OSM Hackfest – Session 4
Adding day-0 configuration to VNFs
Eduardo Sousa (Canonical)
Guillermo Calviño (Altran)
Benjamín Díaz (Whitestack)
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

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

NS: hackfest_cloudinit_nsd

VNF: hackfest_cloudinit_vnfd
CP: vnf-mgmt
CP: vnf-data
VL: mgmtnet
VL: datanet

VNF: hackfest_cloudinit_vnfd
CP: vnf-mgmt
CP: vnf-data
VL: mgmtnet
VL: datanet
VNF: hackfest_cloudinit_vnfd

VDU: mgmtVM
- Image name: hackfest3-mgmt
- VM Flavor: 1 CPU, 1GB RAM, 10 GB disk
- Interfaces:
  - mgmtVM-eth0: VIRTIO
  - mgmtVM-eth1: VIRTIO
- Cloud init input

External Connection point: vnf-mgmt

VL: internal

ICP: mgmtVM-internal

External Connection point: vnf-data

VDU: dataVM
- Image name: hackfest3-mgmt
- VM Flavor: 1 CPU, 1GB RAM, 10 GB disk
- Interfaces:
  - dataVM-eth0: VIRTIO
  - dataVM-xe0: VIRTIO
Prepare the environment

• Get devops tools:
  git clone -b v6.0 https://osm.etsi.org/gerrit/osm/devops.git

• Install python-osm-im:
  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"
  sudo apt-get update
  sudo apt-get install -y python-osm-im
  sudo -H pip install pyangbind

• Get reference packages:
  • https://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/packages/hackfest_cloudinit_vnf.tar.gz
  • https://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/packages/hackfest_cloudinit_ns.tar.gz
Creating the new CloudInit VNF (1/5)

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

• Descriptor located at
  hackfest_cloudinit_vnfd/hackfest_cloudinit_vnfd.yaml

• 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
```...

```yaml
  mgmt-interface:
    cp: vnf-mgmt
car
  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 VNF (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 2:
      - Name: mgmtVM-eth1
      - Position: 2
      - Connection-point-type: INTERNAL
      - INTERNAL-CONNECTION-POINT-REF: mgmtVM-internal
      - Virtual-interface:
        - Type: VIRTIO
Creating the new CloudInit VNF (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 2:
      • Name: dataVM-xe0
      • Position: 2
      • Connection-point-type: EXTERNAL
      • EXTERNAL-CONNECTION-POINT-REF: vnf-data
      • Virtual-interface:
        • Type: VIRTIO
Creating the new CloudInit VNF (4/5)

• Download cloud-init file:
  • https://osm-download.etsi.org/ftp/osm-6.0-six/7th-hackfest/other/cloud-config.txt

• Modify VDU mgmtVM in VNFD:
  • cloud-init-file: cloud-config.txt

• Add cloud-init file to hackfest_cloudinit_vnfd/cloud_init folder
Let's explore the Cloud-init file

• Content:

```yaml
#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.
Creating the new CloudInit VNF (5/5)

• Validate your descriptor using the tool:
  
  devops(descriptor-packages/tools/validate_descriptor.py
  hackfest_cloudinit_vnfd/hackfest_cloudinit_vnfd.yaml

• Generate VNF package:
  
  devops(descriptor-packages/tools/generate_descriptor_pkg.sh -t vnfd -N
  hackfest_cloudinit_vnfd
NS: hackfest_cloudinit_nsd

VNF: hackfest_cloudinit_vnfd
CP: vnf-data
CP: vnf-mgmt
VL: mgmtnet
VL: datanet
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

• The descriptor is located at
  hackfest_cloudinit_nsd/hackfest_cloudinit_nsd.yaml

• Specify constituent VNFs:
  constituent-vnfd:
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index: '1'
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index: '2'
Creating the NS (1/3)

- Add first VLD:
  - id: mgmtnet
    name: mgmtnet
    short-name: mgmtnet
    type: ELAN
    mgmt-network: 'true'
    vim-network-name: PUBLIC
  vnfd-connection-point-ref:
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index-ref: '1'
      vnfd-connection-point-ref: vnf-mgmt
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index-ref: '2'
Creating the NS (2/3)

- Add second VLD:
  - id: datanet
    name: datanet
    short-name: datanet
    type: ELAN
    vnfd-connection-point-ref:
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index-ref: '1'
      vnfd-connection-point-ref: vnf-data
    - vnfd-id-ref: hackfest_cloudinit-vnf
      member-vnf-index-ref: '2'
      vnfd-connection-point-ref: vnf-data
Creating the NSD (3/3)

- Validate your descriptor using the tool:
  devops/descriptor-packages/tools/validate_descriptor.py
  hackfest_cloudinit_nsd/hackfest_cloudinit_nsd.yaml

- Generate NS package:
  devops/descriptor-packages/tools/generate_descriptor_pkg.sh -t nsd -N
  hackfest_cloudinit_nsd
Deploying NS in the UI

• Onboard the VNFD and NSD
• 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