

Open Source MANO

OSM Hackfest – Session 4
Adding day-0 configuration to VNFs

Eduardo Sousa (Canonical)
Guillermo Calviño (Altran)

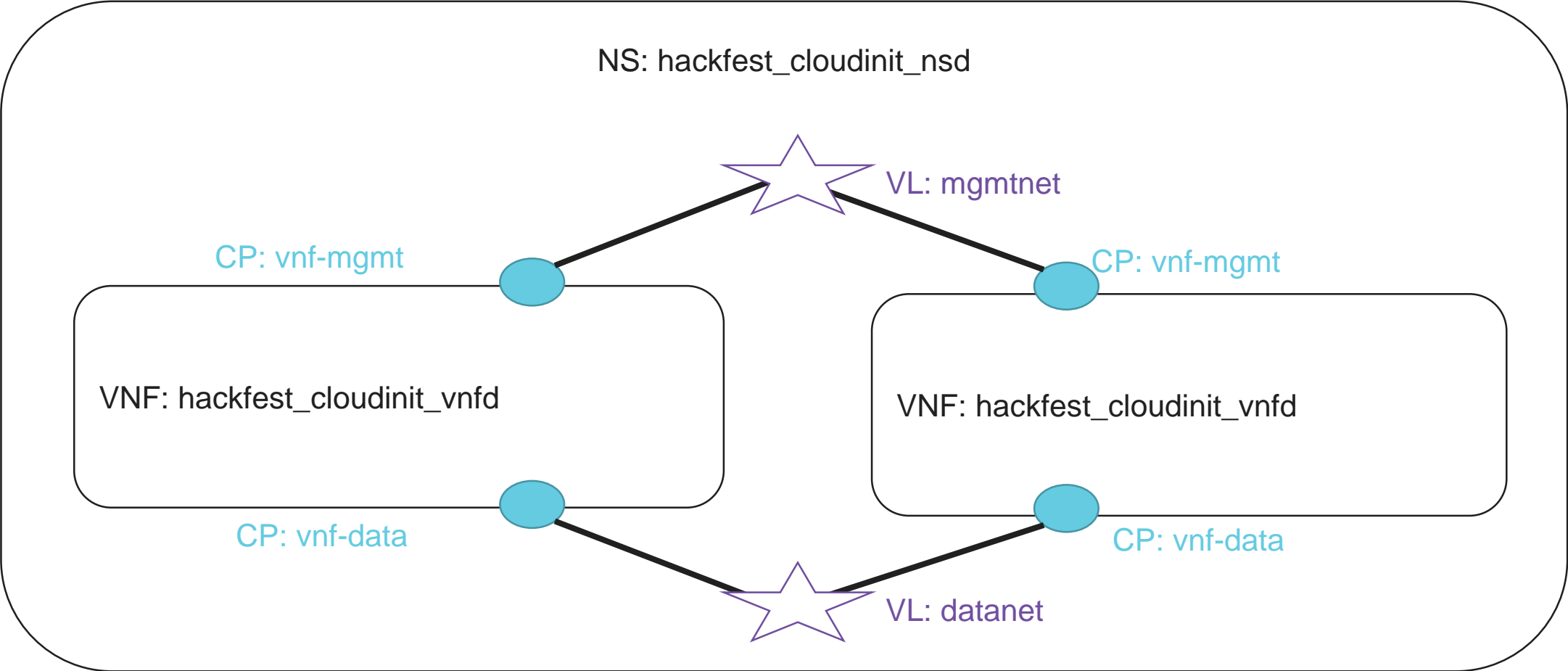
What is cloud-init and what can it be used for?

