

OSM#9 Hackfest Hack 1: OSM System Monitoring and NS Instantiation





Hands-on: OSM System Monitoring





./install_osm.sh -c k8s --k8s_monitor
Access dashboard: http://<osm-host>:3001









• Available in the k8s deployment of OSM.

- •There is a similar feature for the docker swarm (classic) deployment of OSM (not to be discussed here)
- •Aimed at monitoring OSM infrastructure, while VNF/NS metrics are also available since Release 7.1.
- Implementation based on Prometheus operator (Helm chart), plus some Prometheus exporters (node, Kafka, mysql, mongodb), in "monitoring" namespace

More implementation details





6

http://ip-address>:3001 (admin:prom-operator)





Kubernetes cluster

 upstream dashboards
 in Prometheus
 operator helm chart

• Open Source MANO Specific dashboards for OSM

OSM Status summary

• Hosts

Kafka, mongodb, mysql

OSM Status summary





Failed pods / Failed nodes (if any)

K8s resources requested

OSM components status (up/down)

CPU/Memory per OSM component

Hosts status





Mongo, mysql and Kafka dashboards

Kafka

Messages produced/consumed Lag by consumer group Partitions per topic

Mongodb

Connections Document operation stats Network operations **Mysql** Connections Disk occupation (indexes, tables) Network operations

- •See all the objects deployed in the monitoring namespace
 - •kubectl --namespace monitoring get all
- •In particular, the dashboards are stored as configmaps
 - •kubectl --namespace monitoring get configmap
- Servicemonitors specify what is to be scrapped by Prometheus
 - •kubectl --namepsace monitoring get servicemonitor

Force no pods running nbi

kubectl scale --namespace osm --replicas=0 deployment/nbi

We are going to improve the dashboard

Go to Edit ->Visualization

Coloring: Activate "value"

Gauge: Deactivate "show"

Value Mappings: Set value mappings

null -> error

0 -> error

1 -> ok

• Get the summary dashboard configmap definition to your computer

scp ubuntu@<ip-addr>:/home/ubuntu/devops/installers/k8s/sum mary-dashboard.yaml .

- •In grafana, "export" [] "json", and copy in the data contents of the .yaml file defining the configmap
- Upload the modified file
 - scp summary-dashboard.yaml <u>ubuntu@<ip-addr>:/home/ubuntu/devops/installers/k8s</u>
- Update the dashboard
 - kubectl -n monitoring apply -f summary-dashboard.yaml

OSM Packages overview

12

Packages contain the information that orchestrators need to launch a network service. The are basically two types of packages.

The VNF Package

- It contains the characteristics of the VNF, for example:
 - The software image(s) it needs.
 - Compute resources.
 - Network connections between its components (Internal Virtual Links)
 - Performance requirements.
 - Automation scripts.
- Its main element is the VNF Descriptor (VNFD)
- It is built and provided by the VNF vendor.
- This applies in a similar way to new conceptual kinds of Network Functions (NFs), like a Physical NF (PNF), a Containerized NF (CNF), a Kubernetes-based NF (KNF), and Hybrid Network Package (HNF), etc.

Packages contain the information that orchestrators need to launch a network service. The are basically two types of packages.

The Network Service Package

- It contains the characteristics of the Network Service, for example:
 - The VNF(s) it needs.
 - Network connections between VNFs (external Virtual Links)
- Its main element is the NS Descriptor (NSD)
- It is built by the operator from VNFs that conform the Network Service that needs to be provided.

View page source

Package descriptors in OSM are modeled in an increasing alignment to ETSI NFV standards (SOL006) Everything that can be put in a descriptor to model a VNF or NS, is present at OSM's Information Model, maybe the richest model of the NFV MANO industry.

Visit this link to navigate the model: <u>https://osm.etsi.org/docs/user-guide/11-osm-im.html</u>

Open Source ΜΛΝΟ	[
Search docs]
TABLE OF CONTENTS	1
1. OSM Quickstart	١
2. OSM Architecture and Functions 3. Installing OSM	c
4. Setup of Virtual Infrastructure Managers (VIMs)	1
5. OSM Usage	E
6. OSM platform configuration	F
7. What to read next	

Docs » 11. ANNEX 3: OSM Information Model

11. ANNEX 3: OSM Information Model

11.1. YANG model in OSM repos

YANG models can be found in the IM repo under the models folder: https://osm.etsi.org/gitweb/?p=osm/IM.git;a=tree

OSM uses pyang and pyangbind to generate Python classes used by the different OSM components.

