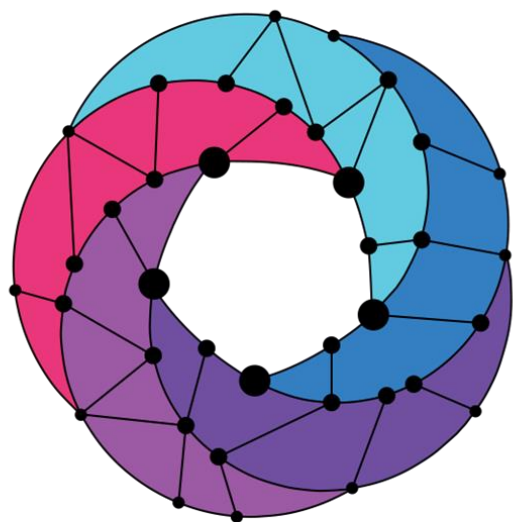




Open Source  
**MANO**

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](http://openldap.org)
  - [OpenLDAP Public License](http://www.openldap.org/pipermail/openldap-announce/201101/msg00001.html)
- 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



- 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

- 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



Open Source  
**MANO**

Adding a K8s cluster



# Adding a K8s cluster

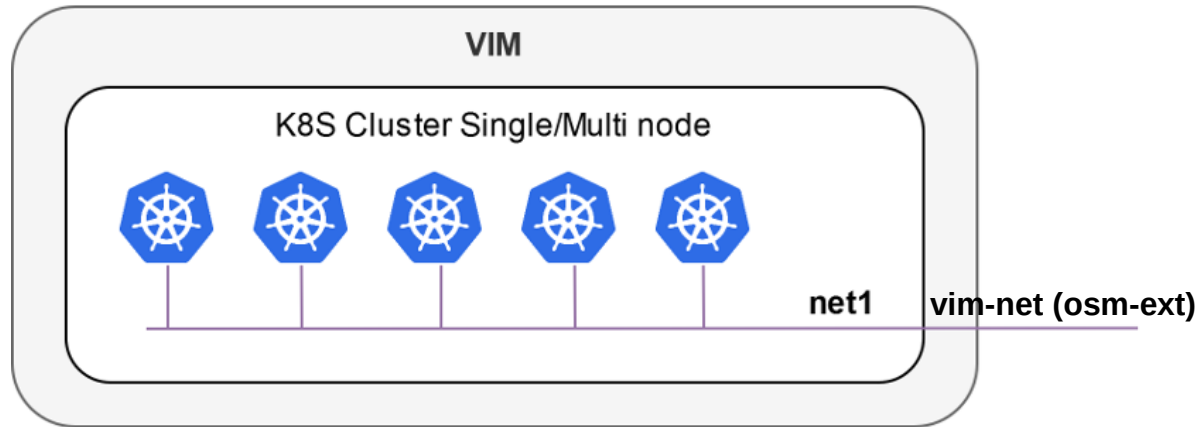
Your Kubernetes cluster needs to meet the following requirements:

- Kubernetes Load Balancer, to expose your CNFs to the network
- Kubernetes default Storage Class, to support persistent volumes.

