Run Helm Manager in kubernetes

Introduction

The helm manager is a service for managing helm charts. The service provides a REST API for onboarding of charts as well as installation and uninstallation of applications based on these charts.

In addition to the helm manager, a chart repository is used to store the available charts for onboarding and installation.

The Helm manager (and chart repo) can be executed in one of the following deployments:

- As kubernetes service and pod on a local machine with kubernetes or in minikube
- As kubernetes service and pod on a kubernetes cluster
- As docker container on a local machine with kubernbetes (see Release E - Run in Docker)

Prerequisites

The following need to be installed, latest version, on the machine

- Kubernetes or minikube including kubectl
- curl
- git (for downloading of the repo
- docker (image registry)
- helm

Preparation

Clone the nonrtric repo from gerrit and change dir to *helm-manager*. Make sure to use the correct branch when cloning, use "master" until the branch for E Release is available.

All needed scripts for starting and running the helm manager, in docker or kubernetes are in this directory.

If the Helm Manager shall be installed in a kubernetes cluster the actions below shall be made on a node in cluster.

```
$ git clone "https://gerrit.o-ran-sc.org/r/nonrtric" -b e-release

$ cd helm-manager
```

Create helm chart for test

Create a helm chart for this test. Package the chart into an archive. Run these two commands. The result should be a file named *simple-app-0.1.0.tgz*.

```
$ helm create simple-app

$ helm package simple-app
```

Run in kubernetes

This instruction is valid for running both in a local kubernetes or in a kubernetes ccluster

If running in a local kubernetes, set the env KUBE_HOST to the host of the kubernetes control plane

```
$ kubectl cluster-info
Kubernetes control plane is running at https://kubernetes.docker.internal:6443
$ KUBE_HOST="kubernetes.docker.internal"
```

If running in a kubernetes cluster or in minikube, set the env KUBE_HOST to the ip of the kubernetes control plane.

Example

```
$ kubectl cluster-info
Kubernetes master is running at https://10.2.0.103:6443
$ KUBE_HOST=10.2.0.103
```

Check if the nonrtric names space exists. If not, create the namespace

```
$ kubectl get ns nonrtric
$ kubectl create ns nonrtric
```

Start the chartmuseum service and pod

```
$ kubectl apply -f kube-cm.yaml
```

Add the chart, created in the section 'Create helm for test', to the chartmuseum repo. The node port of the chartmuseum service is obtained and env var CM_PORT is assigned that port number.

```
$ CM_PORT=$(kubectl get svc chartrepo -n nonrtric -o jsonpath='{...ports[?(@.name=="'http'")].nodePort}')
$ curl --data-binary @simple-app-0.1.0.tgz -X POST http://$KUBE_HOST:$CM_PORT/api/charts
{"saved":true}
```

Create a service account for the helm manager. This example service account bind to the "cluster-admin" role which normally has full permissions to the add/change/read/delete any kubernetes object. It is advisable to bind the service account to a ClusterRole with less permissions if desired.

```
$ kubectl apply -f helm-manager-sa.yaml
serviceaccount/helm-manager-sa created
clusterrolebinding.rbac.authorization.k8s.io/helm-manager-sa-clusterrolebinding created
```

Start the helm manager. Four objects will be created. Note that the service is defined as a NodePort. This enables access from outside the cluster and is also a precondition for the test script to work. Change 'type' to 'ClusterIP' in the 'helmmanagerservice' service definition in the file helm-manager.yaml.

```
$ kubectl apply -f helm-manager.yaml
service/helmmanagerservice created
pod/helmmanagerservice created
persistentvolume/helm-manager-service-pv created
persistentvolumeclaim/helm-manager-service-pvc created
```

The chartmuseum repo need to added to helm. This operation must be called with a url accessible from the helm manager pod.

Go into the helm manager container and add the repo.

```
$ kubectl exec -it helmmanagerservice -n nonrtric -- helm repo add cm http://chartrepo.nonrtric:8080
"cm" has been added to your repositories
```

The helm manager is now running and configured with a chart repo.

