Kubernets Helm

**Helm** is a package manager for **Kubernetes** that allows developers and operators to more easily package, configure, and deploy applications and services onto **Kubernetes** clusters. ... In this article we will give an overview of **Helm** and the various abstractions it uses to simplify deploying applications to **Kubernetes**.

Helm Installation

Reference URL : - https://designisdead.com/blog/discovering-helm-a-package-manager-for-kubernetes

Download the helm script

$ curl https://raw.githubusercontent.com/helm/helm/master/scripts/get-helm-3 > get\_helm.sh

Make sure you can execute the script

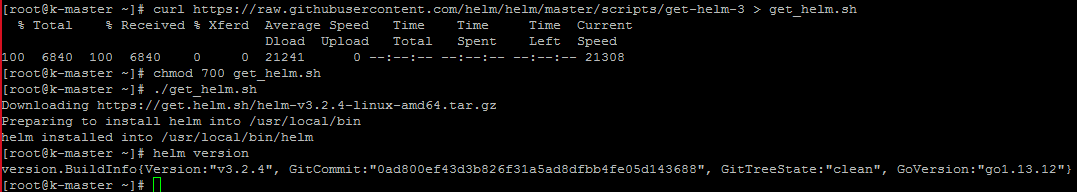
$ chmod 700 get\_helm.sh

Run the install script

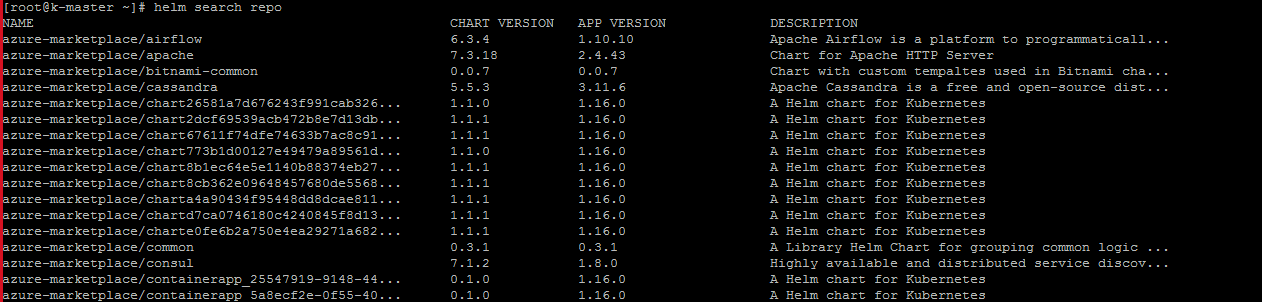
$ ./get\_helm.sh

Check if it’s working

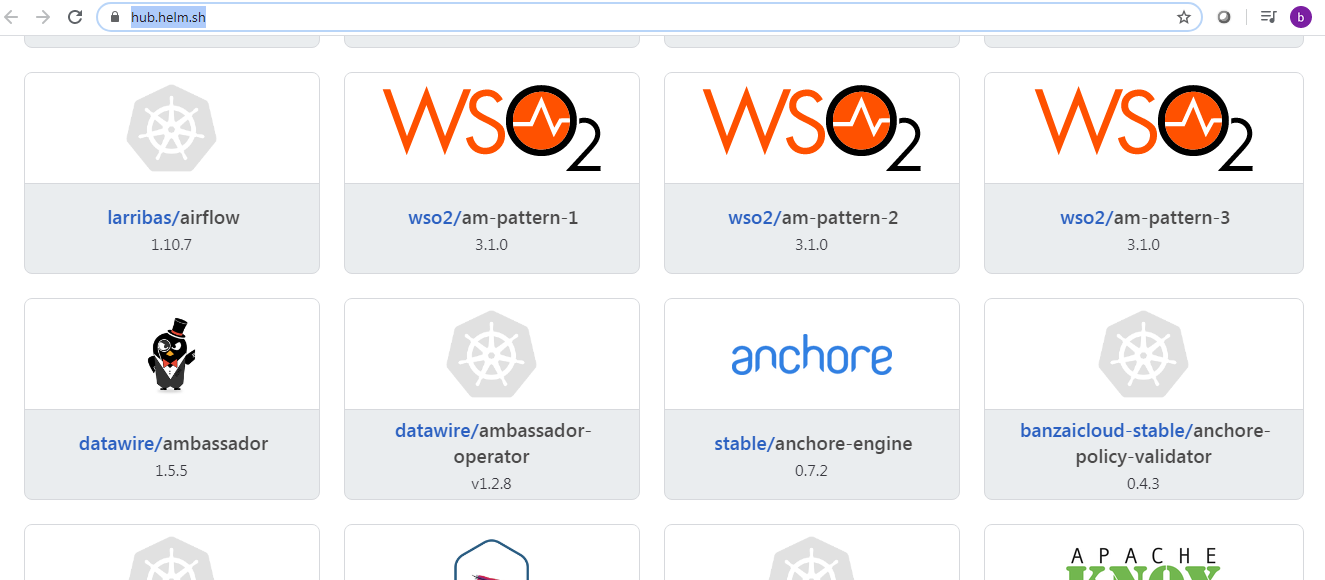
$ helm version



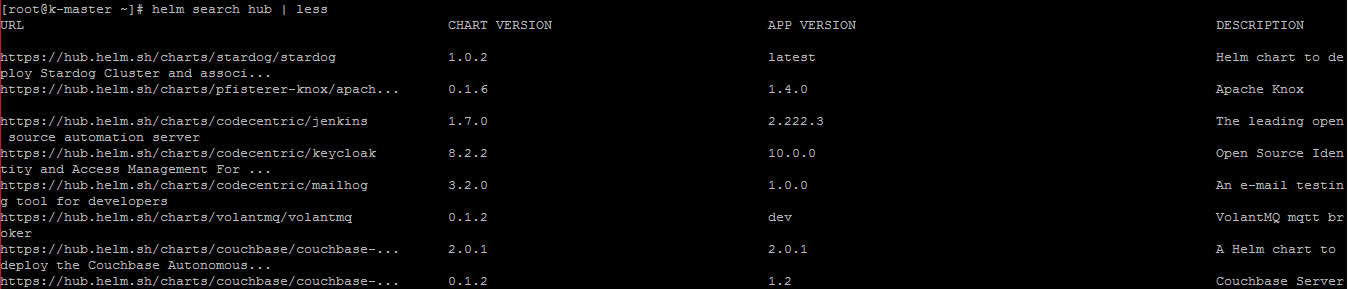
**helm search repo**



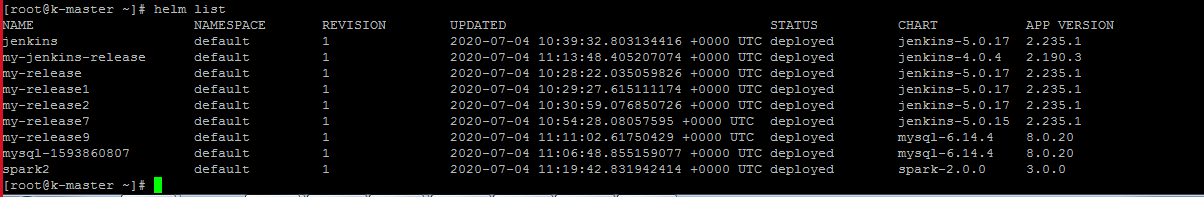
<https://hub.helm.sh/>



helm search hub | less



helm list



Cheat Sheet: <http://linuxroutes.com/helm-commands-cheat-sheet/>

**helm help**

[root@k-master ~]# helm help

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

Environment variables:

| Name | Description |

|------------------------------------|-----------------------------------------------------------------------------------|

| $XDG\_CACHE\_HOME | set an alternative location for storing cached files. |

| $XDG\_CONFIG\_HOME | set an alternative location for storing Helm configuration. |

| $XDG\_DATA\_HOME | set an alternative location for storing Helm data. |

| $HELM\_DRIVER | set the backend storage driver. Values are: configmap, secret, memory, postgres |

| $HELM\_DRIVER\_SQL\_CONNECTION\_STRING | set the connection string the SQL storage driver should use. |

| $HELM\_NO\_PLUGINS | disable plugins. Set HELM\_NO\_PLUGINS=1 to disable plugins. |

| $KUBECONFIG | set an alternative Kubernetes configuration file (default "~/.kube/config") |