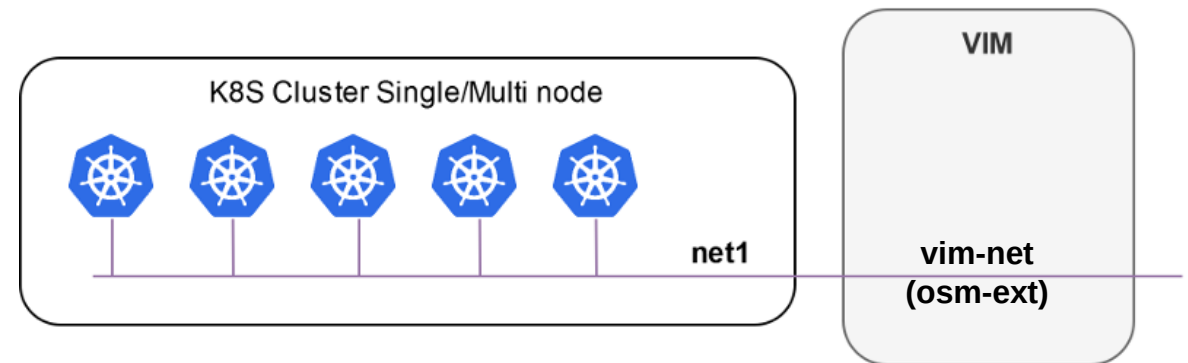
# Association of K8s cluster to a VIM

## A K8s cluster is expected to be connected

K8s cluster deployed inside a VIM



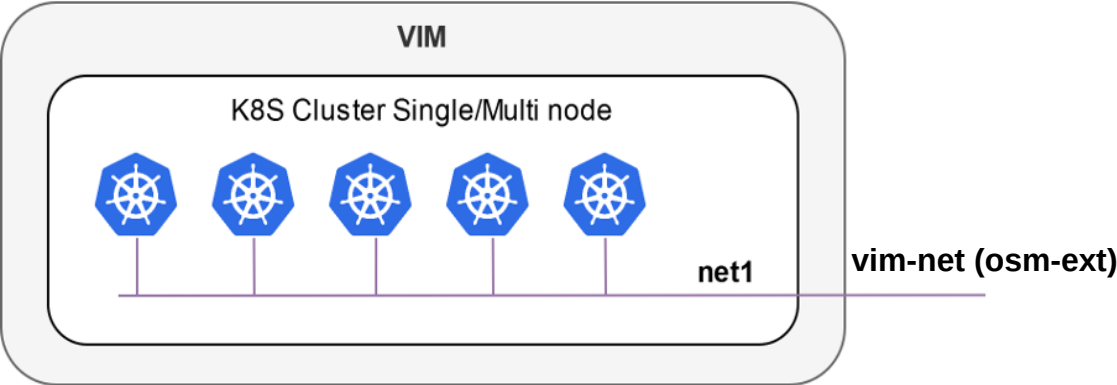
K8s cluster deployed outside a VIM,  
connected to a VIM network



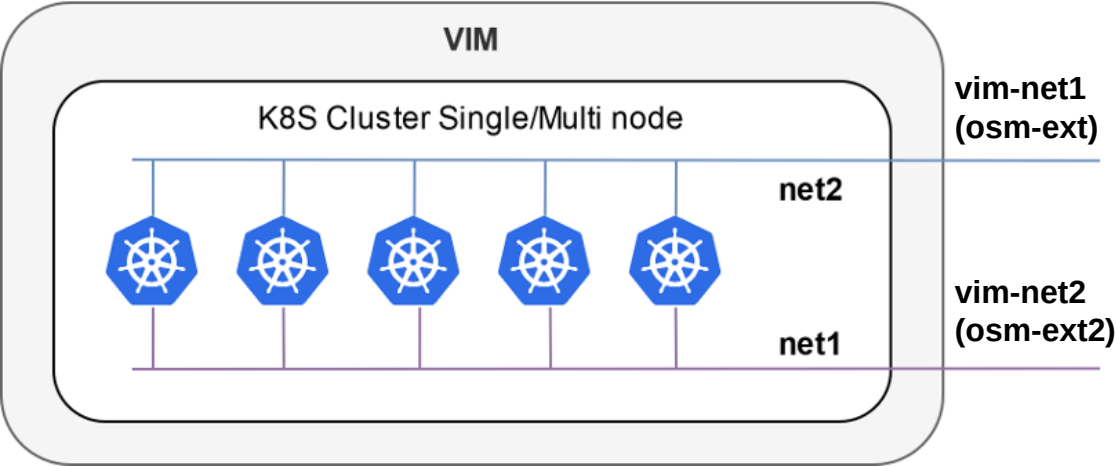


# Services in the K8s cluster might be exposed in a single network, but maybe in two networks

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

```
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"
```

