Deploying Virtual Network Functions on VMware vCenter

- Deploying Virtual Network Functions on VMware vCenter, on page 1
- Deploying VNFs on Single VMware vCenter VIM, on page 2
- Deploying Virtual Network Functions on VMware vCloud Director (vCD), on page 6

Deploying Virtual Network Functions on VMware vCenter

This section describes the deployment scenario for Elastic Services Controller (ESC) and the procedure to deploy VNFs on VMware. You can deploy VNFs using out-of-band image definitions. The following table lists the deployment scenarios:

Scenarios	Description	data model templates	Images	Advantages
Deploying VNFs on a single VIM by creating Images through ESC **Important** Images are also referred to as Templates on VMware vCenter.	The process of VNF deployment is as follows: 1. VNF Deployment- The *deployment data model* refers to the images created and then deploys VNFs.	• *deployment data model* • *image data model*	Images are created through ESC using REST APIs.	• The images can be used in multiple VNF deployments. • You can add or delete image definitions through ESC.

Scenarios	Description	data model templates	Images	Advantages
Deploying VNFs on a single VIM using out-of-band images	1. VNF Deployment- The *deployment data model* refers to the out-of-band images on VMware vCenter and then deploys VNFs.	• *deployment data model* • Image on VMware vCenter	Images cannot be created or deleted through ESC.	• The images can be used in multiple VNF deployments. • You can view images through ESC portal. • During out-of -band deployment, you can choose images.

Note ESC does not support IPv6 deployment for VIM type VMware vSphere.

Deploying VNFs on Single VMware vCenter VIM

The VNF deployment is initiated as a service request either originating from the ESC portal or the northbound interfaces. The service request comprises of XML payloads. ESC supports the following deployment scenarios:

- Deploying the VNFs by creating resources through ESC
- Deploying the VNFs using out-of-band resources

Before you deploy the VNFs, you must ensure that the resources are available on VMware vCenter, or you must create these resources. See Managing Resources Overview. During a deployment, ESC looks for the deployment details in the deployment data model. For more information on the deployment data model, see Cisco Elastic Services Controller Deployment Attributes.

Note Deploying VNFs on multiple VIMs is not supported on VMware vCenter.

Note A single ESC instance only supports one vCenter Distributed Switch (vDS):

- A vDS contains one or many ESXi hosts that are clustered.
- If the ESXi hosts are under one compute cluster, the VMware vCenter HA and DRS capabilities must be disabled.
- Clustered Data stores are not supported.
- If the hosts are clustered, only flat data stores under the cluster or under the datacenter are supported.

ESC only supports a default resource pool. You cannot add or create resource pools. When you see the error message "Networking Configuration Operation Is Rolled Back and a Host Is Disconnected from vCenter Server", it is due to a vCenter's limitation. The auto-select for datastore works as follows:

- ESC selects a host first. If deployment is cluster targeted, host will be selected based on the ratio of number of VMs against computing-host's capacity. Otherwise, host is selected as requested for host targeted deployment.
- From the host, datastore is picked based on its free space.

After every redeploy as part of recovery on VMware vCenter, the VM's interface(s) will have different MAC addresses.

Passing OVF Properties to a VM

As a part of deploying a VNF on VMware vCenter, you can pass the name value pair as OVF property to the VM. To pass these configurations while deploying a VNF, you must include additional arguments in the *deployment data model* template.

A sample configuration is as follows:

```
   <esc_datamodel ...>
    ...
    <config_data>
   <configuration>
      <dst>ovfProperty:mgmt-ipv4-addr</dst>
      <data>$NICID_1_IP_ADDRESS/24</data>
   </configuration>
   <configuration>
      <dst>ovfProperty:com.cisco.csr1000v:hostname</dst>
      <data>$HOSTNAME</data>
      <variable>
         <name>HOSTNAME</name>
         <val>csrhost1</val>
         <val>csrhost2</val>
      </variable>
   </configuration>
</config_data>
     ...
   </esc_datamodel>
```

Deploying VNFs on Multiple Virtual Data Centers (Multi-VDCs)

A Virtual Data Center (VDC) combines virtual resources, operational details, rules, and policies to manage specific group requirements. A group can manage multiple VDCs, images, templates, and policies. This group can allocate quotas and assign resource limits for individual groups at the VDC level.

To view the list of VDCs that are available and on the ESC portal, choose **Datacenters**.

Before you Begin

Before you deploy VNFs on multiple VDCs, ensure that the following conditions are met:

- Verify that a standard external network spanning both VDCs is available for the ESC to ping the deployed VMs.
- Verify that at least one management interface on the VMs is connected to the external network.
- Verify that the VDC is present in the vCenter.

Note

- ESC assumes all required resources to be created in VDC are out of band and present in the VDC.
- Currently, ESC can deploy in any VDC present in a vCenter. There is no scoping or restriction of VDCs that ESC can deploy in.

When you deploy a VNF, you must specify the virtual datacenter locator name on which the VNF needs to be provisioned.

A locator element is introduced in deployment request to create and delete resources.

