LZone

Helm Cheat Sheet

Edit Cheat Sheet

This cheat sheet is organized as per the typical workflow for Helm3

```
1) Add a chart repository.
2) Find a chart to install.
3) Install a Helm chart.
4) See the list of what is installed.
5) Upgrade your installation.
6) Delete the installation.
```

Add, remove, list, search and Update Repos

```
helm repo add [name] [url] #Add a repository from the internet
helm repo remove [name] #Remove a repository from your system
helm repo update #Update repositories
helm repo list #List chart repositories
helm repo index #Generate an index file containing charts found in the current directory
helm search [keyword] #Search charts for a keyword
helm search repo [keyword] #Search repositories for a keyword
helm search hub [keyword] #Search Helm Hub
```

List available packages

To list charts (packages)

```
helm search hub wordpress #helm search hub searches the Artifact Hub
helm search repo wordpress #helm search repo searches the repositories that you have addec
```

Install and Uninstall Apps

```
helm install [name] [chart] --namespace [namespace] #Install an app in a specific namespac
helm install [name] [chart] --values [yaml-file/url] #Override the default values with thc
helm install [name] --dry-run --debug #Run a test install to validate and verify the chart
helm uninstall [release name]  #Uninstall a release
```

Chart Management

```
helm create [name] #Create a directory containing (Chart.yaml, values.yaml,charts/ and tem
helm package [chart-path] #Package a chart into a chart archive
helm lint [chart] #Run tests to examine a chart and identify possible issues
helm show all [chart] #Inspect a chart and list its contents
helm show chart [chart] #Display the chart's definition
helm show values [chart] #Display the chart's values
helm pull [chart] #Download a chart
helm pull [chart] --untar --untardir [directory] #Download a chart and extract the archive
helm dependency list [chart] #Display a list of a chart's dependencies
helm install mychart-0.1.0.tgz --dry-run --debug #Test installation
```

Helm Plugin Management

```
helm plugin list #view a list of all the installed plugins
helm plugin install <plugin URL> #Install Plugins
helm plugin update [plugin1] [plugin2] #update plugin
helm plugin uninstall [plugin1] #Uninstall a plugin
```

Setup Docker Registry as Helm Chart Repo

```
export HELM_EXPERIMENTAL_OCI=1
```

Upgrading ,Rollback and list releases

```
helm upgrade [release] [chart] #Upgrade an app
```

```
helm upgrade [release] [chart] --atomic #Tell Helm to roll back changes if the upgrade fai
helm upgrade [release] [chart] --install #Upgrade a release. If it does not exist on the s
helm upgrade [release] [chart] --version [version-number] #Upgrade to a version other thar
helm rollback [release] [revision] #Roll back a release
helm upgrade --wait <name> # Wait for pods to come up
helm ls #List releases in current namespace
helm ls -A #List all releases in all namespaces
helm ls -A -o json | jq  -r '.[] | select(.status = "deployed") | .name' #Find releases ir
helm get values <release> # Print the values the release was installed with
```

Download Release Information

```
helm get all [release] #Download all the release information
helm get hooks [release] #Download all hooks
helm get manifest [release] #Download the manifest
helm get notes [release] #Download the notes
helm get values [release] #Download the values file
helm history [release] #Fetch release history
```

##Get help and Version Info

```
helm --help #See the general help for Helm
helm [command] #help #See help for a particular command
helm version #See the installed version of Helm
```

##Release Monitoring

```
helm list #List all the available releases in the current namespace
helm list --all-namespaces #List all the available releases across all namespaces
helm list --namespace [namespace] #List all the releases in a specific namespace
helm list --output [format] #List all the releases in a specific output format
helm list --filter ‘[expression]’ #Apply a filter to the list of releases using regular (F
helm status [release] #See the status of a release
helm history [release] #See the release history
helm env #See information about the Helm client environment
```

Using Helm in Terraform

Dependency in main.tf

```
terraform {
  required_providers {
    helm = {
      version = ">= 3.5"
    }
  }
}
```

Declaring Helm ressources with terraform

```
resource "helm_release" "myreleasename" {
  name = "myreleasename"
  namespace = "<namespace>"
  create_namespace = <true|false>
  chart = "<path to local chart|chart name>"
}
```

Best Practices

See https://helm.sh/docs/chart_best_practices/conventions/

Misc

https://kubeapps.com/: Web GUI for installing Helm charts

https://artifacthub.io/ : To find helm charts

Infrastructure as Code solutions for Helm:

- helmfile
- Helmsman
- Terraform

Solutions to host helm charts yourself:

- chartmuseum
- harbour

Nexus

jFrog Artifactory

any Docker registry (experimental feature)

Related Helm Templates kubernetes Openshift