- It is a Linux package used to automate initial configuration of a VM
- VM requirements:
 - Cloud-init package
 - Cloud-init configuration (data source) via `/etc/cloud/cloud.cfg`
 - Config drive
 - Openstack metadata server
 - ...
- What can be done?
 - Setting a default locale
 - Setting an instance hostname
 - Generating instance SSH private keys
 - Adding SSH keys to a user's `.ssh/authorized_keys` so they can log in
 - Setting up ephemeral mount points
 - Configuring network devices
 - Adding users and groups
 - Adding files
- Docs: <http://cloudinit.readthedocs.io/en/latest/>

Cloud-init support in OSM

- Cloud-init is available in Linux VMs and might be supported in other OS
- Not all VIMs support cloud-init via a metadata server
- While cloud-init is supported in OSM, it is not a silver bullet

NS diagram

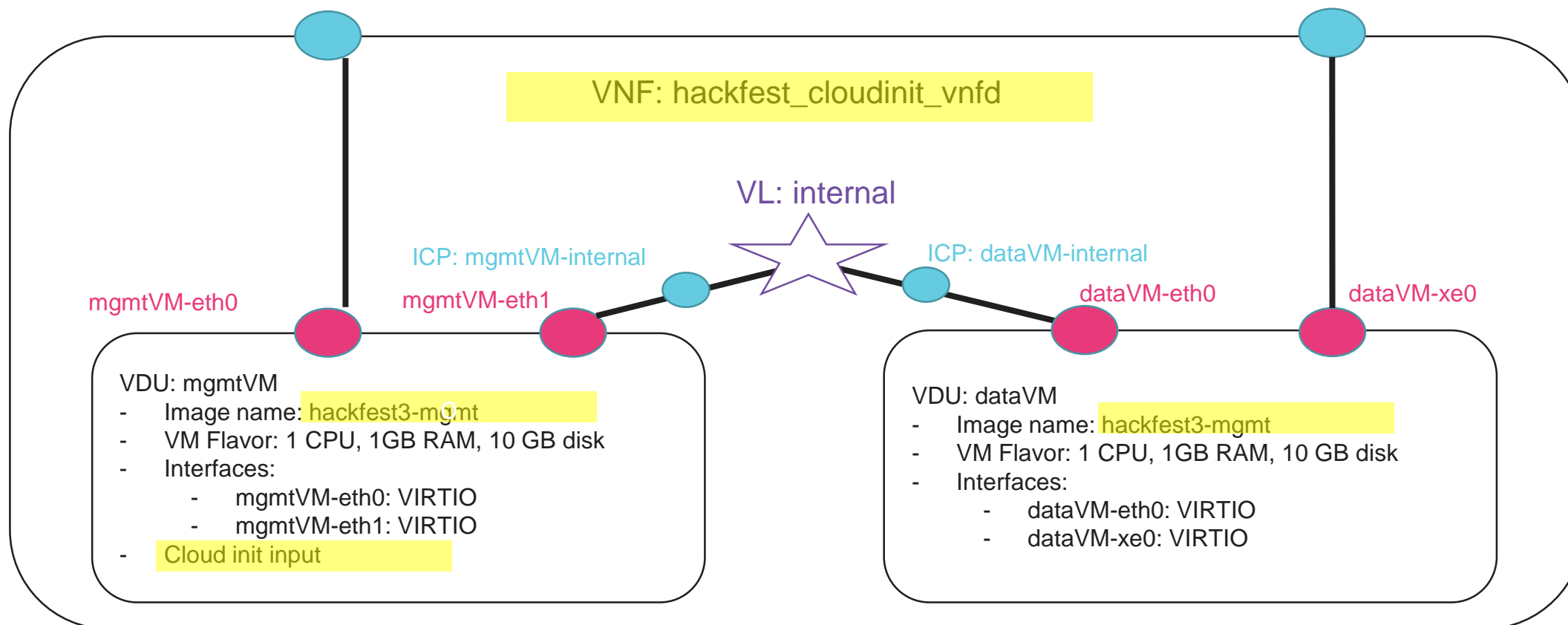


VNF diagram

Changes highlighted in yellow

External Connection point: vnf-mgmt

External Connection point: vnf-data



Creating the new CloudInit VNF (1/5)

