NS diagram

NS: hackfest_multivdu_nsd

VNF: hackfest_multivdu_vnfd
CP: vnf-data
VL: datanet
CP: vnf-data

VNF: hackfest_multivdu_vnfd
CP: vnf-mgmt
VL: mgmtnet
CP: vnf-mgmt
VNF diagram

VNF: hackfest_multivdu_vnfd

VDU: mgmtVM
- Image name: cirros034
- VM Flavor: 1 CPU, 128 RAM, 1 GB disk
- Interfaces:
  - mgmtVM-eth0: VIRTIO
  - mgmtVM-eth1: VIRTIO

External Connection point: vnf-mgmt

VL: internal

VNF: hackfest_multivdu_vnfd

VDU: dataVM
- Image name: cirros034
- VM Flavor: 1 CPU, 128 RAM, 1 GB disk
- Interfaces:
  - dataVM-eth0: VIRTIO
  - dataVM-xe0: VIRTIO

External Connection point: vnf-data
Final Multi-VDU Picture

NS: hackfest_multivdu-ns

VNF: hackfest_multivdu-vnf

VNF 1 / VDU: mgmtVM

VNF 1 / VDU: dataVM

VNF 2 / VDU: mgmtVM

VNF 2 / VDU: dataVM

Internal VLDs & CPs

External VLDs & CPs

© ETSI 2019
VNF/NS Compose

- Compose a VNF or NS graphically.
VNF diagram

External Connection point: vnf-mgmt

VNF: hackfest_multivdu_vnfd

VL: internal

ICP: mgmtVM-internal

mgmtVM-eth0

mgmtVM-eth1

VDU: mgmtVM
- Image name: cirros034
- VM Flavor: 1 CPU, 128MB RAM, 1 GB disk
- Interfaces:
  - mgmtVM-eth0: VIRTIO
  - mgmtVM-eth1: VIRTIO

External Connection point: vnf-data

ICP: dataVM-internal

dataVM-eth0
dataVM-xe0

VDU: dataVM
- Image name: cirros034
- VM Flavor: 1 CPU, 128MB RAM, 1 GB disk
- Interfaces:
  - dataVM-eth0: VIRTIO
  - dataVM-xe0: VIRTIO
User Interface

• Steps:
  • Compose a new VNF
  • Create new Package

[Diagram showing the User Interface for composing a new VNF and creating a new package]
VNFD Composer

• **Steps**
  - VNFD Composer

• **Keyboard shortcuts**
  - Create edge: Select the first vertex by clicking on it, **Shift** + **left-click** on another vertex (different than the selected one).
  - Delete edge: Select the vertex by clicking on it, **right-click** + **Delete**
Creating the new multi-VDU VNF (1/4)

**Steps**
- Create VDUs
- Edit the descriptor to add the flavor:
  ```json
  vm-flavor: {memory-mb: '128', storage-gb: '1', vcpu-count: '1'}
  ```
Creating the new multi-VDU VNF (2/4)

• Steps

  • Create Connection Points: vnf-mgmt vnf-data (Drag and drop)

  • Link CPs with VDUs (Shift + Left Click)
Creating the new multi-VDU VNF (3/4)

• Steps
  • Create Internal VL:  (Drag and drop)
    - Virtual Link
      - **short-name**: internal
      - **Name**: internal
      - **ip-profile-ref**: 
      - **Type**: ELAN
      - **id**: internal
  • Link internal VL with VDUs (Shift + Left Click)
    • VNFD composer automatically create the internal connection points:
      - mgmtVM-internal
      - dataVM-internal
  • Edit the descriptor to add the CP in mgmt-interface

```yaml
vnfd:vnfd-catalog:
  vnfd:
    - connection-point:
      - name: vnf-mgmt
        type: VPORT
      - name: vnf-data
        type: VPORT
    description: ''
    id: MultiVDU_vnf
    internal-vld:
      - id: internal
        internal-connection-point:
          - id-ref: intcp_j5j9
          - id-ref: intcp_lh6y
        ip-profile-ref: ''
        name: internal
        short-name: internal
        type: ELAN
      mgmt-interface: cp: 'vnf-mgmt'
      name: MultiVDU_vnf
      short-name: MultiVDU_vnf
```
Creating the new multi-VDU VNF (4/4)

- Final Scenario multiVDU_vnfd

And finally, this is the sample file of Hackfest Multi VDU VNF Descriptor
https://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/packages/hackfest_multivdu_vnf.tar.gz
NS diagram

NS: hackfest_multivdu_nsd

VNF: hackfest_multivdu_vnfd
CP: vnf-mgmt
CP: vnf-data

VNF: hackfest_multivdu_vnfd
CP: vnf-mgmt
CP: vnf-data

VL: mgmtnet
VL: datanet
User Interface

- **Steps:**
  - Compose a new NS
  - Create new Package
NSD Composer

• Steps
  • NSD Composer

• Keyboard shortcuts

- Create edge: Select the first vertex by clicking on it, **Shift** + **left-click** on another vertex (different than the selected one).
- Delete edge: Select the vertex by clicking on it, **right-click** + **Delete**
Creating the NSD (1/3)

