Managing Existing Deployments

After a deployment is created successfully, the resources within a deployment can be updated. As part of deployment management, you can add or delete resources, or update the configuration of the existing resources. These updates can be made in a running deployment. This chapter describes managing these resources in detail.

Updating an Existing Deployment

You can update an existing deployment by adding new VM groups, interfaces, networks, and so on. You can also update the day-0 configuration, KPIs and Rules for the VM groups. You can add or delete a vm_group, add or delete an ephemeral network in a vm_group, and add or delete an interface in a VM group after successful deployment.

On OpenStack, you can perform all the updates such as add or delete a vm_group, ephemeral network vm_group, and an interface in a single deployment.

During a service update, auto-recovery actions may drive the service to an inconsistent state. To prevent triggering of auto-recovery actions, monitors are disabled before the service update workflow, and enabled after the update is complete.

Updating an existing deployment is supported both on OpenStack and VMware vCenter. The table below lists the components that can be updated in an existing deployment.

Table 1: Updating an Existing Deployment on OpenStack, VMware vCenter and vCloud Director

Update	OpenStack	VMware vCenter	vCloud Director
Adding a VM group	Supported	Supported	Supported
Deleting a VM group	Supported	Supported	Supported
Deleting VM groups when the service is in error state	Supported	Supported	Not supported
Adding an ephemeral network	Supported	Not supported	Not supported

Update	OpenStack	VMware vCenter	vCloud Director
Deleting an ephemeral network	Supported	Not supported	Not supported
Adding an interface	Supported	Not supported	Not supported
Deleting an interface	Supported	Not supported	Not supported
Updating an interface	Supported	Supported	Not supported
Adding a Static IP pool	Supported	Not supported	Not supported
Deleting a Static IP pool	Supported	Not supported	Not supported
Updating the day-0 config in a VM group	Supported	Supported	Not supported
Updating the KPIs and rules	Supported	Supported	Not supported
Updating the number of VMs (Scale In or Scale Out) in a VM group	Supported	Supported	Not supported
Updating the recovery wait time	Supported	Supported	Not supported
Updating the recovery policy	Supported	Not supported	Not supported
Updating an image	Supported	Not supported	Not supported

Note Updating an existing deployment on multiple OpenStack VIMs is also supported. However, the locator attribute within the vm group cannot be updated. For more information on Deploying VMs on Multiple VIMs, see Deploying VNFs on Multiple OpenStack VIMs.

Adding a VM Group

You can add or delete a vm_group from a running deployment using the existing images and flavors.

NETCONF request to add a vm_group:

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc"> <tenants><tenant>
  <name>Admin</name>
   <deployments>
    <deployment>
     <deployment_name>NwDepModel_nosvc</deployment_name>

     <vm_group>
        <image></image>
        <Flavor></Flavor>
        .........
     </vm_group>
```

```
        <vm_group>
   <image></image>
 <Flavor></Flavor>
 .........
        </vm_group>
        <vm_group>
 <image></image>
 <Flavor></Flavor>
 .........
        </vm_group>
    </deployment>
</deployments>
      </tenant></tenants>
</esc_datamodel>
```

NETCONF notification upon successful addition of a VM Group:

```
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
          VM_DEPLOYED
            VM_ALIVE
          SERVICE_UPDATED
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
```

Deleting a VM Group

NETCONF request to delete a vm_group:

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
 <tenants><tenant>
  <name>Admin</name>
   <deployments>
    <deployment>
     <deployment_name>NwDepModel_NoSvc</deployment_name>

     <vm_group>
        <image></image>
        <Flavor></Flavor>
        .........
     </vm_group>
     <vm_group nc:operation="delete">
 <image></image>
 <Flavor></Flavor>
 .........
    </vm_group>
    <vm_group nc:operation="delete">
 <image></image>
 <Flavor></Flavor>
 .........
    </vm_group>
   </deployment>
</deployments>
     </tenant></tenants>
</esc_datamodel>
```

NETCONF notification upon successful deletion of vm_group:

```
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
          VM_UNDEPLOYED
          SERVICE_UPDATED
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
```

Deleting VM Groups in Error State

You can now delete vm groups when the deployment is in error state by performing a deployment update. However, additional configurations to the vm groups such as adding one or more vm groups, or changing the attribute value of a different vm group while deleting a particular vm group are not allowed.