Use the tool to create a new VNFD called "hackfest_cloudinit_vnfd":
`devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t vnfd --image
hackfest3-mgmt -c hackfest_cloudinit`

- Add 2 Connection Points (external):
 - CONNECTION POINT 1:
 - name: vnf-mgmt
 - CONNECTION POINT 2:
 - name: vnf-data
- Add new VLD 'internal' to the VNF:
 - Name: internal
 - TYPE: ELAN
 - Refer to internal CPs we will define later

Internal VLD example

```
...  
  mgmt-interface:  
    cp: vnf-mgmt  
  connection-point:  
  - id: vnf-mgmt  
    name: vnf-mgmt  
    type: VPORT  
  - id: vnf-data  
    name: vnf-data  
    type: VPORT  
  internal-vld:  
  - id: internal  
    name: internal  
    short-name: internal  
    type: ELAN  
    internal-connection-point:  
    - id-ref: mgmtVM-internal  
    - id-ref: dataVM-internal  
...
```

Creating the new CloudInit (2/5)

- Add VDU1 in the VNF
 - Name: mgmtVM
 - Image: hackfest3-mgmt
 - VM Flavor:
 - VCPU COUNT: 1
 - MEMORY MB: 1024
 - STORAGE GB: 10
 - Add 1 internal connection point:
 - ID: mgmtVM-internal
 - Name: mgmtVM-internal
 - Type: VPORT
 - Add 2 interfaces to the VDU:
 - Interface 1:
 - Name: mgmtVM-eth0
 - Position: 1
 - Connection-point-type: EXTERNAL
 - EXTERNAL-CONNECTION-POINT-REF: vnf-mgmt
 - Virtual-interface:
 - Type: VIRTIO

```
...
interface:
- name: mgmtVM-eth0
  position: '1'
  type: EXTERNAL
  virtual-interface:
    type: VIRTIO
  external-connection-point-ref: vnf-mgmt
- name: mgmtVM-eth1
  position: '2'
  type: INTERNAL
  virtual-interface:
    type: VIRTIO
  internal-connection-point-ref: mgmtVM-internal
internal-connection-point:
- id: mgmtVM-internal
  name: mgmtVM-internal
  short-name: mgmtVM-internal
  type: VPORT
...
```

- Interface 2:
 - Name: mgmtVM-eth1
 - Position: 2
 - Connection-point-type: INTERNAL
 - INTERNAL-CONNECTION-POINT-REF: mgmtVM-internal
 - Virtual-interface:
 - Type: VIRTIO

Creating the new CloudInit (3/5)

- Add VDU2 in the VNF
 - Name: dataVM
 - Image: hackfest3-mgmt
 - VM Flavor:
 - VCPU COUNT: 1
 - MEMORY MB: 1024
 - STORAGE GB: 10
 - Add 1 internal connection point:
 - ID: dataVM-internal
 - Name: dataVM-internal
 - Type: VPORT
 - Add 2 interfaces to the VDU:
 - Interface 1:
 - Name: dataVM-eth0
 - Position: 1
 - Connection-point-type: INTERNAL
 - INTERNAL-CONNECTION-POINT-REF: dataVM-internal
 - Virtual-interface:
 - Type: VIRTIO

```
...
interface:
- name: dataVM-eth0
  position: '1'
  type: INTERNAL
  virtual-interface:
    type: VIRTIO
  internal-connection-point-ref: dataVM-internal
- name: dataVM-xe0
  position: '2'
  type: EXTERNAL
  virtual-interface:
    type: VIRTIO
  external-connection-point-ref: vnf-data
internal-connection-point:
- id: dataVM-internal
  name: dataVM-internal
  short-name: dataVM-internal
  type: VPORT
...
```