Helm stores configuration based on the XDG base directory specification, so

- cached files are stored in $XDG\_CACHE\_HOME/helm

- configuration is stored in $XDG\_CONFIG\_HOME/helm

- data is stored in $XDG\_DATA\_HOME/helm

By default, the default directories depend on the Operating System. The defaults are listed below:

| Operating System | Cache Path | Configuration Path | Data Path |

|------------------|---------------------------|--------------------------------|-------------------------|

| Linux | $HOME/.cache/helm | $HOME/.config/helm | $HOME/.local/share/helm |

| macOS | $HOME/Library/Caches/helm | $HOME/Library/Preferences/helm | $HOME/Library/helm |

| Windows | %TEMP%\helm | %APPDATA%\helm | %APPDATA%\helm |

Usage:

helm [command]

Available Commands:

completion generate autocompletions script for the specified shell (bash or zsh)

create create a new chart with the given name

dependency manage a chart's dependencies

env helm client environment information

get download extended information of a named release

help Help about any command

history fetch release history

install install a chart

lint examine a chart for possible issues

list list releases

package package a chart directory into a chart archive

plugin install, list, or uninstall Helm plugins

pull download a chart from a repository and (optionally) unpack it in local directory

repo add, list, remove, update, and index chart repositories

rollback roll back a release to a previous revision

search search for a keyword in charts

show show information of a chart

status display the status of the named release

template locally render templates

test run tests for a release

uninstall uninstall 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 version information

Flags:

--add-dir-header If true, adds the file directory to the header

--alsologtostderr log to standard error as well as files

--debug enable verbose output

-h, --help help for helm

--kube-apiserver string the address and the port for the Kubernetes API server

--kube-context string name of the kubeconfig context to use

--kube-token string bearer token used for authentication

--kubeconfig string path to the kubeconfig file

--log-backtrace-at traceLocation when logging hits line file:N, emit a stack trace (default :0)

--log-dir string If non-empty, write log files in this directory

--log-file string If non-empty, use this log file

--log-file-max-size uint Defines the maximum size a log file can grow to. Unit is megabytes. If the value is 0, the maximum file size is unlimited. (default 1800)

--logtostderr log to standard error instead of files (default true)

-n, --namespace string namespace scope for this request

--registry-config string path to the registry config file (default "/root/.config/helm/registry.json")

--repository-cache string path to the file containing cached repository indexes (default "/root/.cache/helm/repository")

--repository-config string path to the file containing repository names and URLs (default "/root/.config/helm/repositories.yaml")

--skip-headers If true, avoid header prefixes in the log messages

--skip-log-headers If true, avoid headers when opening log files

--stderrthreshold severity logs at or above this threshold go to stderr (default 2)

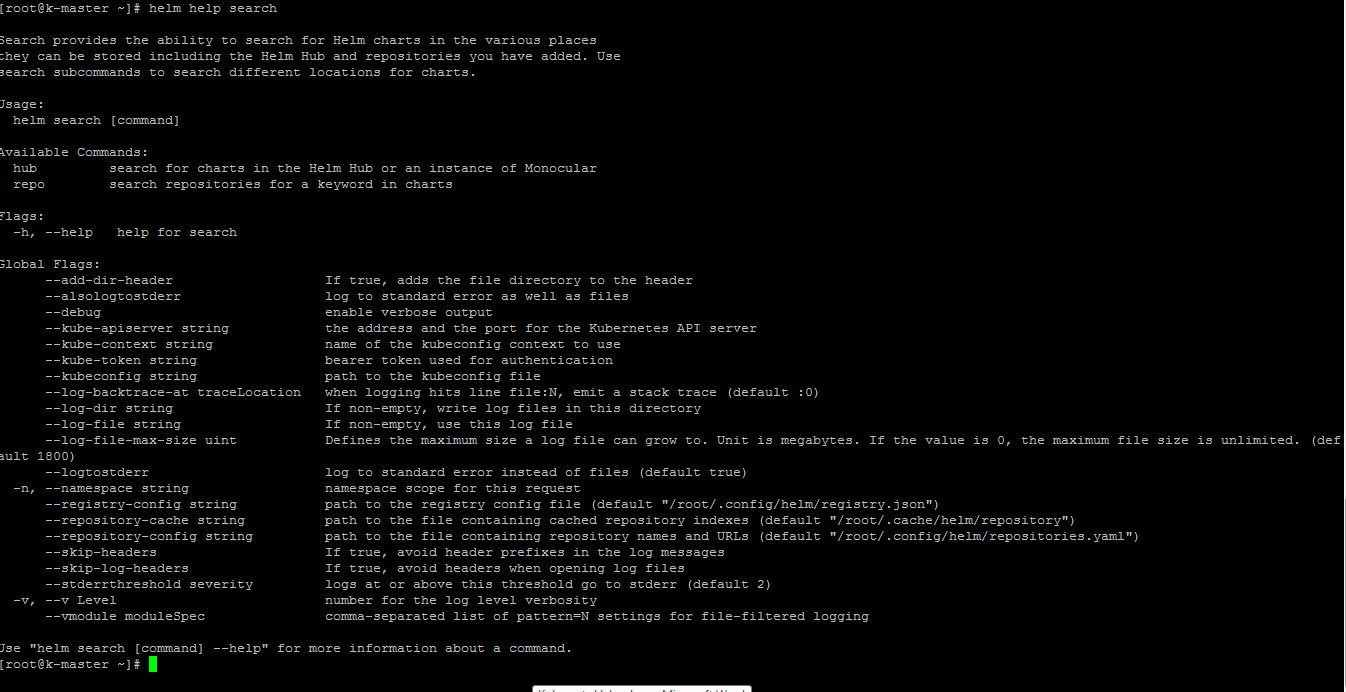
-v, --v Level number for the log level verbosity