- Select VNFs: (Drag and drop)

  **VNF**
  - member-vnf-index: 1
  - vnfd-id-ref: MultiVDU_vnfd

  **VNF**
  - member-vnf-index: 2
  - vnfd-id-ref: MultiVDU_vnfd

- Create VLs: (Drag and drop)

  **Virtual Link**
  - Vim network name: osm-ext
    - Name: mgmtnet
    - Mgmt network: true
    - Type: ELAN
    - Id: mgmtnet
  
  **Virtual Link**
  - Vim network name: 
    - Name: datanet
    - Mgmt network: false
    - Type: ELAN
    - Id: datanet
Creating the NSD (2/3)

• Steps
  • Link VLs with VNFs (Shift + Left Click)
    • You need to know the name for the CPs (vnf-data and vnf-mgmt)
Creating the NSD (3/3)

- Final Scenario multiVDU_nsd

And finally, this is the sample file of Hackfest Multi VDU VNF Descriptor
https://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/packages/hackfest_multivdu_ns.tar.gz
Deploying NS in the UI

• Onboard VNFD and NSD to catalog using the UI

• Launch the NS from the UI
  • Depending on the VIM, specify a VIM network name to map mgmtnet
  • If you need to change the VIM, change the network name using config:
    
    ```
    {vld: [{name: mgmtnet, vim-network-name: osm-ext}]}
    ```

• Click the info button to see the mgmt IP address of each VNF

• Connect to each VNF:
  • `ssh cirros@<IP>`
    • password: `cubswin:`
Final Multi-VDU Picture

NS: hackfest_multivdu-ns
VNF: hackfest_multivdu-vnf

VNF 1 / VDU: mgmtVM
VNF 1 / VDU: dataVM

VNF 2 / VDU: mgmtVM
VNF 2 / VDU: dataVM

Internal VLDs & CPs
External VLDs & CPs
Network modelling with IP Profiles

• Using IP Profiles, we can configure the attributes of subnets that are created by OSM. We can do it for internal or external VLDs.

• Subnet's DHCP server will not deliver a default gateway if explicitly set to 0.0.0.0

External VLD with IP Profile (NSD level)

```json
... - id: MultiVDU_nsnd
... ip-profiles:
- name: profile_external1
  description: external network
  ip-profile-params:
    ip-version: ipv4
dns-server: 8.8.8.8
gateway-address: 0.0.0.0
subnet-address: 192.168.17.0/24
dhcp-params:
  enabled: true
vld:
- id: datanet
  ip-profile-ref: profile_external1
...
```

Internal VLD with IP Profile (VNFD level)

```json
... - id: MultiVDU_vnfd
... ip-profiles:
- name: p1
  description: p1
  ip-profile-params:
    ip-version: ipv4
dns-server: 0.0.0.0
gateway-address: 0.0.0.0
subnet-address: 192.168.100.0/24
dhcp-params:
  enabled: true
internal-vld:
- id: internal
  ip-profile-ref: p1
...```

Note: attributes with no values (0.0.0.0) are informational only, they can be removed.
Network modelling with Static IPs

- We can also set static IP addresses, having IP Profile and DHCP enabled.

**External VLD with IP Profile (NSD level)**

```
...  - id: MultiVDU_ns
    ...
    ip-profiles:
    - name: profile_external1
      description: external network
      ip-profile-params:
        ip-version: ipv4
        subnet-address: 192.168.17.0/24
        dhcp-params:
          enabled: true
    vld:
    - id: datanet
      ip-profile-ref: profile_external1
      ... 
      vnfd-connection-point-ref:
      - ...
      ip-address: 192.168.17.100
```

**Internal VLD with IP Profile (VNFD level)**

```
...  - id: MultiVDU_vnfd
    ...
    ip-profiles:
    - name: p1
      description: p1
      ip-profile-params:
        ip-version: ipv4
        subnet-address: 192.168.100.0/24
        dhcp-params:
          enabled: true
    internal-vld:
    - id: internal
      ip-profile-ref: p1
      ...
      internal-connection-point:
      - id-ref: mgmtVM-internal
        ...
        ip-address: 192.168.100.100
```
Network modelling with MACs

- We can set MAC addresses as well, just set them up at the VDU level.

```yaml
Changing MAC (VNFD level)
...
  interface:
    - name: mgmtVM-eth0
      position: '1'
      type: EXTERNAL
      virtual-interface:
        type: VIRTIO
        external-connection-point-ref: vnf-mgmt
        mac-address: '01:02:03:01:02:03'
    - name: mgmtVM-eth1
      position: '2'
      type: INTERNAL
      virtual-interface:
        type: VIRTIO
        internal-connection-point-ref: mgmtVM-internal
        mac-address: '03:02:01:03:02:01'
...
```

--> Be careful about duplicated MACs!