- Interface 2:
 - Name: dataVM-xe0
 - Position: 2
 - Connection-point-type: EXTERNAL
 - EXTERNAL-CONNECTION-POINT-REF: vnf-data
 - Virtual-interface:
 - Type: VIRTIO

Creating the new CloudInit (4/5)

- **Modify VDU mgmtVM:**

- Cloud init input:

- Filename

- Cloud init file: cloud-config.txt

- Inside the 'vdu' list at the VNFD, put a line referring to the file inside the "cloud_init" folder of the package:

- cloud-init-file: cloud-config.txt

- **Add a new asset:**

- CLOUD_INIT:

- Upload file: cloud-config.txt

- It can be downloaded from: <https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/other/cloud-config.txt>

Creating the new CloudInit (5/5)

- Validate your descriptor using the tool:
`devops/descriptor-packages/tools/validate_descriptor.py <DESCRIPTOR_FILE>`
- Generate VNF package **(from parent folder)**
`devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t vnfd -N
<VNFD_FOLDER>`
- And finally, this is the sample file:
Hackfest Cloud Init VNF Descriptor - https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/packages/hackfest_cloudinit_vnf.tar.gz

Let's explore the Cloud-init file

- Download it from here:
 - <https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/other/cloud-config.txt>

- Content:


```
#cloud-config
password: osm4u
chpasswd: { expire: False }
ssh_pwauth: True

write_files:
- content: |
    # My new helloworld file

    owner: root:root
    permissions: '0644'
    path: /root/helloworld.txt
```



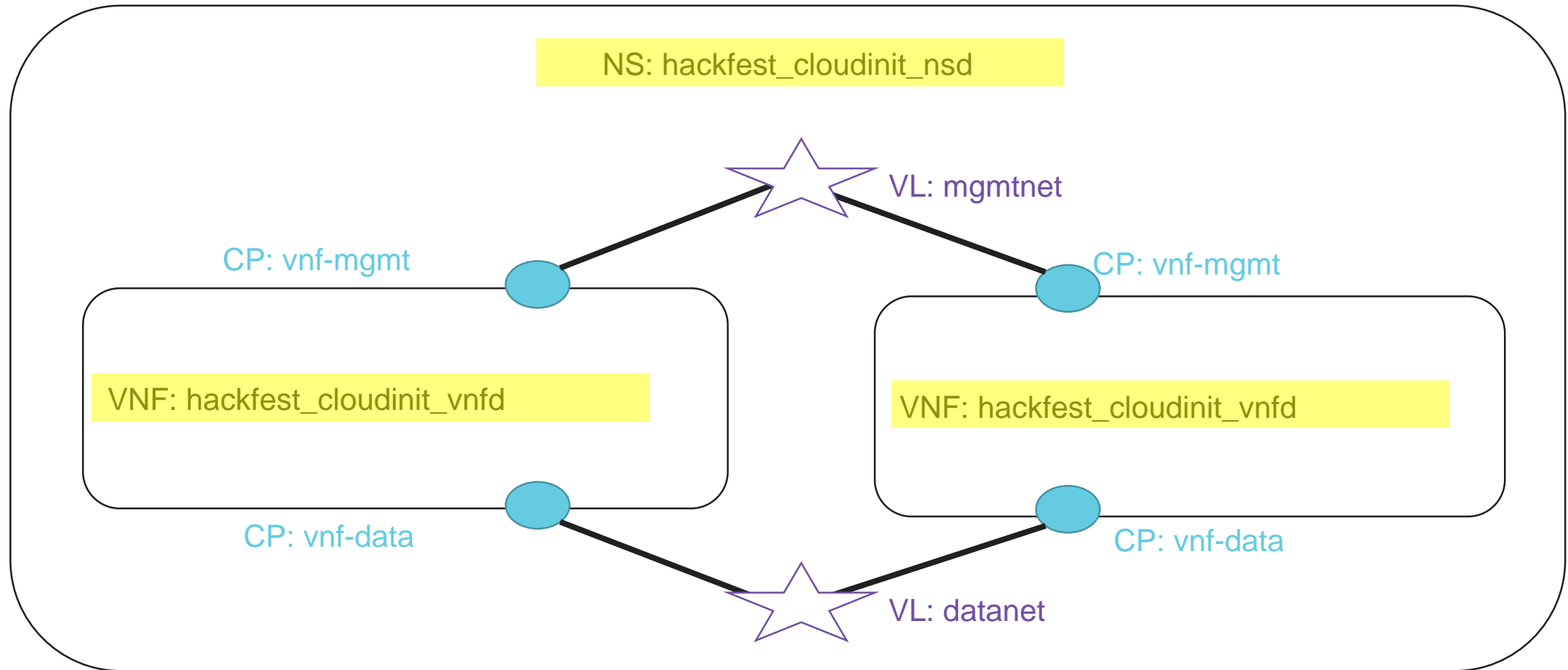
A password is added for the default user ('ubuntu').