Adding an Ephemeral Network in a VM Group

You can add an ephemeral network in a vm_group using the existing images and flavors.

NETCONF request to add an ephemeral in a vm_group:

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc"> <tenants><tenant>
  <name>Admin</name>
   <deployments>
    <deployment>
     <deployment_name>NwDepModel_nosvc</deployment_name>
      <networks>
      <network>
.........
</network>
<network>
.........
</network>
<network>
.........
</network>
      </networks>
      <vm_group>
         <image></image>
         <Flavor></Flavor>
         .........
      </vm_group>
   </deployment>
</deployments>
     </tenant></tenants>
</esc_datamodel>
```

NETCONF notification upon successful addition of an ephemeral network in a vm_group:

```
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
          CREATE_NETWORK
          CREATE_SUBNET
         SERVICE_UPDATED
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
```

Deleting an Ephemeral Network in a VM Group

NETCONF request to delete an ephemeral network in a vm_group

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc"> <tenants><tenant>
  <name>Admin</name>
   <deployments>
    <deployment>
     <deployment_name>NwDepModel</deployment_name>
      <networks>
      <network nc:operation="delete">
.........
</network>
<network>
.........
</network>
<network nc:operation="delete">
.........
```

```
 </network>
      </networks>
      <vm_group>
         <image></image>
         <Flavor></Flavor>
         .........
      </vm_group>
   </deployment>
</deployments>
     </tenant></tenants>
</esc_datamodel>
```

NETCONF notification upon successful deletion of an ephemeral network in a vm_group:

```
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
         DELETE_SUBNET
         DELETE_NETWORK
         SERVICE_UPDATED
UPDATE SERVICE REQUEST RECEIVED (UNDER TENANT)
```

Adding an Interface in a VM Group (OpenStack)

You can add an interface in a vm_group from a running deployment using the existing images and flavors.

NETCONF request to add an interface in a vm_group:

```
<interfaces>
                <interface>
                  <nicid>0</nicid>
                  <network>my-network</network>
                </interface>
                 <interface>
                  <nicid>1</nicid>
                  <network>utr-net</network>
                </interface>
                 <interface>
                  <nicid>2</nicid>
                  <network>utr-net-1</network>
                </interface>
 </interfaces>
```

Note　ESC Release 2.3 and later supports adding and deleting interfaces using the ESC Portal for OpenStack.

ESC supports adding and deleting interfaces from a vm_group using both REST and NETCONF APIs.

Deleting an Interface in a VM Group (OpenStack)

NETCONF request to delete an interface in a vm_group:

```
<interfaces>
             <interface>
              <nicid>0</nicid>
               <network>my-network</network>
             </interface>
               <interface>
                <nicid>1</nicid>
                <network>utr-net</network>
               </interface>
               <interface nc:operation="delete">
                <nicid>2</nicid>
                <network>utr-net-1</network>
```

```
                    </interface>
                </interfaces>
```

You can simultaneously add and delete interfaces in a VM group (OpenStack only) in the same deployment request.

Note   ESC does not support the following:

- Updating the properties of an existing vm_group, network or subnet.

- Updating the image and flavor of a vm_group.

- Blank names for resource names (that is, vm_group, network, subnet or Interface).

In Cisco ESC Release 2.0 or earlier, the ephemeral networks or subnets can only be added or deleted.

ESC does not support the day 0 configuration of new interfaces added during a deployment update. You must perform additional configuration separately in the VNF as part of the day-n configuration. If you delete an interface with token replacement, you must update the day 0 configuration to remove that interface. In future, ESC will use the new day 0 configuration for recovery.

A new interface without the nic ids is not configured during a deployment update.

New interfaces with existing day 0 configuration are configured.

Updating an Interface (OpenStack)

Updating an interface on OpenStack deletes the previous interface and creates a new one with the existing nic id.