11.2. OSM IM tree representation

Below you can find tree representations of the VNFD (VNF Descriptor), NSD (Network Service Descriptor), NST (Network Slice Template), VNFR (VNF Record), NSR (Network Service Record), NSI (Network Slice Instance), both in navigable and plain text formats.

The NS Package is the one actually being launched in OSM. It requires constituent VNF Packages to be present in the system.

nsd:nsd-catalog:	
<pre>nsd: - id: hackfest_basic-ns name: hackfest_basic-ns short-name: hackfest_basic-ns</pre>	Network Service "hackfest_basic-ns"
<pre>description: Simple NS with a single VNF and a single VL version: '1.0' logo: osm.png constituent-vnfd:</pre>	It needs VNF "hackfest_basic-vnf" to be present
<pre>short-name: mgmtnet type: ELAN mgmt-network: 'true' vnfd-connection-point-ref: vnfd-id-ref: hackfest_basic-vnf member-vnf-index-ref: '1' vnfd-connection-point-ref: vnf-cp0</pre>	It will put the VNF in a new network called 'mgmtnet'

The VNF Package is the one describing a given Network Function.

It requires VIM/NFVIs to support whatever characteristic is being required through its descriptor.

vnfd:vnfd-catalog:	
vnfd:	VNE "haakfast hasis ynf"
– id: hackfest_basic-vnf	VINF Mackiest_basic-vill
name: hackfest_basic-vnf	
<pre>short-name: hackfest_basic-vnf</pre>	
version: '1.0'	
description: A basic VNF descriptor w/ one VDU	
logo: osm.png	
connection-point:	
– name: vnf-cp0	
type: VPORT	
vdu:	
– id: hackfest_basic-VM	
name: hackfest_basic-VM	
image: ubuntu1604	It has one VDO (VIVI) that requires an image
alternative-images:	called 'ubuntu1604' and a flavor with 1 vCPU
- vim-type: aws	
<pre>image: ubuntu/images/hvm-ssd/ubuntu-artful-17.10-amd64-server-20180509</pre>	1GB RAM and 10GB of storage.
count: '1'	Ū.
vm-flavor:	
vcpu-count: '1'	
memory-mb: '1024'	
storage-gb: '10'	
interface:	
- name: vdu-eth0	
type: EXTERNAL	
virtual-interface:	
type: PARAVIRT	It has an a interface, any sead to the Network Comise
external-connection-point-ref: vnf-cp0	It has one interface, exposed to the Network Service
mgmt-interface:	as external Connection Point "vnf-cn0"
cp: vnf-cp0	

Once NS Packages and their constituent VNF Packages are present in the system, and at least a VIM is registered, a Network Service can be launched.

	=										
MAIN NAVIGATION	NS Packages										ŝ
🛠 Home											
PROJECT											
Cverview	Show 10 ¢ entries										
Packages	Short Name	Identifier 11	Description 1	Vendor 11	Version 1	Actions					
NS Packages VNF Packages	cirros_vdu_alarm_ns	57fb3043-b162-4d0e-805c- a423bdd387da	Simple NS example with a cirros_vdu_alarm_vnf	OSM	1.0	1	r =	e		Ŧ	Û
NetSlice Templates	hackfest_basic-ns	fbd443fb-86e1-4f52-a672- e0eee23bf54e	Simple NS with a single VNF and a single VL		1.0	1	-	e	#	¥	
A Instances	hackfest_basic-ns-metrics	43d2d15f-56f4-4f03-a096- 6786829ad535	Simple NS with a single VNF and a single VL and Metrics		1.0	1	1	e		*	Û
NS Instances VNF Instances	openIdap_ns	17f62d25-bef1-4e21-b077- f4e0a5d1ec35	NS consisting of a single KNF openIdap_knf connected to mgmt network	OSM	1.0	1	-	e	*	¥	Û
 PDU Instances NetSlice Instances 	ubuntu-cnf-ns	4cbeb460-9812-48a2-ba7a- 32171dfcb314	NS with 1 KDU connected to the mgmtnet VL		1.0	1 0	-	e		Ŧ	Û
SDN Controllers	ubuntuvm_vnfmetric_autoscale_ns	d87397bc-2d0a-4729-9dda- a0db73d0c21b	Ubuntu VM with VNF metrics and load-based autoscaling NS	Whitestack		1	-	e		¥	Ū
VIM Accounts	ubuntuvm_vnfvdumetric_autoscale_ns	52e46d31-cebf-42be-aa78- 2cc1f26be008	Ubuntu VM with metrics and load-based autoscaling NS	Whitestack		1	r =	ø		Ŧ	