--vmodule moduleSpec comma-separated list of pattern=N settings for file-filtered logging

Use "helm [command] --help" for more information about a command.

[root@k-master ~]#

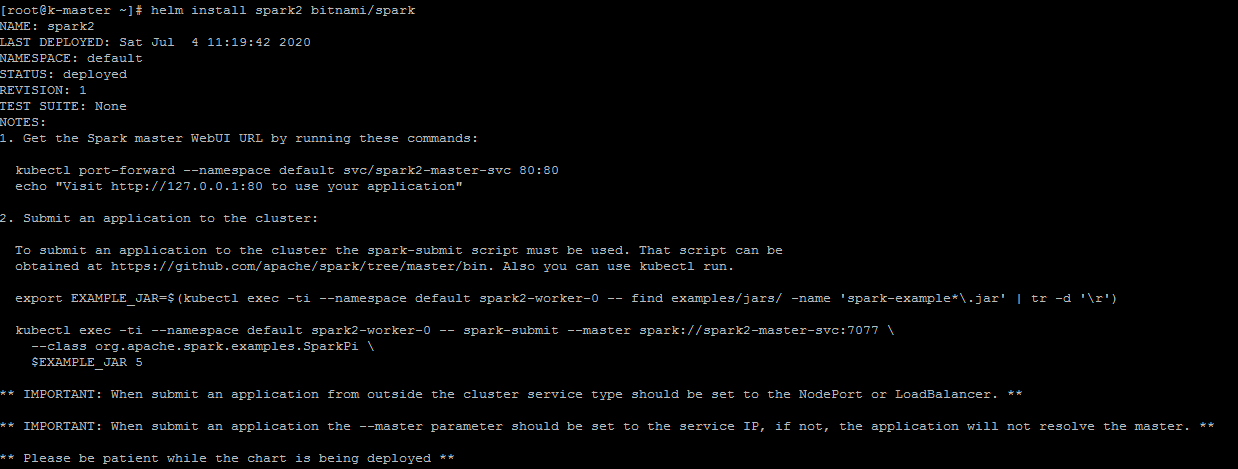
**helm help search**

****

**Spark installation**

helm repo add bitnami <https://charts.bitnami.com/bitnami>

helm install spark2 bitnami/spark



kubectl port-forward --namespace default svc/spark2-master-svc 80:80



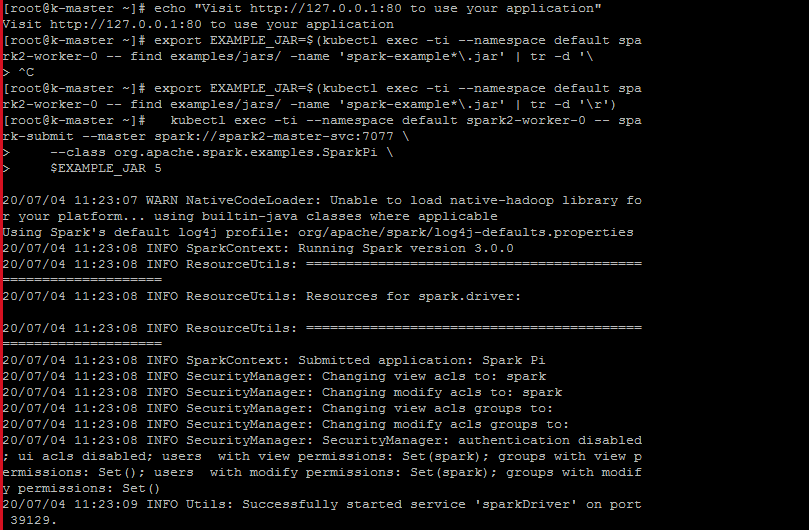
Open new terminal and run below command

echo "Visit http://127.0.0.1:80 to use your application

