Open Source MANO

OSM-MR#10 Hackfest
OSM Primitives for VNFs
Mark Beierl (Canonical)
The Big Picture

Cloud Environment

OpenStack

- Wiki
- Network Automation
- Private Network
- Virtual Desktop
- VNF Orchestration
- KNF Orchestration
- Performance Monitor
- K8s
- Monitoring
- Relations

 PN orchestration

 Firewall

 Employee

 © ETSI 2020
Your Service Architecture

Cloud Environment

OpenStack

Native Charm (Ops Code)

Virtual Network Function (VM)

Private Network

Firewall

Employee

OSM
Focus on Virtualised NF operations

OSM

LXD Operators
- Proxy Charm
- Ops Code

VNF
Native Charm

PNF
VNF
VNF
PNF

Racks

VIM

KNF
KNF

Kubernetes
Helpful Scripts

Scripts in ~/Hackfest/HD2.1-VNF-Primitives

- **virtual-pc-build.sh**
  - Builds the descriptors from source

- **virtual-pc-launch.sh**
  - OSM CLI command to launch the network service

- **virtual-pc-watch-progress.sh**
  - Command to help you watch the progress (ctrl-c to exit)

- **virtual-pc-actions.sh**
  - Lists a series of commands that we can execute later
Building the Package

cd ~/osm-packages

~/Hackfest/HD2.1-VNF-Primitives/virtual-pc-build.sh

```
osm_instructor_5@osm-jumphost:/osm-packages$ ~/Hackfest/HD2.1-VNF-Primitives/virtual-pc-build.sh

Building operator charms

Already using interpreter /usr/bin/python3
Using base prefix '/usr'
New python executable in /home/osm_instructor_5/osm-packages/hackfest_virtual-pc_vnfd/charms/virtual-pc-src/venv/bin/python3
Not overwriting existing python script /home/osm_instructor_5/osm-packages/hackfest_virtual-pc_vnfd/charms/vir
```
View Uploaded Package

### NS Packages

<table>
<thead>
<tr>
<th>Name</th>
<th>Identifier</th>
<th>Version</th>
<th>Designer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackfest_virtual-pc.ns</td>
<td>491b3f953cd40c9-81c1-10a7d4a6487a</td>
<td>1.0</td>
<td>OSM</td>
<td>Virtual Desktop Computer with MATE Desktop and RDP</td>
</tr>
</tbody>
</table>

### Product Name

<table>
<thead>
<tr>
<th>Product Name</th>
<th>Identifier</th>
<th>Version</th>
<th>Provider</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>hackfest_virtual-pc.vnf</td>
<td>d3035a95-8718-4281-991c-d3c675491f7</td>
<td>1.0</td>
<td>Canonical</td>
<td>vnfd</td>
<td>Virtual Desktop Computer with Xubuntu Desktop and RDP</td>
</tr>
</tbody>
</table>

© ETSI
Launch Service

~/.Hackfest/HD2.1-VNF-Primitives/virtual-pc-launch.sh

================================================================================================================================
Launching network service with VIMID bf5f184c-8ce0-4959-99d9-598582483b80
================================================================================================================================
dd336fb8-283d-4ae6-9d36-bb3d370d6399
================================================================================================================================
Done
================================================================================================================================
### NS Instances

<table>
<thead>
<tr>
<th>Name</th>
<th>Identifier</th>
<th>Nsd name</th>
<th>Operational Status</th>
<th>Config Status</th>
<th>Detailed Status</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>virtual-desktop</td>
<td>dd336fb8-283d-4ae6-9d36-bb3d370d6399</td>
<td>hackfest_virtual-pc_ns</td>
<td>![Status Icon]</td>
<td>![Status Icon]</td>
<td>![Status Icon]</td>
<td>![Actions Icon]</td>
</tr>
</tbody>
</table>

Stage 2/5: deployment of KDU, VMs and execution environments. 0/2. Deployed at VIM
VM Launched in OpenStack

- IP address on Private Network
- Management IP address in 172.21.18.0/24 subnet range
Progress with osm ns-show

```
"admin-status": "ENABLED",
"deployment-status": null,
"configuration-status": {
  "status": "INSTALLING SW",
  "meter_statuses": {},
  "status": {
    "status": "Maintenance",
    "agent-status": {
      "status": "executing",
      "workload-status": {
        "status": "Maintenance",
        "agent-status": {
          "status": "started",
          "instance-status": {
            "status": "running",
            "modification-status": {
              "status": "Idle",
              "meter-status": {
                "model-status": {
                  "status": "available",
                  "init": "init",
                  "stage": "Stage 2/5: deployment of KDOs, VMs and execution environments. 0/2. Deployed at VIM",
                  "vim-status": "DONE",
                  "config-status": "DONE",
                  "detailed-status": "10% Preparing libblockdev-part2 (amd64)"
                }
            },
            "VCA-status": "Maintenance",
            "status": "running",
            "status-time": "1614899418.2858396"
          }
        }
      }
    }
  }
}
```
Virtual Desktop is Ready

● Once configuration_status is “Ready”, so is your virtual desktop
● Note
  ○ this will not be reachable via the management network

```
admin-status: "ENABLED"
deploymentStatus: null
configurationStatus: [
  "status": "READY"
]
```
Add Port Forward

- Need to tell VyOS PNF to add a port forward rule
  ~/Hackfest/HD1.7-PNF/firewall-actions.sh

```
DESKTOP_IP=`osm ns-show virtual-desktop --literal | \
yq e '.vcaStatus.*.machines.0.network_interfaces.ens3.ip_addresses.0' -` 

osm ns-action firewall --vnf_name VYOS-PNF --action_name add-port-forward --params "{ruleNumber: '10', sourcePort: '3389', destinationAddress: "${DESKTOP_IP}"}, destinationPort: '3389'}"
```
Connect to Your Desktop
Perform Some Actions

~/Hackfest/HD2.1-VNF-Primitives/virtual-pc-actions.sh
Package Management

● Adds some apt packages to the desktop

```bash
osm ns-action virtual-desktop --vnf_name 1 --action_name add-package --params '{package: "ubuntu-mate-wallpapers-disco,ubuntu-mate-wallpapers-eoan"}'}
```

● Remove apt packages from the desktop

```bash
osm ns-action virtual-desktop --vnf_name 1 --action_name remove-package --params '{package: "ubuntu-mate-wallpapers-disco"}'}
```

● Update system software (apt upgrade)

```bash
osm ns-action virtual-desktop --vnf_name 1 --action_name update-system
```
Snap Management

- Adds a classic confinement snap to the desktop

```
osm ns-action virtual-desktop --vnf_name 1 --action_name add-snap --params '{package: "code --classic"}'
```

- Remove snap from the desktop

```
osm ns-action virtual-desktop --vnf_name 1 --action_name remove-snap --params '{package: "code"}'
```
Desktop Maintenance

- Reboot the desktop computer
  
  osm ns-action virtual-desktop --vnf_name 1 --action_name reboot

- Announce a message
  
  osm ns-action virtual-desktop --vnf_name 1 --action_name announce --params '{message: "Hello from the Hackfest!"}'}
Discuss

- What is a VNF?
- How does OSM model VNFs?
- How does a VNF get launched?
- What is a native charm and how is it different from proxy?
- How do actions get run?