Run the script test.sh to execute the sequence for installing the application 'simpleapp' namespace 'ckhm':

- Namespace 'ckhm' is created in kubernetes if not existing
- Onboard chart
- Install chart
- Uninstall chart
- Remove (the onboarded) chart

All operations should report "OK".

```
$ ./test.sh kube $KUBE_HOST
```

Example output of the script

```
Start test
================
Get apps - empty
================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/charts
 Curl OK
  Response: 200
  Body: {"charts":[]}

================
Add repo
================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/repo -X POST -H
Content-Type:application/json -d @cm-repo.json
 Curl OK
  Response: 201
  Body:

============
Onboard app
===========
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/onboard/chart -X POST -
F chart=@simple-app-0.1.0.tgz -F values=@simple-app-values.yaml -F info=<simple-app.json
 Curl OK
  Response: 200
  Body:

=====================
Get apps - simple-app
=====================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/charts
 Curl OK
  Response: 200
  Body: {"charts":[{"releaseName":"simpleapp","chartId":{"name":"simple-app","version":"0.1.0"},"namespace":"
ckhm","repository":{"repoName":"cm","protocol":null,"address":null,"port":null,"userName":null,"password":
null},"overrideParams":null}]}

===========
Install app
===========
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/install -X POST -H
Content-Type:application/json -d @simple-app-installation.json
 Curl OK
  Response: 201
  Body:

=====================
Get apps - simple-app
=====================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/charts
 Curl OK
  Response: 200
  Body: {"charts":[{"releaseName":"simpleapp","chartId":{"name":"simple-app","version":"0.1.0"},"namespace":"
ckhm","repository":{"repoName":"cm","protocol":null,"address":null,"port":null,"userName":null,"password":
null},"overrideParams":null}]}

=================================================================
helm ls to list installed app - simpleapp chart should be visible
=================================================================
NAME             NAMESPACE        REVISION        UPDATED
STATUS          CHART                  APP VERSION
simpleapp        ckhm             1               2021-12-14 08:10:33.785870875 +0000 UTC
deployed        simple-app-0.1.0        1.16.0
```

```
==========================================
sleep 30 - give the app some time to start
==========================================
============================
List svc and  pod of the app
============================
NAME                   TYPE        CLUSTER-IP      EXTERNAL-IP   PORT(S)   AGE
simpleapp-simple-app   ClusterIP   10.106.179.47   <none>        80/TCP    30s
NAME                                     READY   STATUS    RESTARTS   AGE
simpleapp-simple-app-675f44fc99-mr7lp   1/1     Running   0          30s

========================
Uninstall app simple-app
========================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/uninstall/simple-app/0.
1.0 -X DELETE
 Curl OK
  Response: 204
  Body:

==========================================
sleep 30 - give the app some time to remove
==========================================
============================================================
List svc and  pod of the app - should be gone or terminating
============================================================
No resources found in ckhm namespace.
No resources found in ckhm namespace.

=====================
Get apps - simple-app
=====================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/charts
 Curl OK
  Response: 200
  Body: {"charts":[{"releaseName":"simpleapp","chartId":{"name":"simple-app","version":"0.1.0"},"namespace":"
ckhm","repository":{"repoName":"cm","protocol":null,"address":null,"port":null,"userName":null,"password":
null},"overrideParams":null}]}

============
Delete chart
============
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/chart/simple-app/0.1.0
-X DELETE
 Curl OK
  Response: 204
  Body:

================
Get apps - empty
================
curl -sw %{http_code} http://helmadmin:itisasecret@kubernetes.docker.internal:30712/helm/charts
 Curl OK
  Response: 200
  Body: {"charts":[]}

Test result  All tests ok
End of test
```

Cleanup of all created kubernetes object

```
$ kubectl delete -f helm-manager.yaml
service "helmmanagerservice" deleted
pod "helmmanagerservice" deleted
persistentvolume "helm-manager-service-pv" deleted
persistentvolumeclaim "helm-manager-service-pvc" deleted

$ kubectl delete -f kube-cm.yaml
service "chartrepo" deleted
pod "chartrepo" deleted
persistentvolume "chartrepo-pv" deleted
persistentvolumeclaim "chartrepo-pvc" deleted

$ kubectl delete -f helm-manager-sa.yaml
serviceaccount "helm-manager-sa" deleted
clusterrolebinding.rbac.authorization.k8s.io "helm-manager-sa-clusterrolebinding" deleted
```