A new file '/root/helloworld.txt' will be created at VM creation to illustrate the way this feature works.

NS diagram

Changes highlighted in yellow



Creating the NS (1/3)

Use the tool to create a new NSD called: "hackfest_cloudinit_nsd":

```
devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t nsd -c hackfest_cloudinit
```

- Specify constituent VNFs (hackfest_multivdu_vnfd)
- Add first VLD:
 - VLD1:
 - name (optional): mgmtnet
 - TYPE: ELAN
 - MGMT NETWORK: True
 - VIM NETWORK NAME
 - vim-network-name: **PUBLIC** <- This is to have a default mapped VIM network change accordingly
 - Refer VNF Connection Points to the VL:
 - vnf-mgmt → VL:mgmtnet

Creating the NS (2/3)

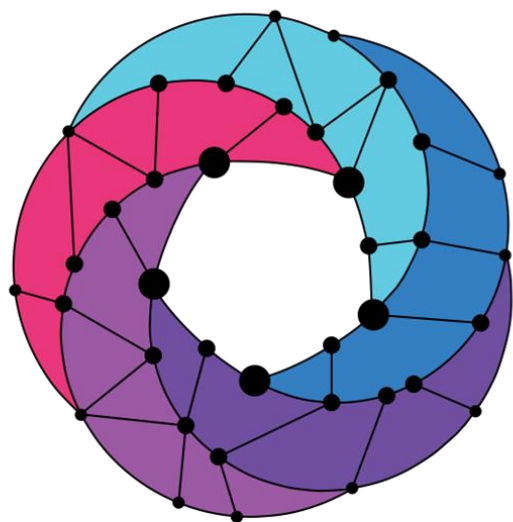
- Add second VLD:
 - VLD2:
 - name (optional): datanet
 - TYPE:ELAN
 - MGMT NETWORK: False (default)
 - Refer VNF Connection Points to the VL:
 - vnf-data → VL:datanet

Creating the NSD (3/3)

- Validate your descriptor using the tool:
`devops/descriptor-packages/tools/validate_descriptor.py <DESCRIPTOR_FILE>`
- Generate VNF package **(from parent folder)**
`devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t nsd -N
<NSD_FOLDER>`
- And finally, against the sample file:
Hackfest CloudInit NS Descriptor - https://osm-download.etsi.org/ftp/osm-5.0-five/5th-hackfest/packages/hackfest_cloudinit_ns.tar.gz

Deploying NS in the UI

- Select `hackfest_cloudinit_nsd` and instantiate it
- Complete the form
 - Add a name to the NS
 - Select the Datacenter where the NS will be deployed
 - Add SSH key
- Go to the dashboard to see the instance and get the mgmt IP address of the VNF
- Connect to each VNF:
 - `ssh ubuntu@<IP>`
- Check that the cloud-config file was executed



Open Source MANO

Find us at:

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