Hands-on: Integrating a VIM & Instantiating a basic Network Service

1. Create a VIM in OSM vía CLI

osm vim-create --name openstack-site-hackfest-x --user osm_hackfest_x --password
<Pass> --auth_url http://<VIM-IP>:5000/v3 --tenant osm_hackfest_x --account_type
openstack --config='{security_groups: default}'

2. Validate the VIM creation . The status should be ENABLED

osm vim-list
osm vim-show openstack-site-hackfest-x

Hands-on: Integrating a VIM

- 1. Create a VIM in OSM vía GUI
- 2. Go to VIM accounts -> add new VIM
 - Name: openstack-site-hackfest-x
 - Type: Openstack
 - VIM URL: http://**<VIM-IP>**:5000/v3
 - VIM Username: osm_hackfest_x
 - VIM Password: *****
 - Tenant name: osm_hackfest_x
- 3. Click in Create button
- 4. Validate the VIM creation . The status should be ENABLED

MANO MANO	=		😂 admin 🔻 😫 admin
MAIN NAVIGATION			
🛠 Home	☆ Home > VIMS		
	NEW VIM ACC	OUNT	
Overview	Name*		
Packages	~	openstack-site-nackiest-30	
NS Packages	Type*	Openstack	Ŧ
VNF Packages	VIM URL*	http://172.21.247.1:5000/v3	
NetSlice Templates	VIM Username*	osm_hackfest_30	
1 Instances	VIM Password*		
🚓 NS Instances	Tenant name*	osm_hackfest_30	
	Description	Description	
NetSlice Instances	Schema Type	Schema Type	
SDN Controllers	Schema Version	Schema Version	
VIM Accounts			
🛃 K8s	~	CONFIG PARAMETERS	
	Back to VIM Acco	unts	Create

Hands-on: Launching your first NS

Hands-on: Launching your first NS

NSD Descriptor and Diagram

1. Download the nsd and vnfd packages

wget http://osm-download.etsi.org/ftp/osm-5.0-five/6th-hackfest/packages/hackfest_basic_vnf.tar.gz wget http://osm-download.etsi.org/ftp/osm-5.0-five/6th-hackfest/packages/hackfest_basic_ns.tar.gz

2. Create the NSD and VNFD in OSM

osm vnfd-create hackfest_basic_vnf.tar.gz
osm nsd-create hackfest_basic_ns.tar.gz

3. Create an SSH key

ssh-keygen

4. Create the Network Service in OSM

osm ns-create --ns_name hackfest1 --nsd_name hackfest_basic-ns --vim_account openstack-site-hackfest-x
--ssh_keys .ssh/id_rsa.pub --config '{vld: [{name: mgmtnet, vim-network-name: osm-ext}] }'

5. Validate NS creation in OSM vía CLI

osm ns-list osm ns-show hackfest1

- 6. Validate NS creation in OSM vía GUI
 - Go to Instances -> NS Instances

NS Instance	es										A Ne	ew NS
Show 10	• entries							Sea	arch:			
Name 斗	Identifier	1t	Nsd name	It	Operational Status	Config Status 🔱	Detailed Status	Act	Actions			
hackfest1	502612af-07f0-4dd3-838e-a04ac3126983		hackfest_basic-ns		running	configured	done	i		Û	Actions -	
Showing 1 to 1	of 1 entries									Prev	vious 1	Next

7. Access to the VM created in Openstack VIM

ssh -i .ssh/id_rsa ubuntu@<MGMT_IP>

8. Delete NS, NSD and VNFD

osm ns-delete hackfest1
osm vnfd-delete hackfest_basic_vnf
osm nsd-delete hackfest_basic_ns

1. Create the VNF Descriptor

osm package-create vnf hackfest-basic

2. Create the NS Descriptor

osm package-create ns hackfest-basic

3. Build the packages

osm package-build hackfest-basic_vnf
osm package-build hackfest-basic_ns

4. Upload VNFD and NSD to OSM

osm vnfd-create hackfest-basic_vnf.tar.gz
osm nsd-create hackfest-basic_ns.tar.gz

5. Create the Network Service

osm ns-create --ns_name hf-basic --nsd_name hackfest-basic_nsd --vim_account openstack-site-hackfest-x
--ssh_keys ~/.ssh/id_rsa.pub --config '{vld: [{name: mgmt, vim-network-name: osm-ext}] }'

- 6. Validate NS creation in OSM vía GUI
 - Go to Instances -> NS Instances

7. Compare the VNFD of this example with the previous Hands-On, find the difference and fix it

Find us at: <u>osm.etsi.org</u> <u>osm.etsi.org/wikipub</u>

