Onboarding Virtual Network Functions

You can onboard any new VNF on OpenStack and VMware vCenter. To onboard the VNF, you must fulfill the prerequisites, and prepare the deployment data model. This chapter describes the prerequisites and the procedure to prepare the deployment data model on OpenStack and VMware vCenter.

Onboarding Virtual Network Functions on OpenStack

You must fulfill the following prerequisites before onboarding VNFs on OpenStack:

- The VNF image formats must be compatible with OpenStack, for example qcow2 format. The image can be onboarded on OpenStack either by the OpenStack glance client, or by ESC using the NETCONF or REST APIs.

- The day 0 configuration file passed into the VM must be compatible with either the OpenStack config-drive or the user-data, so that the VMs can use the day 0 configuration details for bootstrap mechanism.

- The day 0 variables must be in plain text format and use the predefined day 0 variables, so that the VMs can use the static IP information available in the day 0 file.

Preparing the Deployment Data Model

You must prepare the deployment data model as part of VNF onboarding. The deployment data model is an XML file (template) that describes the operational behavior such as resource requirements, networking, monitoring KPI, placement policies, lifecycle stages (LCS), scaling rules and so on.

Preparing the Data Model for OpenStack Deployment

The VNF deployment data model is an XML file or template describing the resource requirements, networking, day zero configuration, and other service operational behaviors such as monitoring KPI, placement policies, lifecycle stages, scaling rules and so on.

To onboard a VNF and define the VNF services in the deployment data model, you must:

1. Prepare the VM Resources

2. Describe the VNF Networking

3. Prepare the Day Zero Configuration

4. Define the operational behaviors such as metrics and KPIs, in the deployment data model

Preparing the VM Resources

The deployment data model refers to resources such as tenants, images, flavors, volumes and so on to deploy the VNFs. You can either create these resources using ESC, or use the preexisting resources already available on OpenStack. For more information, see Managing Resources Overview.

A sample data model with the resources is as follows:

```
<?xml version="1.0" encoding="ASCII"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
     <name>vnf_tenant</name>
     <deployments>
      <deployment>
      ...
    <name>vnf-dep</name>
      <vm_group>
       <name>Grp1</name>
       <flavor>vnf_flavor</flavor>
       <image>vnf_image</image>
      ...
      </vm_group>
      </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

Describing the VNF Network

The deployed VMs in the VNF must connect to specific networks for different purposes. These networks could be the management network, the internal networks within VMs, and so on. Make sure these networks are either available on OpenStack, or created by ESC. You must define these networks in the deployment data model to create them during deployment. For more information, see Managing Networks.

A sample deployment data model showing how to create networks and subnetworks, and specify the network connection for the VM interfaces is as follows:

```
<deployment>
        <name>vnf-dep</name>
        ...
        <networks>
          <network>
           <name>vnf_net</name>
           <shared>false</shared>
           <admin_state>true</admin_state>
           <subnet>
            <name>vnf_subnet</name>
            <ipversion>ipv4</ipversion>
            <dhcp>true</dhcp>
            <address>10.101.1.0</address>
            <netmask>255.255.255.0</netmask>
            <gateway>10.101.1.1</gateway>
           </subnet>
          </network>
        </networks>
        ...
```

```
        </deployment>

    </deployments>

      <vm_group>
       <name>Grp1</name>
       ...
        <interfaces>
        <interface>
         <nicid>0</nicid>
          <network>vnf_management</network>
         </interface>
          <interface>
           <nicid>1</nicid>
           <network>vnf_net</network>
          </interface>
         </interfaces>
        ...
       </vm_group>
```

Preparing the Day Zero Configuration

As part of the Day Zero configuration, the day zero file is passed into the VNF at the time of installation for bootstrapping. The day zero file is described in the deployment data model. For more information, see Day Zero Configuration.

A sample describing the day zero file as config drive and user data is as follows:

```
<config_data>
        <configuration>
         <dst>--user-data</dst>
          <file>file://var/qatest/test-script.sh</file>
           </configuration>
           <configuration>
         <dst>/etc/configure-networking.sh</dst>
          <file>file://var/qatest/configure-networking.sh</file>
         </configuration>
        </config_data>
```

Defining the Operational Behavior

To onboard composite VNFs, you must configure some of the operational behaviors such as network connections, monitoring KPIs, placement policies, lifecycle stages, scaling rules and so on. These behaviors can be described in the deployment data model. For more information, see Configuring Deployment Parameters.

Once you have prepared the deployment data model with these details, you have onboarded the VNF and instantiated the VNF service on OpenStack. Now you can deploy the VNF. When the VNF is deployed, ESC applies the day zero configuration for the new service. For more information, see Deploying Virtual Network Functions on OpenStack.

For information on preparing the VNFs on VMware vCenter, see Preparing the Data Model for VMware vCenter Deployment.

Onboarding Virtual Network Functions on VMware vCenter

The following prerequisites must be fulfilled before onboarding VNFs on VMware vCenter:

- The VNF image format must be compatible with VMware vCenter, for example ova.

- The day 0 configuration file passed into the VM must be compatible with either the ovf properties or reading configurations from the CDROM drive.

- The day 0 variables must be in plain text format on the CDROM drive.

Preparing the Data Model for VMware vCenter Deployment

The VNF deployment data model is an XML file or template describing the resource requirements, networking, day zero configuration, and other operational behaviors such as monitoring KPIs, placement policies, lifecycle stages, scaling rules and so on.

To onboard a VNF and define the VNF services in the deployment data model, you must:

1. Prepare the VM Resources

2. Describe the VNF Networking

3. Prepare the Day Zero Configuration

4. Define the operational behaviors such as metrics and KPIs, in the deployment data model

Preparing the VM Resources

The deployment data model refers to resources to deploy the VNFs. An image (template) is the only resource referred in a VMware deployment. The image can be a pre-existing image, or created by ESC.

Note Tenants do not exist in a VMware vCenter deployment, but the default admin tenant is still required in the deployment data model.

A sample data model with image details are as follows:

```
<?xml version="1.0" encoding="ASCII"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
     <name>admin</name>
     <deployments>
     <deployment>
     ...
     <name>vnf-dep</name>
      <vm_group>
        <image>vnf_image</image>
        ...
     </vm_group>
      </deployment>
      </deployments>
    </tenant>
  </tenants>
 </esc_datamodel>
```

On VMware vCenter, the placement policies and volume details are necessary for each vm_group. A zone_host type placement defines the target computing host or the cluster for a deployment. The volume defines the target data store for the deployment. The following deployment data model defines a deployment target to the computing-cluster cluster1 and allows ESC to choose a data store automatically.

```
<?xml version="1.0" encoding="ASCII"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
...
        <name>vnf-dep</name>
          <vm_group>
            ...
 <placement>
  <type>zone_host</type>
  <zone>cluster1</zone>
  </placement>
      <volumes>
<volume>
<name>auto-select</name>
<volid>1</volid>
</volume>
</volumes>
    </vm_group>
    </deployment>
     </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

The following deployment data model defines a deployment target to the computing-host host1 and data store datastore1.

```
<?xml version="1.0" encoding="ASCII"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>admin</name>
      <deployments>
        <deployment>
...
        <name>vnf-dep</name>
          <vm_group>
            ...
  <placement>
  <type>zone_host</type>
  <host>host1</host>
  </placement>
   <volumes>
     <volume>
      <name>datastore1</name>
      <volid>1</volid>
  </volume>
  </volumes>
   </vm_group>
    </deployment>
     </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

Describing the VNF Network

The deployed VMs in the VNF must connect to specific networks for different purposes. Those networks could be the management network, the internal networks among VMs and other networks for different purposes.

On VMware, a network refers to vDS port group, and a subnet refers to the IP pool under vCenter. ESC supports only static IP for VMware deployment. Make sure those networks are available on VMware vCenter, or created by ESC. To create a network during deployment, you can define the network in the deployment data model. The deployment data model is as follows:

```
<deployment>
  <name>vnf-dep</name>
  ...
 <networks>
    <network>
      <name>vnf_management</name>
      <admin_state>true</admin_state>
      <number_of_ports>8</number_of_ports>
      <shared>false</shared>
      <switch_name>vdswitch1</switch_name>
      <vlan_id>0</vlan_id>
      <subnet>
        <name>vnf_management-subnet</name>
        <ipversion>ipv4</ipversion>
        <dhcp>false</dhcp>
        <address>10.0.0.10</address>
        <netmask>255.255.255.0</netmask>
        <gateway>10.0.0.1</gateway>
      </subnet>
    </network>
  </networks>
  ...

  </deployment>
</deployments>
```

> **Note** On VMware Vcenter, the nicid value starts from 1. On OpenStack the nicid value starts from 0.

```
<vm_group>
        <name>Grp1</name>
        ...
        <interfaces>
          <interface>
           <nicid>1</nicid>
           <network>vnf_management</network>
          </interface>
          <interface>
           <nicid>2</nicid>
           <network>vnf_net</network>
          </interface>
        </interfaces>
        ...
      </vm_group>
```

Preparing the Day Zero Configuration

As part of the day 0 configuration, the day 0 file is passed into the VNF at the time of installation for bootstrapping. The day 0 files have to be described in the deployment data model. For more information, see Day Zero Configuration. The sample day zero file shows the day zero configurations passed in as files in CDROM content attached to the deployed VM.

```
<config_data>
        <configuration>
          <dst>day0-config</dst>
          <file>http://somehost:80/day0.txt</file>
```

```
          </configuration>
          <configuration>
            <dst>idtoken</dst>
            <file>http://somehost:80/idtoken.txt</file>
          </configuration>
       </config_data>
```

The sample below shows day 0 configurations passed through the ofv settings.

```
<config_data>
          <configuration>
          <dst>ovfProperty:mgmt-ipv4-addr</dst>
          <data>$NICID_1_IP_ADDRESS/16</data>
          </configuration>
          <configuration>
          <dst>ovfProperty:com.cisco.csr1000v:hostname</dst>
          <data>$HOSTNAME</data>
          <variable>
           <name>HOSTNAME</name>
           <val>csrhost1</val>
           <val>csrhost2</val>
          </variable>
         </configuration>
        </config_data>
```

Defining the Operational Behaviors

To onboard composite VNFs, you must configure some of the operational behaviors such as network connections, monitoring KPIs, placement policies, lifecycle stages, scaling rules and so on. These behaviors can be described in the deployment data model. For more information, see Configuring Deployment Parameters.

Once you have prepared the deployment data model with these details, you have onboarded the VNF and instantiated the VNF service on OpenStack. Now you can deploy the VNF. When the VNF is deployed, ESC applies the day zero configuration for the new service. For more information, see Deploying Virtual Network Functions on VMware vCenter.

For information on preparing the VNFs on OpenStack, see "Preparing the Data Model for VMware vCenter Deployment".