind of applications you work with?
- Applications you need in order to run your custom application
 - Databases, caches, brokers, etc
- Infra applications
 - Monitoring, Logging, Disaster Recovery, Networking, etc
- Do you deal with just application configuration
- How often do you need to deal with also runtime configuration?



(spero di non)

ARRIVEDERCI

<http://sighup.io>



kubernetes