The locator element contains:

- a datacenter name tag—to specify the target VDC for the resource (Deployment, Image, Network and Subnets).
- switch_name—to specify the target VDS to associate the network with.

Using the locator element,

- An image or a template can be created on another VDC by providing the datacenter attribute within the locator. For example,

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
    <images>
        <image>
            <name>automated-uLinux</name>
            <src>http://VAR_FILE_SERVER_IP/share/images/uLinux/uLinux.ovf</src>
            <locators>
                <datacenter>VAR_VDC2</datacenter>
            </locators>
        </image>
    </images>
</esc_datamodel>
```

- A network can be created and deleted from a VDC.

Note If the network is part of unified deployment, then the datacenter attribute is taken from the deployment attribute in deployment request.

```
<network>
    <locators>
        <datacenter>DC-03</datacenter>
        <switch_name>dvSwitch</switch_name>
    </locators>
    <name>test-yesc-net-u</name>
    <shared>false</shared>
    <admin_state>true</admin_state>
</network>
```

Cisco Elastic Services Controller Portal allows you to choose the VDC on which the VM is provisioned. When you are creating a service request, you can choose the VDC on which this VM is provisioned. For more information on deploying VNFs on a VDC, see .

The *default_locators* container in ESC operational data shows default locators configured in ESC.

Note The *default_locators* container is not displayed if there are no locators configured.

Sample operational data is as follows:

```
Operational Data
/opt/cisco/esc/confd/bin/netconf-console --port=830 --host=172.16.0.1 --user=admin
--privKeyFile=/var/confd/homes/admin/.ssh/confd_id_dsa --privKeyType=dsa --get -x
"esc_datamodel/opdata"
<?xml version="1.0" encoding="UTF-8"?><rpc-reply
xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
    <data>
        <esc_datamodel xmlns="http://www.cisco.com/esc/esc">
            <opdata>
                <status>OPER_UP</status>
                <stats>
                    <hostname>test-ESC-host</hostname>
                    <os_name>Linux</os_name>
                    <os_release>2.6.32-573.22.1.el6.x86_64</os_release>
                    <arch>amd64</arch>
                    <uptime>9481</uptime>
                    <cpu>
                        <cpu_num>4</cpu_num>
                    </cpu>
                </stats>
                <system_config>
                    <active_vim>VMWARE</active_vim>
                    <vmware_config>
                        <vcenter_ip>172.16.1.0</vcenter_ip>
                        <vcenter_port>80</vcenter_port>
                        <vcenter_username>root</vcenter_username>
                    </vmware_config>
                </system_config>
                <default_locators>
                    <datacenter>DC-4</datacenter>
                </default_locators>
                <tenants>
```

