Helm CHEAT SHEET (intro)

Overview

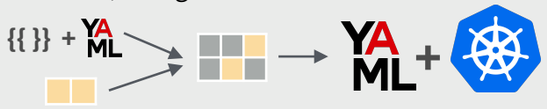
WHAT IS HELM?

A client-side package manager for Kubernetes.

The CLI is called **helm**. It uses "helm charts" from "helm repositories"

Helm charts help package, manage, and maintain resources in Kubernetes.

A helm chart is a **tar.gz** of Go-Templating and YAML files, along with version metadata



Parameters are called values and are injected by a **values file** or a **--set** argument

Set Up

INSTALLATION OPTIONS

1. You can download binaries directly from **https://github.com/helm/helm/releases**

2. **System installation** uses brew, choco, apt or some other package manager to install helm

3. There is an install-helper script and more options available on the helm website
https://helm.sh/docs/intro/install/

Find a Repository

artifacthub.io is a centralized location where many charts (from many repos) are organized

Software vendors often maintain their own helm charts on **GitHub** or **GitLab** (look for "helm" or "charts" git repositories)

Once you find a helm repository with charts:

```
helm repo add name myurl.com
```

local name, just for you

The url to the repository

Prepare a Chart

```
helm repo list
helm search repo pattern
```

Optional wildcard search

```
helm search repo --devel
helm search repo -l
```

Include dev versions

List all versions (not just latest)

Once you have found a chart, you need to prepare your values

DISCOVER VALUES

"Values" are the available parameters for a chart

There are conventions, but no standards for values / parameters, so each chart can vary

```
helm show values name/chart
```

DEFINE VALUES - OPTIONS

1. Arbitrary yaml files, using the same object hierarchy as **helm show values**

some-file.yaml

```
pod:
  key:
    - "some"
    - "text"
```

Variable types matter and are interpreted according to YAML conventions

2. The **--set** argument allows setting values with shell semantics (i.e. read environment variables)

```
… --set pod.key[1]="value"
```

3. Chart default values are included

4. Unset default values with **null**

PREVIEW THE OUTPUT

helm template is used to preview chart output

```
helm template \
  release-name name/chart \
  -f some-file.yaml \
  --set key=value
```

Some semantics for installation will differ from template output
- Kubernetes version specific behavior
- Helm's **lookup** internal function

Requires a Kubernetes Cluster

Explore Releases

See existing helm releases in a namespace

```
helm list -n namespace
```

Or all namespaces

```
helm list -A
```

If curious about a particular release and its history or configuration:

```
helm history release-name
helm get values release-name
helm get all release-name
helm status release-name
```

Or investigate created resources directly with **kubectl**. **Annotations** and **labels** will show which helm release created a resource

Modify a Release

Syntax is identical for templating, first install, and "upgrades"

```
helm install \
  release-name name/chart
helm upgrade \
  release-name name/chart
```

It is possible to have an idempotent command:

```
helm upgrade --install \
  release-name name/chart
```

MOST USED COMMAND

The **--atomic** argument will automatically roll-back on failure

Rollback can take you back to a previous chart release

```
helm rollback release-name
```

To remove a chart release:
CAUTION: will delete resources in cluster

```
helm uninstall release-name
```

Helm Plugins

Helm plugins are *extensions* of the helm CLI

They add functionality to the helm CLI, but are maintained independently of the helm project

Plugin Name	Description
helm-diff	View diff between releases
helm-git	Use a git branch or repo as a helm repo
helm-s3	Use an s3 bucket directly as a helm repo
helm-gcs	Use Google Cloud Storage as a helm repo
helm-secrets	Plugin for secret management
helm-unittest	Plugin for unit testing charts

And many more!! Check out:
https://helm.sh/docs/community/related/

Useful Tips

You can use "**local charts**," (i.e. charts that are just files on disk) by using filesystem paths

```
helm template \
  release-name ./path/to/chart
helm template \
  release-name /abs/path/too
```

In production environments, *be sure* to pin the version of the chart using **--version**

```
… --version=0.1.0
```

VERY important

ADVANCED --set SYNTAX

```
… --set key.val="text"
… --set somekey=null
… --set "key\.dot=$ENV_VAR"
… --set "key*star"="val*"
… --set "array={1,2,3}"
… --set array[0].key=value
```

DETECT NEW CHART VERSIONS

```
helm repo update
```