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
```



Open Source  
**MANO**

Onboarding of NS  
and NF packages

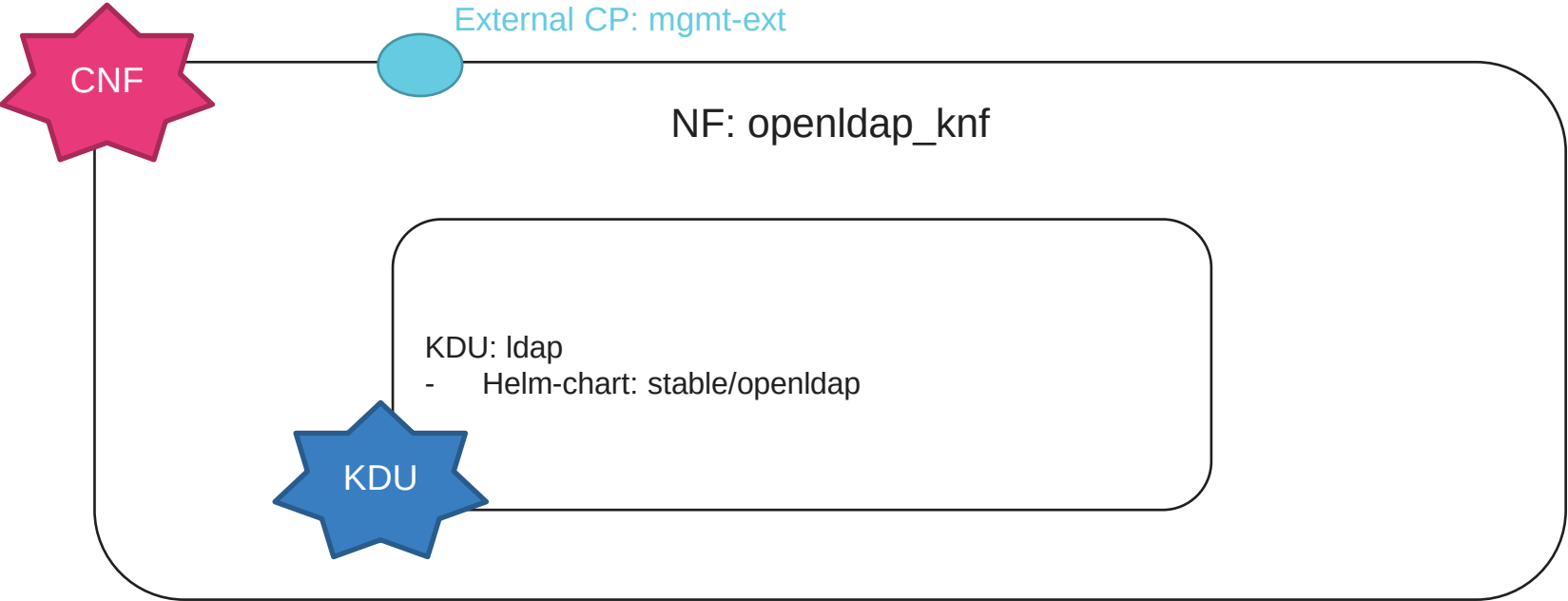


# Building and onboarding the CNF and NS packages