export EXAMPLE\_JAR=$(kubectl exec -ti --namespace default spark2-worker-0 -- find examples/jars/ -name 'spark-example\*\.jar' | tr -d '\r')

kubectl exec -ti --namespace default spark2-worker-0 -- spark-submit --master spark://spark2-master-svc:7077 --class org.apache.spark.examples.SparkPi \

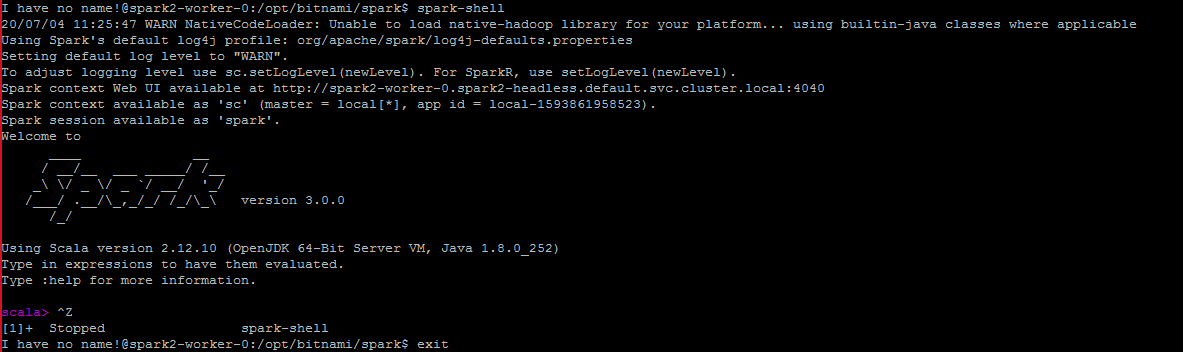
$EXAMPLE\_JAR 5

****

kubectl exec -ti --namespace default spark2-worker-0 bash

****

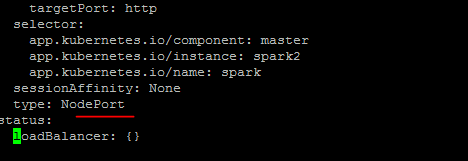
spark-shell

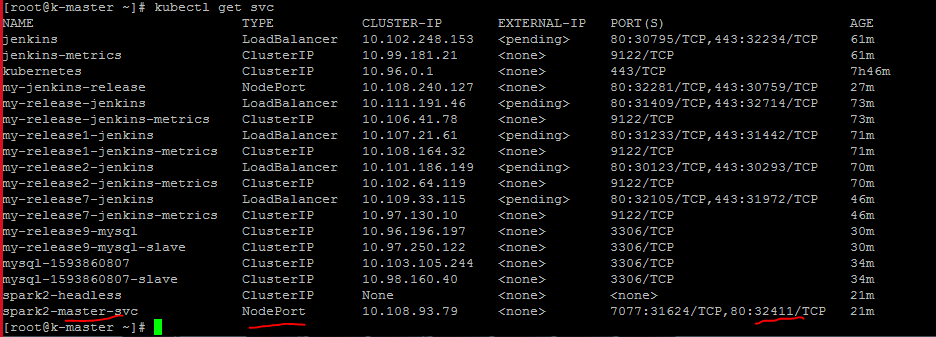


kubectl edit svc spark2-master-svc

Changed

ClusterType to NodePort





<http://34.72.106.152:32411/>

