TUT1139 Kubernetes Helm for beginners
How to use Helm in SUSE CaaS Platform

Jonas Forsberg / Peter Lunderbye
SE
SUSE
Agenda

* Short description of CaaS
* Demo: kubectl
* What is HELM
* Demo: helm
* wrap-up

Demo Application

* Deployment with kubectl
* building a helm Chart
* Expanding the helm Chart
Building a container stack from the ground up is not for everyone.

Container Services in the Public Cloud. Gartner 2017
You need more than just the engine.

You need to manage. And you need to scale.

Provision Manage Automate Host Services
Deploying at Scale Requires Automation

Orchestration
• Scheduling
• Service discovery

Performance and availability
• Scaling
• Load balancing
• Self-healing
• Monitoring

Maintenance
• Rollout
• Rollback
SUSE Software stack abstraction layers

Key: You Manage  Infrastructure Manages

<table>
<thead>
<tr>
<th>SUSE SLES</th>
<th>SUSE OpenStack</th>
<th>SUSE container platform</th>
<th>SUSE Cloud Application platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>You Manage Everything</td>
<td>IaaS</td>
<td>CaaS</td>
<td>PaaS</td>
</tr>
<tr>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
<td>Applications</td>
</tr>
<tr>
<td>Data</td>
<td>Data</td>
<td>Data</td>
<td>Data</td>
</tr>
<tr>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
<td>Runtime</td>
</tr>
<tr>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
<td>Middleware</td>
</tr>
<tr>
<td>Operating System</td>
<td>Operating System</td>
<td>Operating System</td>
<td>Operating System</td>
</tr>
<tr>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
<td>Virtualization</td>
</tr>
<tr>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
<td>Servers</td>
</tr>
<tr>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
<td>Storage</td>
</tr>
<tr>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
<td>Networking</td>
</tr>
</tbody>
</table>
Application example

Kubectl :( 

```
<table>
<thead>
<tr>
<th>PHP Frontend</th>
<th>PHP Frontend</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redis slave</td>
<td>Redis slave</td>
</tr>
<tr>
<td></td>
<td>Redis</td>
</tr>
</tbody>
</table>
```
Package manager for k8s

Like zypper is for SUSE
Some definitions

Chart
A chart is a collection of files that describe a related set of Kubernetes resources.

Repository
A chart repository is an HTTP server that houses one or more packaged charts.

Release
A deployed chart have a version number.
Available Commands:

- completion: Generate autocompletions script for the specified shell (bash or zsh)
- create: Create a new chart with the given name
- delete: Delete a chart
- dependency: Manage a chart's dependencies
- fetch: Download a chart from a repository and (optionally) unpack it in local directory
- get: Download a named release
- history: Fetch release history
- home: Displays the location of HELM_HOME
- init: Initialize Helm on both client and server
- inspect: Inspect a chart
- install: Install a chart's dependencies
- list: List installed charts
- list-releases: List available releases
- package: Package a chart directory into a chart archive
- plugin: Add, list, or remove Helm plugins
- repo: Add, list, remove, update, and index chart repositories
- reset: Uninstall Tiller from a cluster
- rollback: Roll back a release to a previous revision
- search: Search for a keyword in charts
- serve: Start a local HTTP web server
- status: Displays the status of the named release
- template: Render a template
- test: Test a release
- upgrade: Upgrade a release
- verify: Verify that a chart at the given path has been signed and is valid
- version: Print the client/server version information

~ $ helm search nginx

<table>
<thead>
<tr>
<th>NAME</th>
<th>CHART VERSION</th>
<th>APP VERSION</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>stable/nginx-ingress</td>
<td>1.1.4</td>
<td>0.21.0</td>
<td>An nginx Ingress controller that uses ConfigMap...</td>
</tr>
<tr>
<td>stable/nginx-ldapauth-proxy</td>
<td>0.1.2</td>
<td>1.13.5</td>
<td>nginx proxy with ldapauth</td>
</tr>
<tr>
<td>stable/nginx-lego</td>
<td>0.3.1</td>
<td></td>
<td>Chart for nginx-ingress-controller and kube-lego</td>
</tr>
<tr>
<td>stable/gcloud-endpoints</td>
<td>0.1.2</td>
<td>1</td>
<td>DEPRECATED Develop, deploy, protect and monitor...</td>
</tr>
</tbody>
</table>
But why? there’s kubectl

- Easier testing
- Rollback
- Portability
- ...
- ...
HELM

helm
The client

tiller
The server
### Initial CaaS Platform Configuration

#### Generic settings

**Internal Dashboard Location**

admin.suse.lab

#### Cluster services

- **Install Tiller ( Helm server component)**

#### Overlay network settings

#### Proxy settings

Enable Disable

#### SUSE registry mirror

Enable Disable

#### Container runtime

The choice of container runtime is completely transparent to end-users of the cluster. Neither Kubernetes manifests nor container images need to be changed.

**Choose the runtime**

- **Docker open source engine (default)**

Docker open source engine (default) is a production-ready runtime, fully supported by SUSE.
Application example

The HELM way :)

```
PHP Frontend
  ↓
Redis slave
  ↓
Redis
```

```
PHP Frontend
  ↓
Redis slave
  ↓
Redis
```
wrap-up

Helm:

● helps us define complete application stacks and dependencies.
● uses templates.
● uses values from different inputs.
● helps us to build easier CI/CD pipelines