```
cd $HOME/osm-packages  
$HOME/Hackfest/HD2.4-CNF-Helm/openldap-build-and-onboard.sh  
osm nfpkg-list  
osm nspkg-list
```

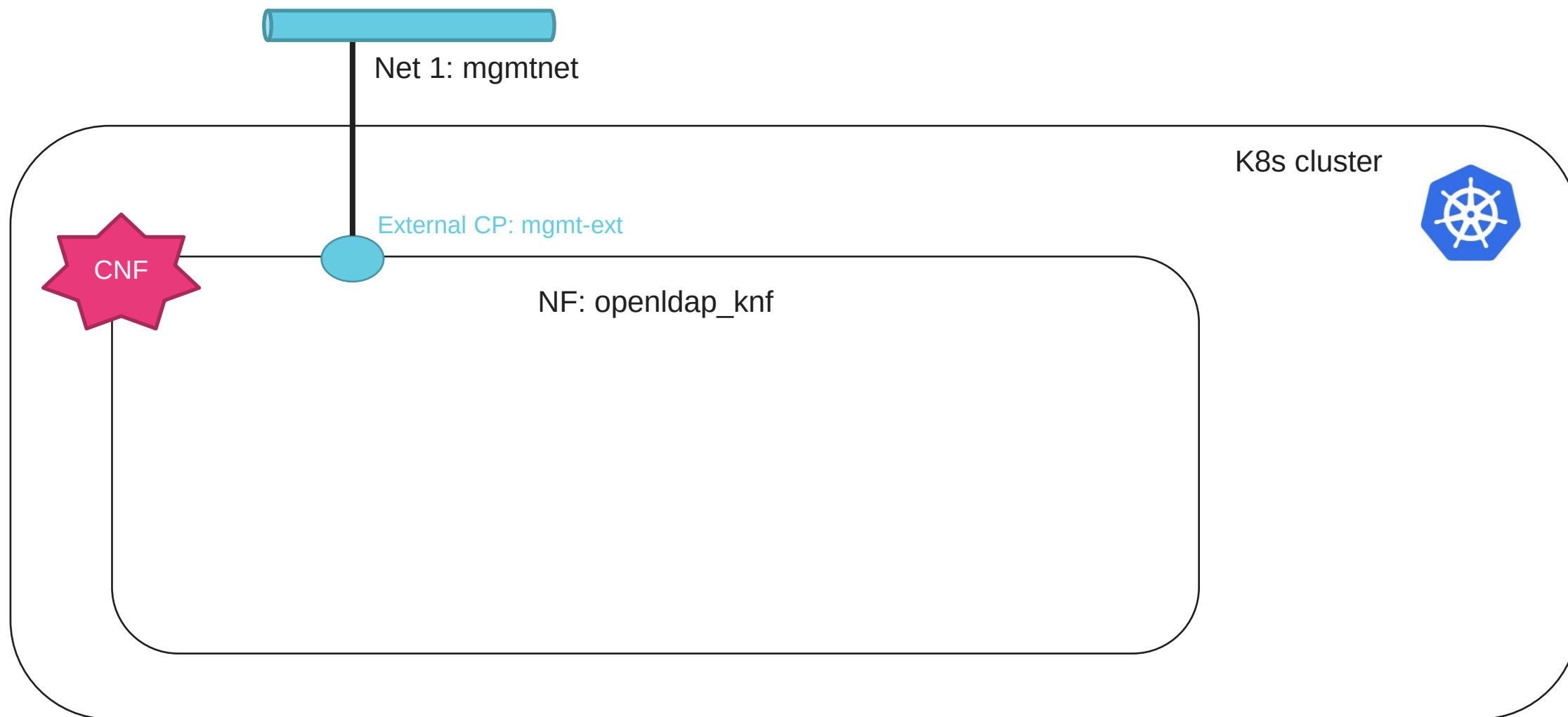
