

# Open Source MANO

## OSM Hackfest – Session 2 Creating a basic VNF and NS

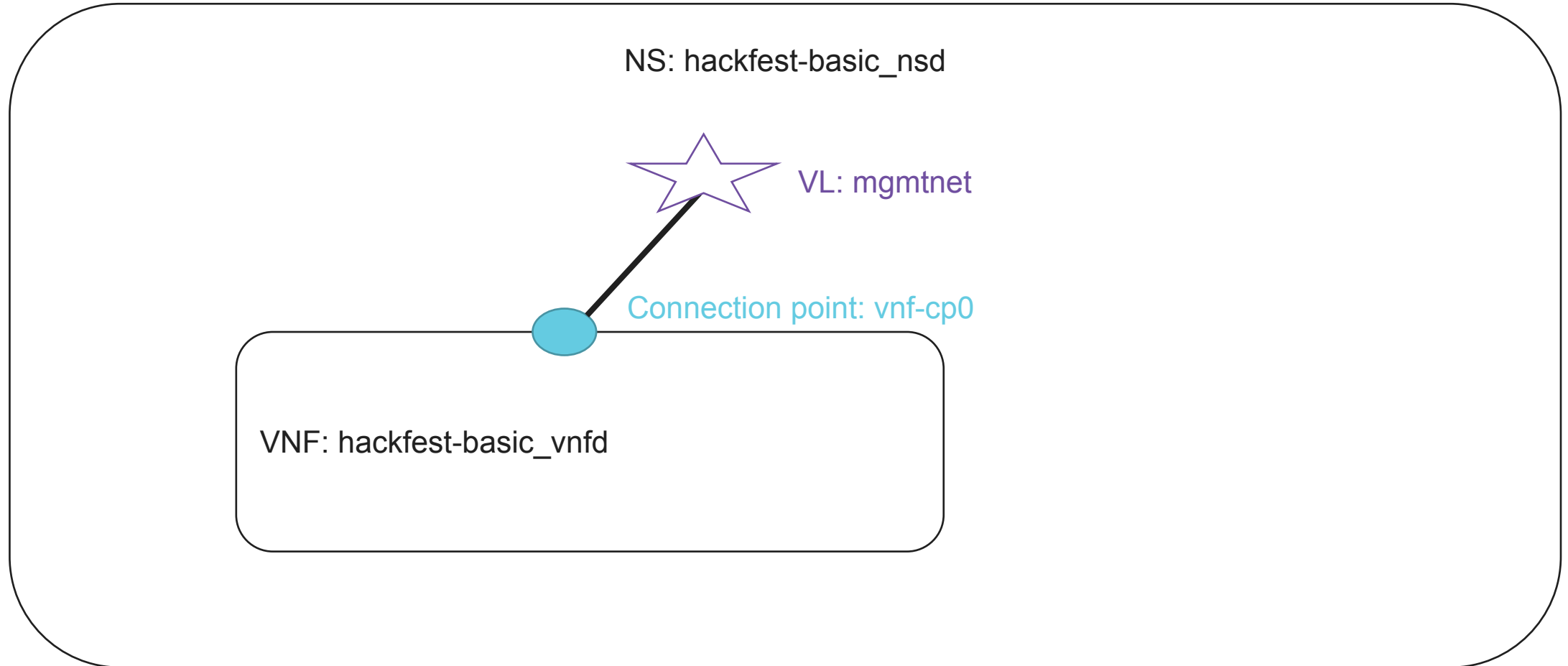
Benjamín Díaz (Whitestack)

Guillermo Calviño (Altran)

