OSM-MR#10 Hackfest – Day 2
Session 4. Orchestrating a CNF with Helm Charts - OpenLDAP

Gerardo García (Telefónica)
Context of this session
OpenLDAP server

- LDAP server is **slapd** from openldap.org
  - OpenLDAP Public License
- LDAP helm chart:
  - Old repository: https://github.com/helm/charts/tree/master/stable/openldap
  - New repository: https://artifacthub.io/packages/helm/geek-cookbook/openldap
- Docker image:
  - https://github.com/osixia/docker-openldap
  - MIT License
Session topics

• Adding a K8s cluster (association to a VIM)

• Onboarding of NS and NF packages
  • Building and onboarding NF and NS packages
  • CNF openldap_knf: structure and descriptor
  • NS openldap_ns: structure and descriptor
Session topics

• Helm charts
  • Intro
  • Adding a Helm chart repo

• Day-1 configuration for a helm chart

• LCM of the NS
  • Instantiation
  • Testing
  • Termination
Helpful scripts

Scripts in ~/Hackfest/HD2.4-CNF-Helm

- openldap-show-vim-and-cluster.sh
  - Shows VIM and K8s cluster already added to OSM
- openldap-build-and-onboard.sh
  - Builds the descriptors from source, and onboards them to OSM
- openldap-launch.sh
  - OSM CLI command to launch the network service
- openldap-watch-progress.sh
  - Command to help you watch the progress (ctrl-c to exit)
- openldap-check-k8s-status.sh
  - Checks the status of the deployment in Kubernetes
Adding a K8s cluster
Adding a K8s cluster

Your Kubernetes cluster needs to meet the following requirements:

• Kubernetes Load Balancer, to expose your CNFs to the network
• Kubernetes default Storage Class, to support persistent volumes.
Association of K8s cluster to a VIM
A K8s cluster is expected to be connected
Services in the K8s cluster might be exposed in a single network, but maybe in two networks

K8s cluster services exposed in a single VIM network

K8s cluster services exposed in two VIM networks
What if I have an isolated K8s cluster?

• Although these situations are discouraged (an isolated K8s cluster does not make sense in the context of an operator network), it is still possible to deploy there from OSM by creating a dummy VIM target and associating the K8s cluster to that VIM target

```bash
osm vim-create --name mylocation1 --user u --password p --tenant p --account_type dummy \
   --auth_url http://localhost/dummy

osm k8scluster-add cluster --creds .kube/config --vim mylocation1 \
   --k8s-nets '{k8s_net1: null}' --version "1.18" \
   --description="Isolated K8s cluster in mylocation1"
```
Your K8s cluster

A K8s cluster was already created and added to OSM for you!

- Information used to create the cluster:
  - Credentials file: kube.yaml
  - Version: 1.17
  - VIM: osm_hackfest_X
  - K8s nets:
    - net1: osm-ext
  - ClusterName: osm_hackfest_X
Checking your K8s cluster

cd $HOME/osm-packages
$HOME/Hackfest/HD2.4-CNF-Helm/openldap-show-vim-and-cluster.sh
Onboarding of NS and NF packages
Building and onboarding the CNF and NS packages

cd $HOME/osm-packages
$HOME/Hackfest/HD2.4-CNFK-Helm/openldap-build-and-onboard.sh
osm nfpkg-list
osm nspkg-list
CNF structure
The CNF consists of a single KDU (a helm chart)

NF: openldap_knf
KDU: ldap
   - Helm-chart: stable/openldap

External CP: mgmt-ext
CNF structure
K8s cluster requirements: services exposed by single network

Net 1: mgmtnet
External CP: mgmt-ext
NF: openldap_knf
CNF descriptor

vnfd:
  id: openldap_knf
  description: KNF with single KDU using a helm-chart for openldap
  df:
    - id: default-df
      
      k8s-cluster:
        nets:
          - id: mgmtnet

      kdu:
        - name: ldap
          helm-chart: stable/openldap

      ext-cpd:
        - id: mgmt-ext
          k8s-cluster-net: mgmtnet
          mgmt-cp: mgmt-ext
          product-name: openldap_knf
          provider: Telefonica
          version: '1.0'

K8s cluster requirements: 1 network to expose services

One KDU based on helm chart stable/openldap (repo:stable)

All the services exposed in the network are exposed through a single external CP
NS structure

NS: openldap_ns

NF: openldap_knf

VL: mgmtnet

CP: mgmt.-ext

CNF
NS descriptor

nsd:
  nsd:
  - description: NS consisting of a single KNF openldap_knf connected to mgmt network
    designer: OSM
    id: openldap_ns
    name: openldap_ns
    version: '1.0'

vnfd-id:
  - openldap_knf

virtual-link-desc:
  - id: mgmtnet
    mgmt-network: 'true'

df:
  - id: default-df

vnf-profile:
  - id: openldap
    vnfd-id: openldap_knf
    virtual-link-connectivity:
      - constituent-cpd-id:
        - constituent-base-element-id: openldap
          constituent-cpd-id: mgmt-ext
          virtual-link-profile-id: mgmtnet

One CNF: openldap_knf
One VL where the CNF will expose its services
Connecting the CNF to the VL
Intro to helm charts

