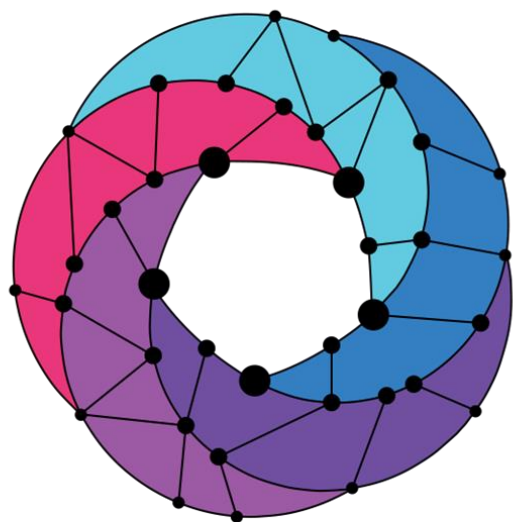




Open Source
MANO

OSM#10 Hackfest – Day 4

Session 3a. Modeling CNFs with helm charts

Gerardo García (Telefónica)

- CNF package
 - Create a skeleton
 - Modify descriptor
 - Validate and build
 - Onboard
- NS package
 - Create a skeleton
 - Modify descriptor
 - Validate and build
 - Onboard
- LCM
 - Making OSM aware of the helm chart repo
 - Instantiate
 - Upgrade and rollback
 - Terminate