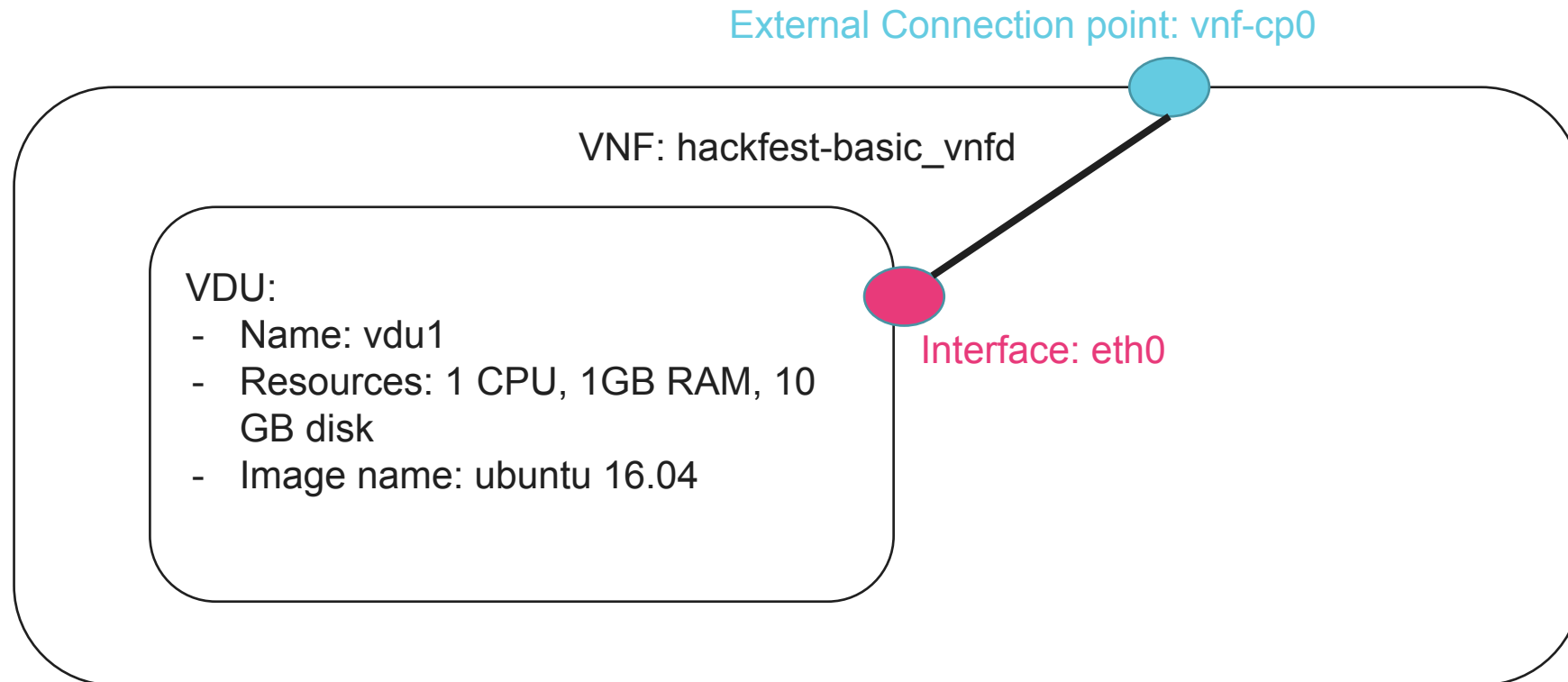
Tomás Villaseca (Whitestack)



# NS diagram



# VNF diagram



# Creating the VNF with the client

- Clone the devops repository
  - `git clone -b v6.0 https://osm.etsi.org/gerrit/osm/devops.git`
- Generate skeleton folder (VNF with only 1 VDU)
  - `devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t vnfd --image ubuntu1604 -c hackfest-basic`
- Go to `hackfest-simple_vnfd` folder and edit the descriptor:
  - Use the IM tree representation of VNFD as a reference:
    - <http://osm-download.etsi.org/ftp/osm-doc/vnfd.html>
  - Descriptor language is YAML:
    - Indentation is part of the markup
    - Use always the same indentation characters (TAB, 4 spaces, 2 spaces)
      - Recommendation: 2 spaces is the preferred indentation

# Editing the VNF descriptor

```
vnfd:
-   id: hackfest-basic_vnfd
  name: hackfest-basic_vnfd
  ...
  mgmt-interface:
    cp: vnf-cp0
  vdu:
-   id: hackfest-basic_vnfd-VM
  name: hackfest-basic_vnfd-VM
  vm-flavor:
    vcpu-count: 1
    memory-mb: 1024
    storage-gb: 10
    image: ubuntu 16.04
  interface:
-   name: eth0
    virtual-interface:
      type: VIRTIO
      ...
      external-connection-point-ref: vnf-cp0
  connection-point:
-   name: vnf-cp0
  ...
```

# OSM Information Model

- Information Model

- [https://osm.etsi.org/wikipub/index.php/OSM\\_Information\\_Model](https://osm.etsi.org/wikipub/index.php/OSM_Information_Model)

Path	Type	Subtype	Config	Required	URI
vnfd	module				
vnfd-catalog	container		config	current	/vnfd:vnfd-catalog
schema-version	leaf	string	config ?	current	/vnfd:vnfd-catalog/vnfd:schema-version
vnfd[id]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd
ip-profiles[name]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:ip-profiles
connection-point[name]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point
name	leaf	string	config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:name
id	leaf	string	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:id
short-name	leaf	string	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:short-name
type	leaf	manotypes:connection-point-type	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:type
port-security-enabled	leaf	boolean	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:port-security-enabled
internal-vld-ref	leaf	leafref	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:connection-point/vnfd:internal-vld-ref
vdu[id]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu
interface[name]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface
name	leaf	string	config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:name
position	leaf	uint32	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:position
mgmt-interface	leaf	boolean	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:mgmt-interface
type	leaf	interface-type	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:type
mac-address	leaf	string	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:mac-address
(connection-point-type)	choice		config	Choice current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface
:(internal)	case		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface
:(external)	case		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface
external-connection-point-ref	leafref	leafref	config ?	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:external-connection-point-ref
virtual-interface	container	leafref	config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:interface/vnfd:virtual-interface
volumes[name]	list		config	current	/vnfd:vnfd-catalog/vnfd:vnfd/vnfd:vdu/vnfd:volumes