# CNF structure

The CNF consists of a single KDU (a helm chart)



# CNF structure

K8s cluster requirements: services exposed by single network



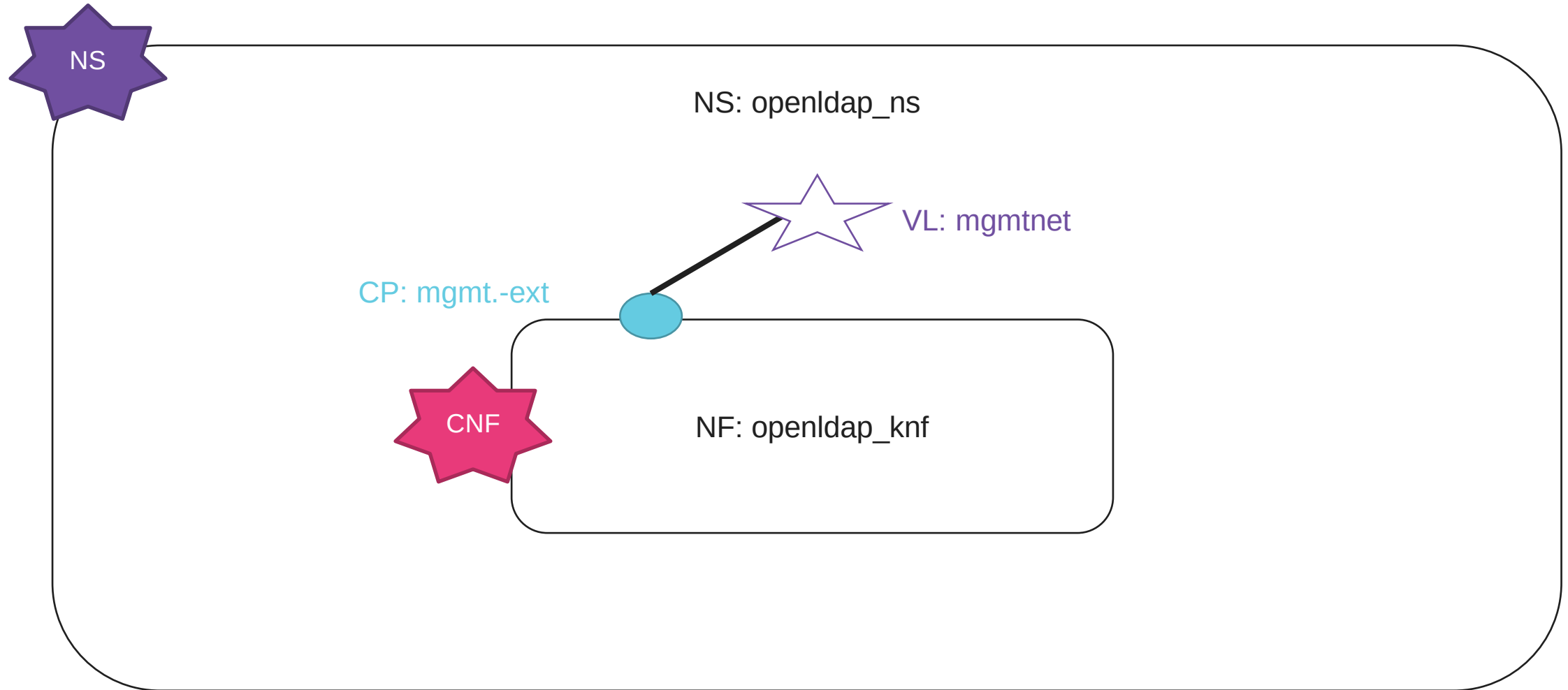