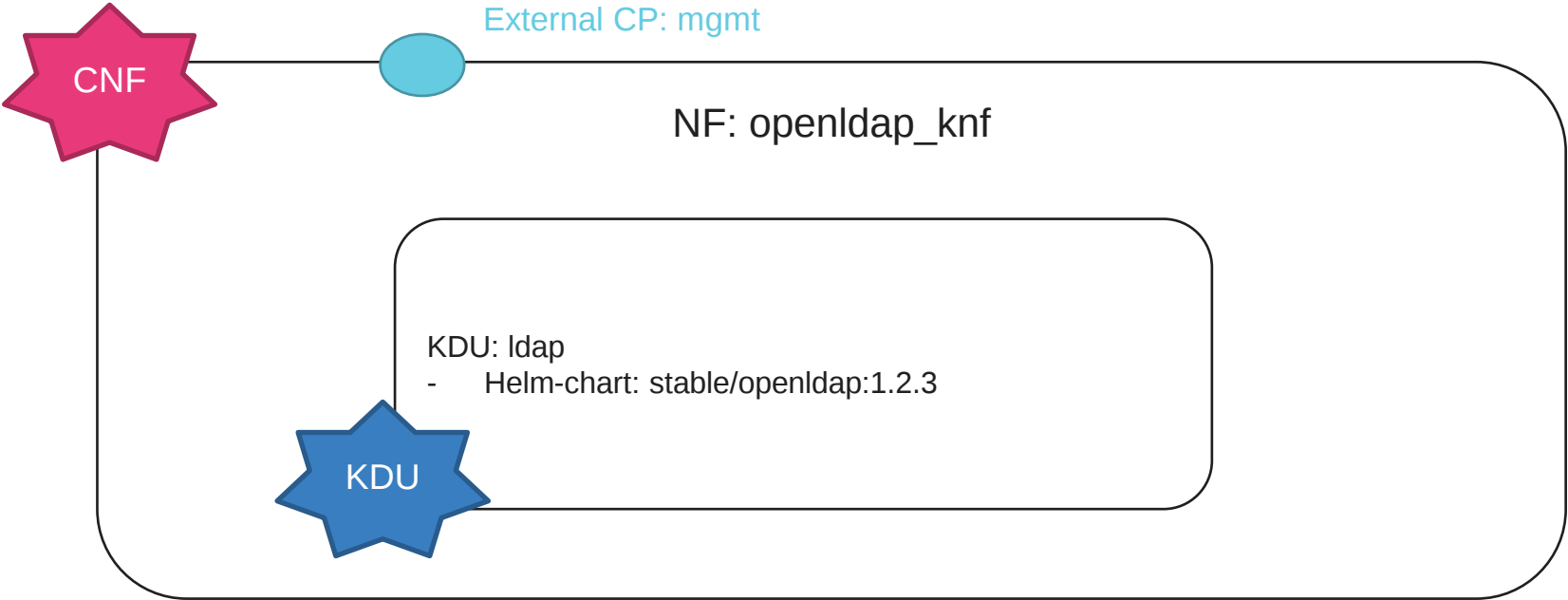
Open Source
MANO

CNF package



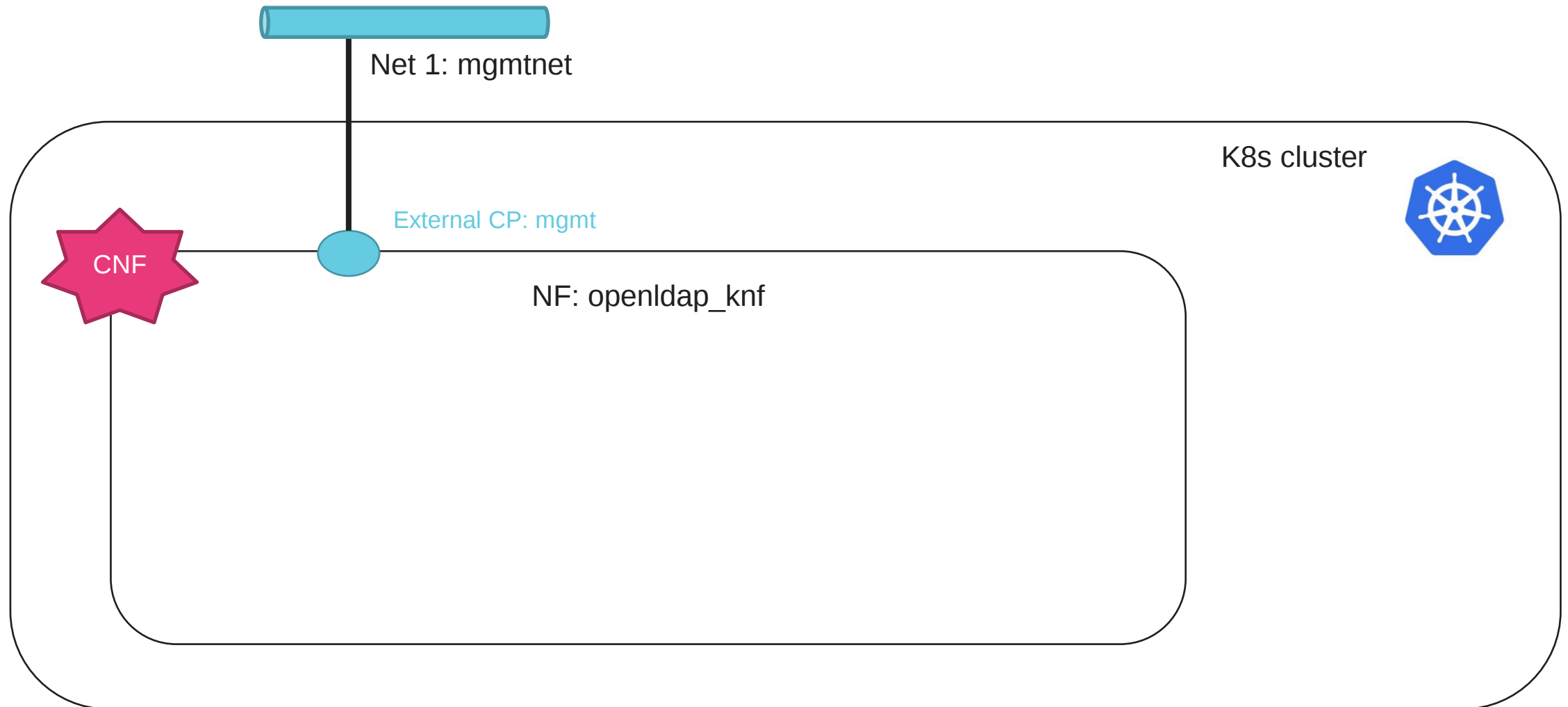
CNF structure

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



CNF structure

K8s cluster requirements: a single network



CNF package

Create skeleton and modify descriptor

- Our CNF: an LDAP server based on openldap helm chart
 - Stable helm chart repo: <https://charts.helm.sh/stable>

- Create skeleton

```
osm package-create vnf openldap
```

- Modify descriptor (copy it from shared folder)

```
cp /home/ubuntu/examples/hd4/openldap_vnfd.yaml openldap_vnf/
```

```
vnfd-catalog:
  schema-version: '3.0'
  vnfd:
    - id: openldap_knf
      name: openldap_knf
      short-name: openldap_knf
      description: KNF with single KDU using a helm-chart for openldap version 1.2.3
      vendor: Telefonica
      version: '1.0'
      mgmt-interface:
        cp: mgmt
      connection-point:
        - name: mgmt
      k8s-cluster:
        nets:
          - id: mgmtnet
            external-connection-point-ref: mgmt
      kdu:
        - name: ldap
          helm-chart: stable/openldap:1.2.3
```