# Validate the VNF descriptor and generate VNF package

- [https://osm.etsi.org/wikipub/index.php/Creating\\_your\\_own\\_VNF\\_package#Validate\\_descriptors](https://osm.etsi.org/wikipub/index.php/Creating_your_own_VNF_package#Validate_descriptors)

- The first time we need to install the python OSM IM package:

```
curl "https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX/OSM%20ETSI%20Release%20Key.gpg" | sudo apt-key add -
sudo apt-get update
sudo add-apt-repository -y "deb [arch=amd64] https://osm-download.etsi.org/repository/osm/debian/ReleaseSIX stable IM osmclient
devops"
```

```
# Install/update python-osm-im and its dependencies
sudo apt-get update
sudo apt-get install python-osm-im
sudo -H pip install pyangbind
```

- Validate VNF descriptor

- `devops/descriptor-packages/tools/validate_descriptor.py hackfest-basic_vnfd/hackfest-basic_vnfd.yaml`
- **Note:** No output means that the package was successfully validated.

- Generate VNF package

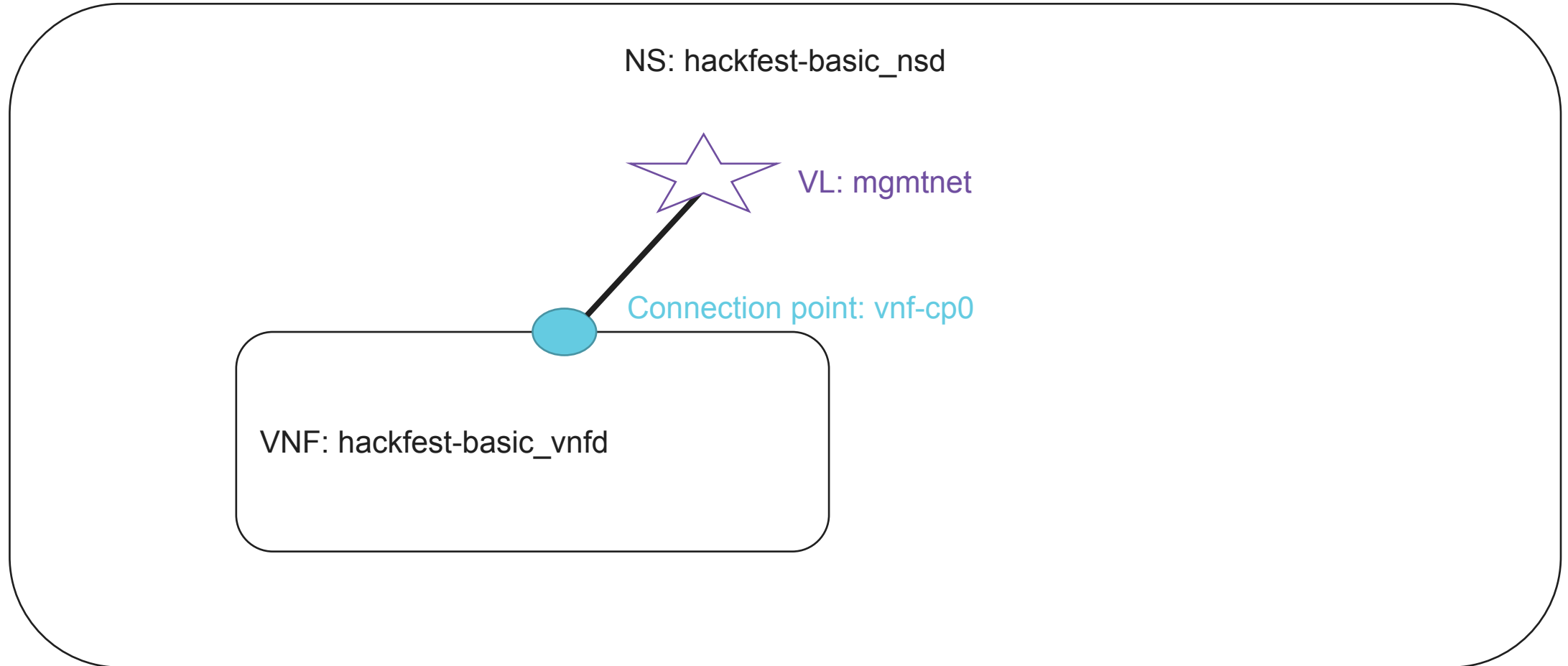
- `devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t vnfd -i hackfest-basic_vnfd/`

