OSM#9 Hackfest
Hack 1: Architecture & Installation
OSM Architecture Review
OSM Architecture overview

1. Unified message bus for async communications
2. Integrated components for placement, policy, fault and performance management
3. VCA controller for Generic NF configuration & indicator management
4. End to end orchestrator: LifeCycle Management (LCM) component
5. Unified Northbound Interface
6. Complete control through CLI and stand-alone UI
7. common DB, storage, authentication and TSDB systems

Common Services

- OSM IM
  - Common Database (NoSQL)
- Object Storage
- TSDB (Metrics)
- Auth

Components:

- osmclient
- light-ui
- NBI
- LCM
- VCA
- NaVC
- RO
- OSM IM
- PLA
- POL
- MON
- Kafka bus
When dealing with the creation, modification or deletion of users, projects and roles, the interacting components vary according to the selected backend.
When reading, uploading, modifying and deleting a Network Slice Template, Network Service Package or VNF Package, the following components interact.

CLI Example: `osm vnfpkg-create myvnfpackage.tar.gz`
Adding VIM/SDNC Sessions

When registering new sessions with VIMs or SDN Controllers, the following components interact.

CLI Example: `osm vim-create --name myVIM --user myuser --password myprecious --auth_url http://172.21.7.5:5000/v3 --tenant mytenant --account_type openstack`
When registering new sessions with Kubernetes clusters, the following components interact.

CLI Example: osm k8scluster-add --creds myCredentials.yaml --version '1.15' --vim myVIM --description "My K8s cluster" --k8s-nets '{"net1": "myVIMnet"}' myK8Cluster
When launching a new instance of a Network Service or Slice Instance (n x VNFs), the following components interact.

CLI Example: `osm ns-create --ns_name myNS --nsd_name myNSD --vim_account myVIM`
When launching a new instance of a Network Service or Slice Instance (n x VNFs), the following components interact.

CLI Example: `osm ns-create --ns_name myNS --nsd_name myNSD --vim_account myVIM`
When launching a new instance of a Network Service or Slice Instance (n x VNFs), with Day-1/2 automation, direct interaction with the NF is needed, so the following components interact.

CLI Example of Day-2 primitive: `osm ns-action myNS --vnf_name 1 --action_name myAction`
Instantiating with Placement

When launching a new instance of a Network Service or Slice Instance (n x VNFs), with placement support, the following components interact.

CLI Example: `osm ns-create --ns_name myNS --nsd_name myNSD --vim_account myVIM --config '{placement-engine: PLA, placement-constraints: {...}}'`

1. API call
2. Store NSR data
3. Launch call to LCM
4. Request NF placement (tied to predefined VIM metrics)
5. Instantiate NF resources through RO/VCA as usual
When launching a new instance of a Network Service or Slice Instance (n x VNFs) which is described with the collection of VNF Metrics that come from infrastructure (NFVI), the following components interact.

1. **commonDB (mongo)** continuously looks for active VNFs with metrics
2. **API calls to VIM, to collect metrics**
3. **looks for active metrics**
4. **reads & presents nfvi metrics**

Components:
- **tsdb (prometheus)**
- **grafana**
- **mon-collector**
- **VIM (external)**

© ETSI 2020
When launching a new instance of a Network Service or Slice Instance \((n \times \text{VNFs})\) which is described with the collection of VNF Metrics that come from the VNF itself, the following components interact.

1. **lcm** (metrics-collection primitive) is automatically executed after instantiation.
2. **Juju-metrics primitive** collects metrics from the VNF.
3. **tsdb (prometheus)** continuously looks for active VNFs with juju metrics.
4. **API calls to VNFs** are made to collect metrics using **mon-collector**.
5. **VCA** looks for active metrics.
6. **grafana** reads and presents VNF metrics.
When configuring alarms associated to scaling actions or just webhook notifications (through the VNFD), the following components interact.