```
vnfd:
  id: openldap_knf
  description: KNF with single KDU using a helm-chart for openldap
  df:
    - id: default-df
  k8s-cluster:
    nets:
      - id: mgmtnet
  kdu:
    - name: ldap
      helm-chart: stable/openldap
  ext-cpd:
    - id: mgmt-ext
      k8s-cluster-net: mgmtnet
  mgmt-cp: mgmt-ext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



Open Source  
**MANO**

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)

```
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

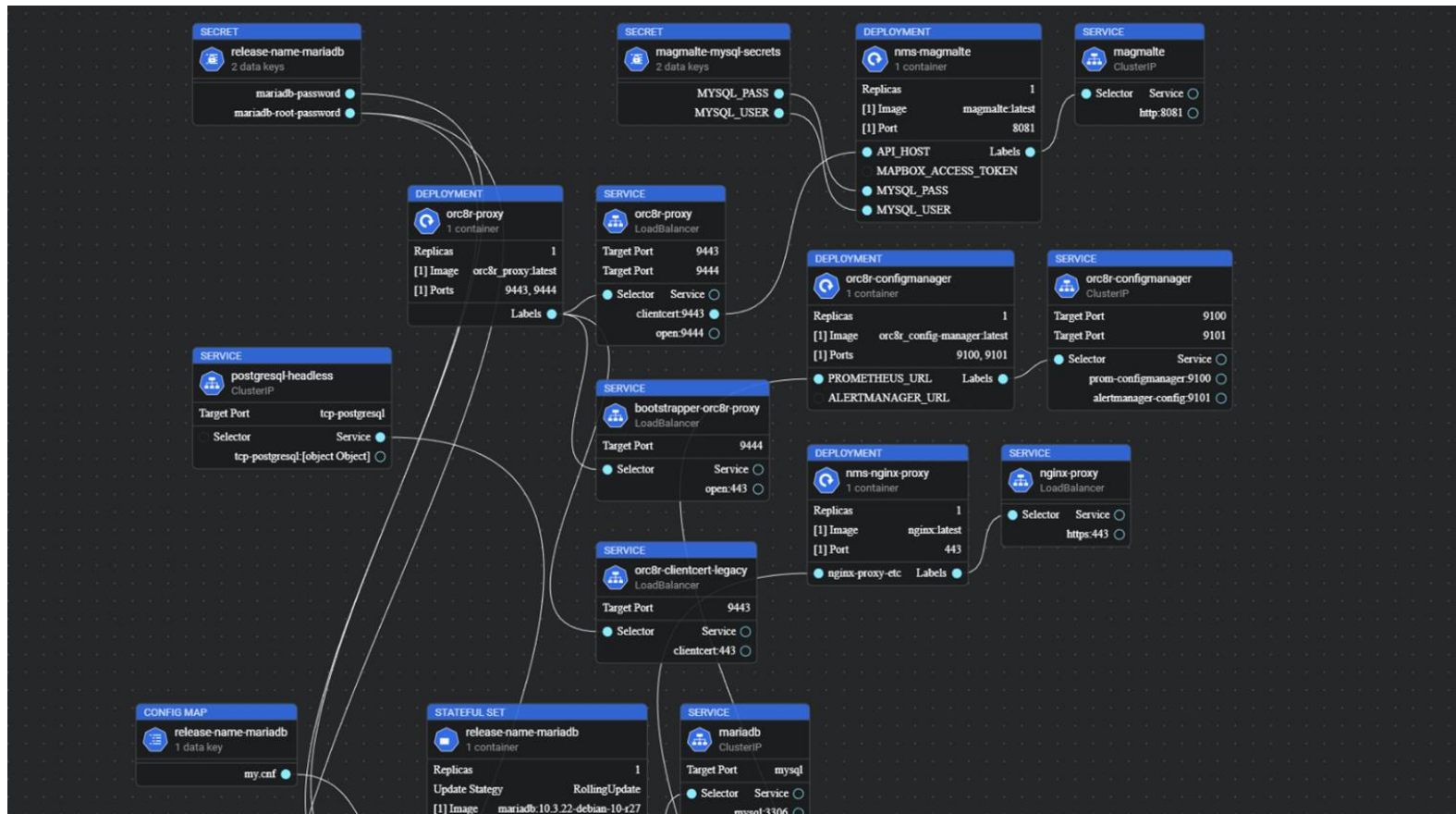
## Values.yaml

```
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>

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

- When the CNF is deployed, its helm chart will be downloaded from the repo



Open Source  
**MANO**

## Day-1 configuration



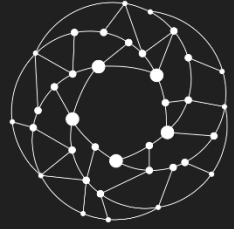
- The values available for a helm chart can be provided to OSM as instantiation parameters

```
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
```





Open Source  
**MANO**

# 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-CNF-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](#)



Search docs

TABLE OF CONTENTS

1. OSM Quickstart
2. OSM Architecture and Functions
3. Installing OSM
4. Setup of Virtual Infrastructure Managers (VIMs)
5. OSM Usage
  - 5.1. Deploying your first Network Service
  - 5.2. Advanced instantiation: using instantiation parameters
  - 5.3. Understanding Day-1 and Day-2 Operations
  - 5.4. Monitoring and autoscaling
  - 5.5. Using Network Slices
  - 5.6. Using Kubernetes-based VNFs (KNFs)
    - 5.6.1. Kubernetes installation
    - 5.6.2. OSM Kubernetes requirements
    - 5.6.3. Adding Kubernetes cluster

## 5.6. Using Kubernetes-based VNFs (KNFs)

From Release SEVEN, OSM supports Kubernetes-based Network Functions (KNF). This feature unlocks more than 20.000 packages that can be deployed besides VNFs and PNFs. This section guides you to deploy your first KNF, from the installation of multiple ways of Kubernetes clusters until the selection of the package and deployment.

### 5.6.1. Kubernetes installation

KNFs feature requires an operative Kubernetes cluster. There are several ways to have that Kubernetes running. From the OSM perspective, the cluster is not an isolated element, but it is a technology that enables the deployment of microservices in a cloud-native way. To facilitate the connection to the infrastructure, the cluster has to be associated to a VIM. There is a special case where the cluster is deployed in a baremetal environment without the management of the networking part but in general, OSM considers that the Kubernetes cluster is managed by a VIM.

