# Helm

# Kubernetes Installation

[root@ip- kubernetes] kubectl get nodes

NAME STATUS ROLES AGE VERSION

ip- xxxxxxx Ready master 15m v1.18.0

ip-xxxxxxx Ready <none> 10m v1.18.0

**Install Helm**

[root@ip- kubernetes] mkdir helm

[root@ip- kubernetes] cd helm

[root@ip- kubernetes] curl -fsSL -o get\_helm.sh <https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3>

[root@ip- kubernetes] chmod 700 get\_helm.sh

[root@ip- kubernetes] ./get\_helm.sh

[root@ip- kubernetes] helm help

Expected output

The Kubernetes package manager

Common actions for Helm:

- helm search: search for charts

- helm pull: download a chart to your local directory to view

- helm install: upload the chart to Kubernetes

- helm list: list releases of charts

**Useful Helm commands**

***helm search hub***Find publicly available charts.

For exp:- helm search hub wordpress

***helm search repo*** *Searches the repositories that you have added to your local helm client*

For Exp:- helm repo add brigade <https://brigadecore.github.io/charts>

helm search repo brigade

***helm repo list*** To get the list of repositories configured in Helm

***helm install*** To install a helm package

For exp:- helm install mydb stable/mariadb

***helm status*** To check the status of a release state

For Exp:- helm status mydb

***helm upgrade*** Takes an existing release and upgrades it according to the information

you provide

For Exp:- helm upgrade –f myvalues.yaml mydb stable/mariadb

***helm rollback*** To roll back to a previous release

For Exp:- helm rollback mydb <version #>

**Use Helm to deploy a chart**

Initialize a Helm chart repository

$ helm repo add stable <https://kubernetes-charts.storage.googleapis.com/>

$ helm repo add my-charts https://my-charts.storage.googleapis.com

$ helm repo update

$ helm repo list

$ helm search repo mysql

$ helm show values stable/mysql

$ helm install mysqldb stable/mysql

Install it in a Kubernetes namespace

$ helm install mysqldb --namespace db-system stable/mysql

To pass values

$ helm install --set user.name='student',user.password='passw0rd' stable/mysql

$ helm install --values myvalues.yaml stable/mysql

$ helm upgrade

$ helm uninstall mysqldb

**Create a Helm chart**

Use the helm create command to scaffold out an example

[root@ip- kubernetes] helm create hostname-api

└── hostname-api

├── charts

├── Chart.yaml

├── templates

│   ├── deployment.yaml

│   ├── \_helpers.tpl

│   ├── hpa.yaml

│   ├── ingress.yaml

│   ├── NOTES.txt

│   ├── serviceaccount.yaml

│   ├── service.yaml

│   └── tests

│   └── test-connection.yaml

└── values.yaml

**Templates**

# The template directory holds the YAML definitions for your Services, Deployments and other Kubernetes objects. If you already have services and deployment yaml definitions, you can replace them with the entries in templates.

**Values**

# The template in service.yaml makes use of the Helm-specific objects .Chart and .Values.

# Chart.yaml provides metadata about the chart to your definitions such as the name, or version.

# Values.yaml is used to expose configuration that can be set at the time of deployment. The defaults for this object are defined in the values.yaml file.

[root@ip- kubernetes] helm install --dry-run --debug ./hostname-api --set service.internalPort=8080

# Deploy your first Chart

helm install myhostname-api ./hostname-api --set service.type=NodePort

helm upgrade myhostname-api ./hostname-api --set service.type=NodePort

helm upgrade myhostname-api ./hostname-api --set service.type=NodePort,deployment.replicaCount=2

helm uninstall myhostname-api

**Packages the chart**

helm package ./hostname-api

Helm will create a hostname-api-0.1.0.tgz package in our working directory, using the name and version from the metadata defined in the Chart.yaml file. A user can install from this package instead of a local directory by passing the package as the parameter to helm install.

helm install mywebetest hostname-api-0.1.0.tgz --set service.type=NodePort

# helm serve ( To run the helm local repository)

# 

# helm search local

# To Add dependencies

cat > ./mywebapp/requirements.yaml <<EOF

dependencies:

- name: mariadb

version: 0.6.0

repository: https://kubernetes-charts.storage.googleapis.com

EOF

# helm dep update ./mywebapp

# 