NBI

commonDB (mongo)

mon-evaluator

tsdb (prometheus)

lcm

pol

webhook service (external)

(1) continuously looks for configured alarms at VNF record

(2) queries for metric values

(3) when triggered, puts alarm in bus for pol to take actions

(4a) if action is to scale: send to bus for LCM to proceed and store action to commonDB

(4b) if action is to notify, send notification to webhook service

UNDER CONSTRUCTION (REL8 - Mid 2020)
When creating Projects or Network Services, Grafana dashboards are created automatically and the following elements interact.

1. **commonDB (mongo)** continuously looks for new projects or NS with metrics

2. (a) if project:
   - create Project dashboard

2. (b) if NS:
   - create NS dashboard (with sample graphs)

(*) continuously delete obsolete dashboards

(2b) if NS: create NS dashboard (with sample graphs)
Troubleshooting OSM

A general approach for OSM Troubleshooting is to first look for error messages in “show” commands, as in:

```
osm ns-show [ns]
osm vim-show [vim]
```

Besides that, knowing which components interact for each operation, you can troubleshoot by looking at the logs of each component. All troubleshooting tips are being documented in the user guide, here:

https://osm.etsi.org/docs/user-guide/09-troubleshooting.html
OSM Installation methods
1. OSM can be installed in a single server or VM with the following requirements:

<table>
<thead>
<tr>
<th></th>
<th>CPU</th>
<th>RAM</th>
<th>DISK</th>
<th>NIC</th>
<th>Internet</th>
<th>SO</th>
</tr>
</thead>
<tbody>
<tr>
<td>MINIMUM</td>
<td>2</td>
<td>4GB</td>
<td>20GB</td>
<td>1</td>
<td>Yes</td>
<td>Ubuntu18.04 (64-bit variant required)</td>
</tr>
<tr>
<td>RECOMMENDED</td>
<td>2</td>
<td>8GB</td>
<td>80GB</td>
<td>1</td>
<td>Yes</td>
<td>Ubuntu18.04 (64-bit variant required)</td>
</tr>
</tbody>
</table>

2. Once you have prepared the host with the previous requirements, all you need to do is:

```bash
chmod +x install_osm.sh
```
OSM Installation methods

OSM R7 can be installed using these main options:

- **Docker Swarm**
  - `./install_osm.sh`

- **kubernetes**
  - `./install_osm.sh -c k8s`
  - `./install_osm.sh -c charmed`

For more information go to [https://osm.etsi.org/docs/user-guide/01-quickstart.html#installing-osm](https://osm.etsi.org/docs/user-guide/01-quickstart.html#installing-osm)
Hands-on: OSM Installation over Kubernetes
Hands-on: OSM Installation over Kubernetes

Scenario

K8S Single Cluster

- kube-system
- monitoring
- osm

Where:
- Namespace
- Pod
- Container

OSM GUI

© ETSI 2020
1. Follow the user-guide at:  
https://osm.etsi.org/docs/user-guide/01-quickstart.html#installing-osm

2. Prepare a VM with the requirements.

3. Download the installer for OSM R7

   ```bash
   ```

4. Make the installer executable

   ```bash
   chmod +x install_osm.sh
   ```
5. Run the installer with -c k8s flag

./install_osm.sh -c k8s

6. You will be asked to confirm the installation of the following components:

The installation will do the following
1. Install and configure LXD
2. Install juju
3. Install docker CE
4. Disable swap space
5. Install and initialize Kubernetes as pre-requisites.
Do you want to proceed (Y/n)? Y
7. When installation is finished, execute the following commands to check k8s installation:

```
kubectl get nodes
kubectl get namespaces
kubectl get pods --all-namespaces
kubectl get all -n kube-system
kubectl get all -n osm
kubectl describe pod light-ui-xyz -n osm
```

8. Test the OSM client:

```
osm --help
osm user-list
```
9. Go to OSM GUI at http://<VM-IP> and access with admin/admin