```
                    <tenant>
                        <name>admin</name>
                        <tenant_id>SystemAdminTenantId</tenant_id>
                    </tenant>
                </tenants>
            </opdata>
        </esc_datamodel>
    </data>
</rpc-reply>
[admin@test-ESC-host esc-cli]$
```

Deploying Virtual Network Functions on VMware vCloud Director (vCD)

This section describes the deployment scenario for Elastic Services Controller (ESC) and the procedure to deploy VNFs on VMware vCloud Director (vCD). To install ESC on vCD, see the *Cisco Elastic Services Controller Install and Upgrade Guide*.

Resources such as organization, and organization VDC and so on must be created on vCD before deployment. For more information, see Managing Resources on vCloud Director (vCD).

Note ESC supports VMware vCloud Director 8.2.

To deploy the VNF, you must:

1. Add a VIM connector, with the organization and organization user details preconfigured in the VMware vCD. See VIM Connector Configuration for VMware vCloud Director (vCD).

   The vim_vdc leaf under the locator refers to the vDC, the deployment is targeted to.

2. Deploy the VNF with organization VDC, catalog and vApp template parameters preconfigured in the VMware vCD.

See the VMware vCloud Director Documentation to create these resources.

You must set the following key parameters, before deploying the VNFs on vCD:

- VMWARE_VCD_PARAMS—Specify the *VMWARE_VCD_PARAMS* parameter in the extensions section of the datamodel under each deployment section. The *VMWARE_VCD_PARAMS* parameter includes *CATALOG_NAME* and *VAPP_TEMPLATE_NAME*.
- CATALOG_NAME—Specify the name of the preconfigured catalog that contains references to vApp templates and the media images.
- VAPP_TEMPLATE_NAME—Specify the name of the preconfigured vApp template that contains virtual machine image that is loaded with an operating system, application, and data, it ensure that virtual machines are consistently configured across an entire organization.

A sample deployment is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc" xmlns:ns0="http://www.cisco.com/esc/esc"
 xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
```

```
xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns3="http://www.cisco.com/esc/esc_notifications">
   <tenants>
      <tenant>
         <!-- ESC scope tenant -->
         <name>esc-tenant</name>
         <vim_mapping>false</vim_mapping>
         <deployments>
            <deployment>
               <!-- vApp instance name -->
               <name>vapp-inst1</name>
               <policies>
                  <placement_group>
                     <name>placement-anti-affinity</name>
                     <type>anti_affinity</type>
                     <enforcement>strict</enforcement>
                     <vm_group>g1</vm_group>
                     <vm_group>g2</vm_group>
                  </placement_group>
               </policies>
               <extensions>
                  <extension>
                     <name>VMWARE_VCD_PARAMS</name>
                     <properties>
                        <property>
                           <name>CATALOG_NAME</name>
                           <value>catalog-1</value>
                        </property>
                        <property>
                           <name>VAPP_TEMPLATE_NAME</name>
                           <value>uLinux_vApp_Template</value>
                        </property>
                     </properties>
                  </extension>
               </extensions>
               <vm_group>
                  <name>g1</name>
                  <locator>
                     <!-- vCD vim connector id -->
                     <vim_id>vcd_vim</vim_id>
                     <!-- vCD organization corresponding to the vim connector -->
                     <vim_project>organization</vim_project>
                     <!-- vDC pre-preconfigured in organization -->
                     <vim_vdc>VDC-1</vim_vdc>
                  </locator>
                  <!-- VM name in vAppTemplate -->
                  <image>vm-001</image>
                  <bootup_time>150</bootup_time>
                  <recovery_wait_time>30</recovery_wait_time>
                  <interfaces>
                     <interface>
                        <nicid>0</nicid>
                        <network>MgtNetwork</network>
                        <ip_address>172.16.0.0</ip_address>
                     </interface>
                  </interfaces>
                  <scaling>
                     <min_active>1</min_active>
                     <max_active>1</max_active>
                     <elastic>true</elastic>
                     <static_ip_address_pool>
                        <network>MgtNetwork</network>
                        <ip_address>172.16.0.0</ip_address>
                     </static_ip_address_pool>
```

```
    </scaling>
    <kpi_data>
       <kpi>
          <event_name>VM_ALIVE</event_name>
          <metric_value>1</metric_value>
          <metric_cond>GT</metric_cond>
          <metric_type>UINT32</metric_type>
          <metric_collector>
             <type>ICMPPing</type>
             <nicid>0</nicid>
             <poll_frequency>3</poll_frequency>
             <polling_unit>seconds</polling_unit>
             <continuous_alarm>false</continuous_alarm>
          </metric_collector>
       </kpi>
    </kpi_data>
    <rules>
       <admin_rules>
          <rule>
             <event_name>VM_ALIVE</event_name>
             <action>"ALWAYS log"</action>
             <action>"TRUE servicebooted.sh"</action>
             <action>"FALSE recover autohealing"</action>
          </rule>
       </admin_rules>
    </rules>
    <config_data>
       <configuration>
          <dst>ovfProperty:mgmt-ipv4-addr</dst>
          <data>$NICID_0_IP_ADDRESS/24</data>
       </configuration>
    </config_data>
 </vm_group>
 <vm_group>
    <name>g2</name>
    <locator>
       <!-- vCD vim connector id -->
       <vim_id>vcd_vim</vim_id>
       <!-- vCD organization corresponding to the vim connector -->
       <vim_project>organization</vim_project>
       <!-- vDC pre-preconfigured in organization -->
       <vim_vdc>VDC-1</vim_vdc>
    </locator>
    <!-- VM name in vAppTemplate -->
    <image>vm-002</image>
    <bootup_time>150</bootup_time>
    <recovery_wait_time>30</recovery_wait_time>
    <interfaces>
       <interface>
          <nicid>0</nicid>
          <network>MgtNetwork</network>
          <ip_address>172.16.0.1</ip_address>
       </interface>
    </interfaces>
    <scaling>
       <min_active>1</min_active>
       <max_active>1</max_active>
       <elastic>true</elastic>
       <static_ip_address_pool>
          <network>MgtNetwork</network>
          <ip_address>172.16.0.1</ip_address>
       </static_ip_address_pool>
    </scaling>
    <kpi_data>
```

```
                        <kpi>
                           <event_name>VM_ALIVE</event_name>
                           <metric_value>1</metric_value>
                           <metric_cond>GT</metric_cond>
                           <metric_type>UINT32</metric_type>
                           <metric_collector>
                              <type>ICMPPing</type>
                              <nicid>0</nicid>
                              <poll_frequency>3</poll_frequency>
                              <polling_unit>seconds</polling_unit>
                              <continuous_alarm>false</continuous_alarm>
                           </metric_collector>
                        </kpi>
                     </kpi_data>
                     <rules>
                        <admin_rules>
                           <rule>
                              <event_name>VM_ALIVE</event_name>
                              <action>"ALWAYS log"</action>
                              <action>"TRUE servicebooted.sh"</action>
                              <action>"FALSE recover autohealing"</action>
                           </rule>
                        </admin_rules>
                     </rules>
                     <config_data>
                        <configuration>
                           <dst>ovfProperty:mgmt-ipv4-addr</dst>
                           <data>$NICID_0_IP_ADDRESS/24</data>
                        </configuration>
                     </config_data>
                  </vm_group>
               </deployment>
            </deployments>
         </tenant>
      </tenants>
   </esc_datamodel>
```