# Creating the NS with the client

- Generate skeleton folder (NS with only 1 VNF)
  - `devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t nsd -c hackfest-basic`
- Go to `hackfest-basic_nsd` folder and edit the descriptor:
  - Use the IM tree representation of NSD as a reference:
    - <http://osm-download.etsi.org/ftp/osm-doc/nsd.html>
  - Descriptor language is YAML:
    - Indentation is part of the markup
    - Use always the same indentation characters (TAB, 4 spaces, 2 spaces)
      - Recommendation: spaces preferred over tab



# NS diagram



# Editing the NS descriptor

```
nsd:
- id: hackfest-basic_nsd
  name: hackfest-basic_nsd
  ...
  constituent-vnfd:
  - member-vnf-index: 1
    vnfd-id-ref: hackfest-basic_vnfd
vld:
- id: mgmtnet
  name: mgmtnet
  type: ELAN
  mgmt-network: true
  vnfd-connection-point-ref:
  - member-vnf-index-ref: 1
    vnfd-connection-point-ref: vnf-cp0
    vnfd-id-ref: hackfest-basic_vnfd
```

# Validate the NS descriptor and generate NS package

- Validate NS descriptor

- `devops/descriptor-packages/tools/validate_descriptor.py hackfest-basic_nsd/hackfest-basic_nsd.yaml`

- Generate NS package

- `devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t nsd -N hackfest-basic_nsd/`

# Before the deployment

## Adding VNF and NS packages

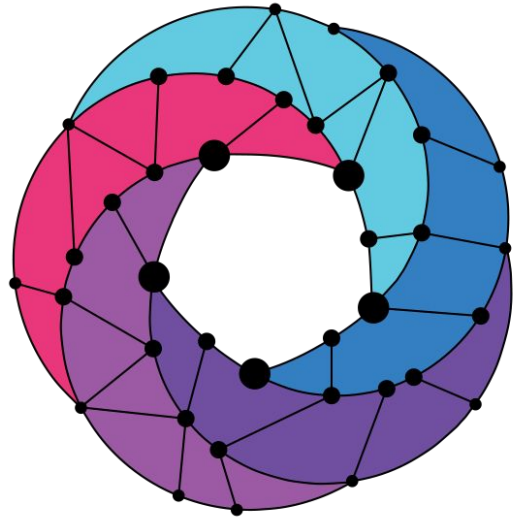
- VNF package:
  - `osm vnfd-list`
  - `osm vnfd-create hackfest-basic_vnfd.tar.gz`
  - `osm vnfd-show hackfest-basic_vnfd`
  - `osm vnfd-delete ...`
- NS package:
  - `osm nsd-list`
  - `osm nsd-create hackfest-basic_nsd.tar.gz`
  - `osm nsd-show hackfest-basic_nsd`
  - `osm nsd-delete ...`

# Deploying NS with the client

- Create an SSH key
  - `cat /dev/zero | ssh-keygen -q -N ""`
- Create the NS
  - `osm ns-create --ns_name hf-basic --nsd_name hackfest-basic_nsd \  
--vim_account openstack1 \  
--ssh_keys ~/.ssh/id_rsa.pub \  
--config '{vld: [ {name: mgmtnet, vim-network-name: osm-ext} ] }'`
- `osm ns-show hf-basic`
- `osm ns-delete hf-basic`
- Check VNF instances to see the instance and get the mgmt IP address of the VNF
  - `osm vnf-list`
  - `osm vnf-show ...`
- Connect to the VNF:
  - `ssh ubuntu@<IP>`

# Deploying NS with the UI

- Go to NS packages. In hackfest1-ns, click in “Actions: Instantiate NS”
- Complete the form
  - Add a name to the NS instance
  - Select the VIM where the NS will be deployed
  - Specify in the config section a default VIM network name to map “mgmtnet”:
    - `{vld: [ {name: mgmtnet, vim-network-name: osm-ext} ] }`
  - Paste your SSH key
- Go to VNF instances to see the instance and get the mgmt IP address of the VNF, then connect to the VNF:
  - `ssh ubuntu@<IP>`



# Open Source MANO

Find us at:

[osm.etsi.org](https://osm.etsi.org)  
[osm.etsi.org/wikipub](https://osm.etsi.org/wikipub)