K8s cluster requirements: 1 network to expose services

One KDU based on helm chart magma/orc8r (repo:magma)

CNF package

Validate, build and onboard

- Validate

```
osm package-validate openldap_vnf
```

- Build

```
osm package-build openldap_vnf
```

- Onboard

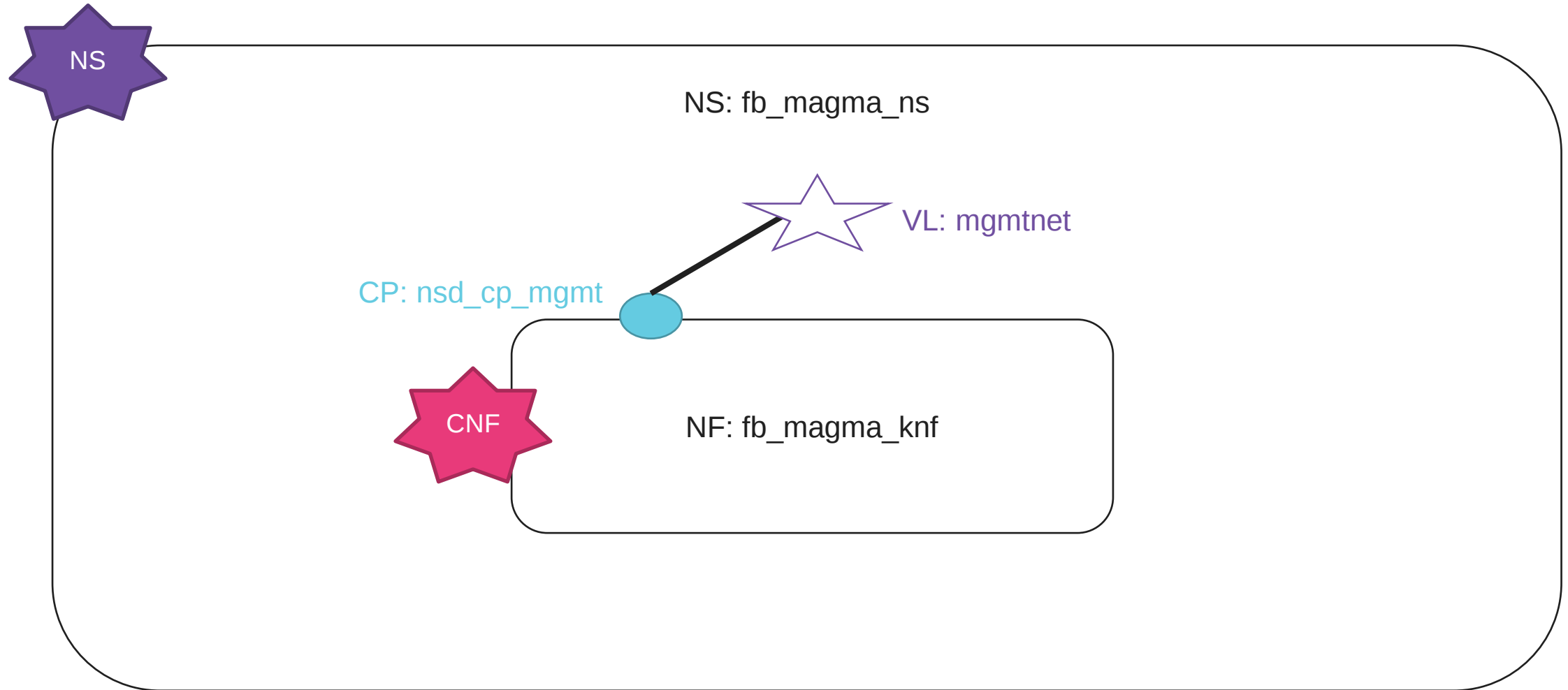
```
osm nfpkg-create openldap_vnf.tar.gz
```



Open Source
MANO

NS package





NS package

Create skeleton and modify descriptor

- Our NS: a simple NS with 1 CNF and 1 VLD

- Create skeleton

```
osm package-create ns openldap
```

- Modify descriptor (copy it from shared folder)

```
cp /home/ubuntu/examples/hd4/openldap_nsd.yaml openldap_ns/
```