• Concepts:
  • Helm-chart: a packaged format for deploying K8s applications
  • Helm: a package manager

• Documentation:
  • Helm: https://helm.sh/
  • Charts: https://helm.sh/docs/topics/charts/

• Well known repos bringing access to more than 20,000 K8s applications:
  • Stable*: https://charts.helm.sh/stable
  • Incubator*: https://charts.helm.sh/incubator
  • Bitnami: https://charts.bitnami.com/bitnami

(*) Repo locations: https://helm.sh/blog/new-location-stable-incubator-charts/
Helm chart structure
Chart File Structure

```
wordpress/
    Chart.yaml  # A YAML file containing information about the chart
    LICENSE     # OPTIONAL: A plain text file containing the license for the chart
    README.md   # OPTIONAL: A human-readable README file
    values.yaml # The default configuration values for this chart
    values.schema.json # OPTIONAL: A JSON Schema for imposing a structure on the values.yaml file
    charts/     # A directory containing any charts upon which this chart depends.
    crds/       # Custom Resource Definitions
    templates/  # A directory of templates that, when combined with values,
                # will generate valid Kubernetes manifest files.
    templates/NOTES.txt # OPTIONAL: A plain text file containing short usage notes
```
Helm chart structure
Template files (K8s manifest files + Go)

```yaml
apiVersion: v1
kind: ReplicationController
metadata:
  name: deis-database
  namespace: deis
  labels:
    app.kubernetes.io/managed-by: deis
spec:
  replicas: 1
  selector:
    app.kubernetes.io/name: deis-database
  template:
    metadata:
      labels:
        app.kubernetes.io/name: deis-database
    spec:
      serviceAccount: deis-database
      containers:
        - name: deis-database
          image: {{ .Values.imageRegistry }}/postgres:{{ .Values.dockerTag }}
          imagePullPolicy: {{ .Values.pullPolicy }}
          ports:
            - containerPort: 5432
          env:
            - name: DATABASE_STORAGE
              value: {{ default "minio" .Values.storage }}
```
Helm chart structure
Values.yaml

imageRegistry: "quay.io/deis"
dockerTag: "latest"
pullPolicy: "Always"
storage: "s3"
The helm chart could be very simple, or as complex as the Magma Orchestrator helm chart

- **Magma Orchestrator helm chart structure**
Where is the helm chart specified in the descriptor?

vnfd:
    description: KNF with single KDU using a helm-chart for openldap
    df:
        - id: default-df
    ext-cpd:
        - id: mgmt-ext
            k8s-cluster-net: mgmtnet
        id: openldap_knf
    k8s-cluster:
        nets:
            - id: mgmtnet
    kdu:
        - name: ldap
            helm-chart: stable/openldap
            mgmt-cp: mgmt-ext
            product-name: openldap_knf
            provider: Telefonica
            version: '1.0'

One KDU based on helm chart stable/openldap (repo:stable)
Adding a helm-chart repo

- OSM needs to know where to obtain the helm chart from
- The stable helm-chart repo is added by default
- You can other helm-chart repos if needed
  - Repo name: incubator
  - URI: [https://charts.helm.sh/incubator](https://charts.helm.sh/incubator)

```bash
osm repo-add --type helm-chart \
  --description "Incubator repository for helm-charts" \n  incubator \n  https://charts.helm.sh/incubator
osm repo-list
```

- When the CNF is deployed, its helm chart will be downloaded from the repo
Day-1 configuration
The values available for a helm chart can be provided to OSM as instantiation parameters

```yaml
vld:
  - name: mgmtnet
    vim-network-name: osm-ext
    additionalParamsForVnf:
      - member-vnf-index: openldap
    additionalParamsForKdu:
      - kdu_name: ldap
  additionalParams:
    # replicaCount: 2
    service:
      type: LoadBalancer
      loadBalancerIP: '172.21.251.X' # MetalLB IP Address
      adminPassword: osm4u
      configPassword: osm4u
      env:
        LDAP_ORGANISATION: "Example Inc."
        LDAP_DOMAIN: "example.org"
        LDAP_BACKEND: "hdb"
        LDAP_TLS: "true"
        LDAP_TLS_ENFORCE: "false"
        LDAP_REMOVE_CONFIG_AFTER_SETUP: "true"
```
Prepare your instantiation parameters

```
cp $HOME/Hackfest/HD2.4-CNF-Helm/openldap-params.yaml $HOME
vi $HOME/openldap-params.yaml
# Edit loadBalancerIP
```
Life Cycle Management of the NS
Instantiating the Network Service

$HOME/Hackfest/HD2.4-CNF-Helm/openldap-launch.sh
osm ns-list
$HOME/Hackfest/HD2.4-CNF-Helm/openldap-watch-progress.sh
Checking the status of the deployment in K8s

$HOME/Hackfest/HD2.4-CNFCnf-Helm/openldap-check-k8s-status.sh
Testing the Network Service

```
ldapsearch -x -H ldap://<LB_IP>:389 -b dc=example,dc=org -D "cn=admin,dc=example,dc=org" -w osm4u
```
Terminating the Network Service

```
osm ns-delete ldap
```
More information

- **User Guide - Using Kubernetes-based VNFs**

- **VNF Onboarding Guidelines - KNF walkthrough**