For OSM you can use one of these three different ways to install your Kubernetes cluster:

1. OSM Kubernetes cluster Network Service
2. Self-managed Kubernetes cluster in a VIM
3. Kubernetes baremetal installation

### 5.6.2. OSM Kubernetes requirements

After the Kubernetes installation is completed, you need to check if you have the following components in your cluster.

1. **Kubernetes Loadbalancer**: to expose your KNFs to the network
2. **Kubernetes default Storageclass**: to support persistent volumes.

### 5.6.3. Adding Kubernetes cluster to OSM

In order to test Kubernetes-based VNF (KNF), you require a K8s cluster connected to a network in the VIM (e.g. `vim-net`). If you have already installed Kubernetes, you will need to add a VIM in order to add the Kubernetes cluster.



Search docs

TABLE OF CONTENTS

- Introduction
- Onboarding Requirements
- Day 0: VNF Instantiation & Management setup
- Day 1: VNF Services Initialization
- Day 2: VNF Runtime Operations
- Reference NSD/VNFD & Charms
- VNF Onboarding Walkthrough
- KNF Onboarding Walkthrough (Work in Progress)
  - Introduction
  - KNF Requirements
  - Building the KNF Descriptor
  - Launching the KNF Package
  - Testing the KNF Package
  - Testing functionality
- Advanced Charm Development
- Additional References

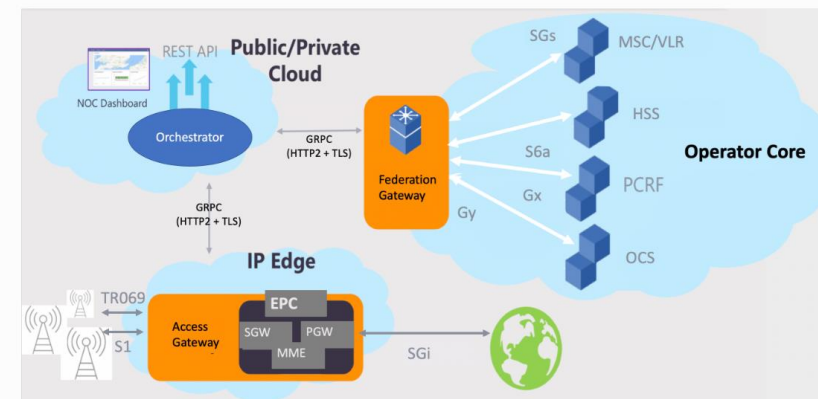
Docs » KNF Onboarding Walkthrough (Work in Progress)

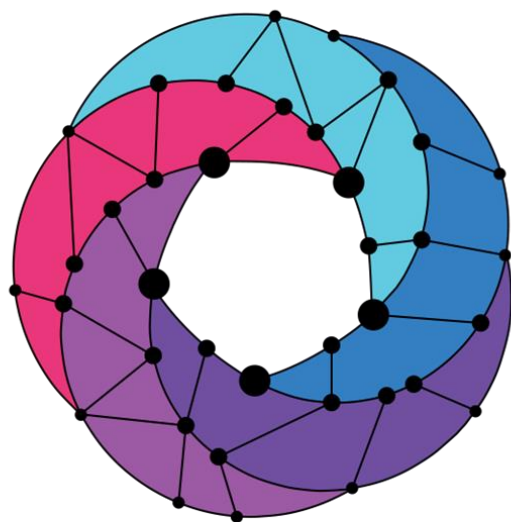
[View page source](#)

## KNF Onboarding Walkthrough (Work in Progress)

### Introduction

This section uses Facebook's Magma, an open-source software platform that gives network operators an open, flexible and extendable mobile core network solution.





# Open Source MANO

Find us at:

[osm.etsi.org](https://osm.etsi.org)  
[osm.etsi.org/docs/user-guide](https://osm.etsi.org/docs/user-guide)  
[opensourcemanoslack.com](https://opensourcemanoslack.com)

