```
Helm commands Ref:  https://helm.sh/docs/intro/cheatsheet/

============================================================

Lab: 1 Install Helm in kubernetes Cluster
=========================================

wget https://get.helm.sh/helm-v3.5.0-rc.2-linux-amd64.tar.gz

tar -xzvf  helm-v3.5.0-rc.2-linux-amd64.tar.gz

mv linux-amd64/helm /bin

helm version


Helm Commands
# this will print helm help
helm

# to see all available package in hub
helm search hub

# to list mysql or nginx charts
helm search hub mysql

# to list all available repo
helm repo list

# add helm repo named bitnami
helm repo add bitnami https://charts.bitnami.com/bitnami

# search bitnami named repo in server
helm search repo bitnami

#update repo
helm repo update

# Search chart in specific repo
helm search repo bitnami/nginx
helm search repo â€“l bitnami/mysql   #list all available versions
helm search repo nginx --versions


# To remove the repo
helm repo remove bitnami
helm repo list


# add again helm repo named bitnami
helm repo add bitnami https://charts.bitnami.com/bitnami


# install nginx using chart
helm install mywebserver bitnami/nginx

# install in specific namesape then first create name space then install
kubectl create ns dev
helm install -n dev  mywebserver bitnami/nginx
# helm show default name space by default
helm list
# to check in the specific name space
helm list -n dev

# delete nginx
helm uninstall mywebserver

# delete from specific name space
```

```
helm uninstall mywebserver -n dev
helm list -n dev
helm list
kubectl get pod


# update the repo
helm install mywebserver bitnami/nginx
helm list
helm repo update
helm list
helm status mywebserver
helm upgrade mywebserver bitnami/nginx          # --reuse-values
helm status mywebserver
helm list



Lab: 2 Chart Structure
=====================================

apt update && apt install tree -y

helm create <Chart-Name>        # martuj-mychart


controlplane $ tree mychart/
mychart/
|-- Chart.yaml
|-- charts
|-- templates
|   |-- NOTES.txt
|   |-- _helpers.tpl
|   |-- deployment.yaml
|   |-- hpa.yaml
|   |-- ingress.yaml
|   |-- service.yaml
|   |-- serviceaccount.yaml
|   `-- tests
|       `-- test-connection.yaml
`-- values.yaml


Dry- Run
helm install --debug --dry-run mychart ./mychart/
helm install mychart ./mychart/
helm get manifest mychart




Lab: 3 Create Custom Chart
==========================

mkdir -p myapp/charts && mkdir -p myapp/templates

cd myapp

vi Chart.yaml
apiVersion: v2
name: mychart
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"


#create empty file
touch values.yaml
```

```
vi templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mydep
spec:
  replicas: 1
  selector:
    matchLabels:
      app: myapp
  template:
    metadata:
      labels:
        app: myapp
    spec:
      containers:
      - image: nginx
        name: nginx
        ports:
        - containerPort: 80



vi templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: myapp-svc
spec:
  selector:
    app: myapp
  ports:
  - protocol: TCP
    port: 80
  type: NodePort



$ tree
.
|-- Chart.yaml
|-- charts
|-- templates
|   |-- deployment.yaml
|   `-- service.yaml
`-- values.yaml

#type cd to comeout from the dir
cd
helm install mychart1 ./myapp
kubectl get deployment
kubectl get pods

helm uninstall mychart1



Lab: 4  Create custom template
========================

mkdir -p myapp1/charts && mkdir -p myapp1/templates

cd myapp1

vi Chart.yaml
apiVersion: v2
name: mychart
```

```
description: A Helm chart for Kubernetes
type: application
version: 0.1.0
appVersion: "1.16.0"



vi values.yaml
replicaCount: 1
app:
  name: myapp
  image: nginx
  ports:
    containerPort: 80



vi templates/deployment.yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: {{ .Release.Name }}
spec:
  replicas: {{ .Values.replicaCount }}
  selector:
    matchLabels:
      {{- include "myapp.applabels" . | nindent 7 }}
  template:
    metadata:
      labels:
        {{- include "myapp.applabels" . | nindent 9 }}
    spec:
      containers:
      - image: {{ .Values.app.image }}
        name: {{ .Values.app.name }}
        ports:
        - containerPort: {{ .Values.app.ports.containerPort }}


vi templates/service.yaml
apiVersion: v1
kind: Service
metadata:
  name: myapp-svc
spec:
  selector:
    app: myapp
  ports:
  - protocol: TCP
    port: 80
  type: NodePort




vi templates/_helpers.tpl
{{- define "myapp.applabels" -}}
app: myapp
{{- end }}



vi templates/NOTES.txt
Thank you for support {{ .Chart.Name }}.

 Your release is named {{ .Release.Name }}.

 To learn more about the release, try:
```

```
    $ helm status {{ .Release.Name }}
    $ helm get all {{ .Release.Name }}
    $ helm uninstall {{ .Release.Name }}
```

```
$ tree
.
|-- Chart.yaml
|-- charts
|-- templates
|   |-- deployment.yaml
|   `-- service.yaml
`-- values.yaml

#come out from dir
cd
helm install mychart2 ./myapp

helm uninstall mychart2
```

```
Lab: 5 Create package using chart
=====================================

helm package mychart
ls


Store Charts in chart repo
mkdir charts
chmod 777 charts

# Install docker

apt update -y
apt install curl -y
curl -SSL https://get.docker.com/ | sh
service docker status
docker --version

# rundocker container

docker run -itd \
  -p 8080:8080 \
  -e DEBUG=1 \
  -e STORAGE=local \
  -e STORAGE_LOCAL_ROOTDIR=/charts \
  -v $(pwd)/charts:/charts \
  ghcr.io/helm/chartmuseum:v0.14.0


#check the container
docker ps

helm repo ls

#add repo locally

helm repo add mychartrepo http://localhost:8080

helm repo ls

helm repo update

# to see all available package in hub but no result now
```

```
helm search repo mychartrepo



#install package from this repo
helm install myapp mychartrepo/mychart


#upload helm chart into repo
curl --data-binary "@mychart-0.1.0.tgz" http://localhost:8080/api/charts

helm repo update
helm repo ls
helm search repo mychartrepo

#install package from this repo
helm install myapp mychartrepo/mychart
helm ls


#Create new version of chart  modify Chart.yaml's version

vi mychart/Chart.yaml           #change the versions

# now create
helm package mychart            #(dir name)
ls
# now you can see 2.0 zip file

#upload new version chart  into repo
curl --data-binary "@mychart-0.2.0.tgz" http://localhost:8080/api/charts

helm repo update

#list helm chart in repo

helm search repo mychartrepo -l

# Upgrade Helm installed Chart
helm upgrade myapp mychartrepo/mychart
helm ls

helm history myapp

helm rollback myapp 1
helm history myappp

#delete repo
 helm repo remove mychartrepo
 helm repo ls
```