NS descriptor

```
nsd-catalog:
  nsd:
  - id: openldap_ns
    name: openldap_ns
    short-name: openldap_ns
    description: NS consisting of a single KNF openldap_knf connected to mgmt network
    vendor: OSM
    version: '1.0'
    logo: osm.png
    constituent-vnfd:
      - member-vnf-index: openldap
        vnfd-id-ref: openldap knf
    vld:
      - id: mgmtnet
        name: mgmtnet
        type: ELAN
        mgmt-network: 'true'
        vnfd-connection-point-ref:
          - member-vnf-index-ref: openldap
            vnfd-id-ref: openldap_knf
            vnfd-connection-point-ref: mgmt
```

One CNF: openldap_knf

One VL where the CNF will expose its services

NS package

Validate, build and onboard

- Instead of using 3 commands, we will use only one

```
osm nspkg-create openldap_ns
```



Open Source
MANO

Life Cycle Mgmt
(LCM) of the NS



Adding a helm-chart repo

- We don't need to add a repo because the stable helm chart repo is added by default
- Example:

```
osm repo-add --type helm-chart \  
             --description "Bitnami helm chart repo" \  
             bitnami "https://charts.bitnami.com/bitnami"
```

- Launch the NS

```
osm ns-create --ns_name ldap --nsd_name openldap_ns \  
              --vim_account osm_hackfest_XX
```

- Check status of NS in OSM (READY)

```
osm ns-list
```

Checking status of the KDU instance

- Once ready, get the details of the KDU

```
osm vnf-list --ns ldap
VNF_ID=`osm vnf-list --ns ldap |grep ldap |awk '{print $2}'`
osm vnf-show $VNF_ID --kdu ldap
```

- You can also check the status of the deployment in K8s (one namespace per OSM project):

```
osm project-list
PROJECT_ID=`osm project-list|grep $OSM_PROJECT|awk '{print $4}'`
kubectl -n $PROJECT_ID --kubeconfig ~/kube.yaml get all
```

- Upgrade your KDU to use 2 replicas

```
osm ns-action ldap --vnf_name openldap --kdu_name ldap \  
    --action_name upgrade --params '{"replicaCount": "2"}'  
osm ns-op-list ldap
```

- Rollback to the previous state

```
osm ns-action ldap --vnf_name openldap --kdu_name ldap --action_name rollback
osm ns-op-list ldap
```

- Delete the NS

```
osm ns-delete magma_orc8r
osm ns-list
```

More information

- [User Guide - Using Kubernetes-based VNFs](#)
- [VNF Onboarding Guidelines - KNF walkthrough](#)
- [OSM Client Reference Guide - Package creation tools](#)



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

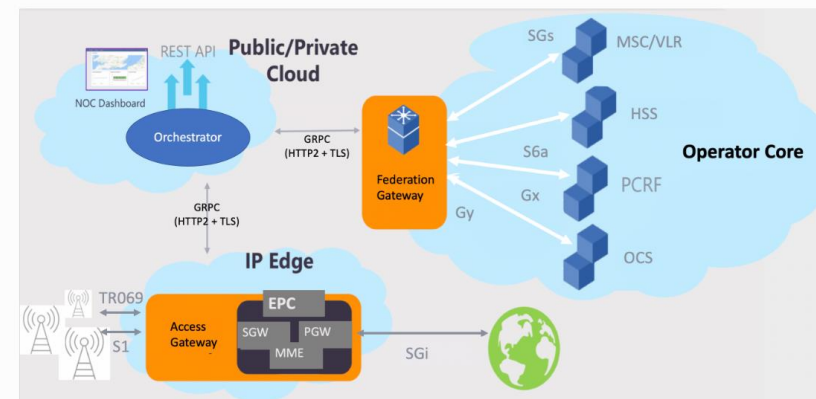
Docs » KNF Onboarding Walkthrough (Work in Progress)

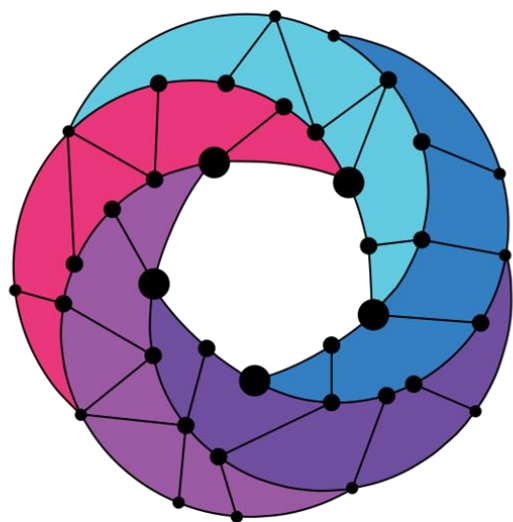
[View page source](#)

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
osm.etsi.org/docs
osm.etsi.org/wikipub