The datamodel is as follows:

```
<interfaces>
                <interface>
                  <nicid>0</nicid>
                  <network>my-network</network>
                </interface>
                 <interface>
                  <nicid>1</nicid>
                  <network>utr-net-2</network>
                </interface>
        </interfaces>
```

A VM_UPDATED notification is sent with the details of all the interfaces in a VM, followed by a SERVICE_UPDATED notification after the workflow is updated.

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2015-07-25T00:45:27.64+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>VM has been updated successfully. vm:
utr-80__7515__utr-80__utr-80utr-80utr-801.2__0__utr-80__0</status_message>
    <svcname>utr-80</svcname>
    <svcversion>1.2</svcversion>
    <depname>utr-80</depname>
    <tenant>utr-80</tenant>
    <svcid>c1294ad1-fd7b-4a73-8567-335160dce90f</svcid>
    <depid>ecedf755-502c-473a-82f2-db3a5485fdf5</depid>
```

```
<vm_group>utr-80</vm_group>
<vm_source>
  <vmid>4b20024f-d8c8-4b1a-8dbe-3bf1011a0bcb</vmid>
  <hostid>71c7f3afb281485067d8b28f1734ec6b63f9e3225045c581168cc39d</hostid>
  <hostname>my-server</hostname>
  <interfaces>
    <interface>
      <nicid>0</nicid>
      <port_id>6bbafbf5-51a1-48c0-a4a5-cd6092657e5c</port_id>
      <network>7af5c7df-6246-4d53-91bd-aa12a1607656</network>
      <subnet>7cb6815e-3023-4420-87d8-2b10efcbe14e</subnet>
      <ip_address>192.168.0.10</ip_address>
      <mac_address>fa:16:3e:bc:07:d5</mac_address>
      <netmask>255.255.255.0</netmask>
      <gateway>192.168.0.1</gateway>
    </interface>
    <interface>
      <nicid>1</nicid>
      <port_id>6d54d3a8-b793-40b8-9a32-c7e2f08e0917</port_id>
      <network>4f85613a-d3fc-4b49-9cb0-b91d4360918b</network>
      <subnet>c3724a64-ffed-43b6-aba8-63287c5344ea</subnet>
      <ip_address>10.91.90.2</ip_address>
      <mac_address>fa:16:3e:49:d0:00</mac_address>
      <netmask>255.255.255.0</netmask>
      <gateway>10.91.90.1</gateway>
    </interface>
    <interface>
      <nicid>3</nicid>
      <port_id>04189123-fc7a-4418-877b-61c24a5e8508</port_id>
      <network>f9c7978f-800e-4bfc-bc20-1c29acef87d9</network>
      <subnet>63ae5e39-c41a-4b28-9ac7-ed94b5e477b0</subnet>
      <ip_address>172.16.0.97</ip_address>
      <mac_address>fa:16:3e:5e:2e:e3</mac_address>
      <netmask>255.240.0.0</netmask>
      <gateway>172.16.0.1</gateway>
    </interface>
  </interfaces>
</vm_source>
<vm_target>
</vm_target>
<event>
  <type>VM_UPDATED</type>
</event>
</escEvent>
</notification>
```

Note
- Interfaces are unique based on nic ids. If new interfaces are added, they should have different nic ids. If an interface is edited, and has the same nic id, it is considered as an update to the existing interface.

Updating an Interface (VMware vCenter)

You can update a network associated with an interface, while updating an existing deployment. Replace the old network name with a new name in the deployment request to update the network. The port group on the interfaces is updated for all VMs in the VM group during the network update.

> **Note** IP update is not supported during an interface update on VMware vCenter.
>
> Static IP and mac pool updates are not supported during an interface update on VMware vCenter when min > 1 in a vm group.

The datamodel update is as follows:

Existing datamodel:

```
<interface>
  <nicid>1</nicid>
  <network>MgtNetwork</network>
</interface>
```

New datamodel:

```
<interface>
  <nicid>1</nicid>
  <network>VNFNetwork</network>
</interface>
```

The following notification is received after successful update:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-08-17T12:03:12.518+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Updated 1 interface: [net=VNFNetwork,nicid=1]</status_message>
    <depname>u1-asa</depname>
    <tenant>admin</tenant>
    <tenant_id>SystemAdminTenantId</tenant_id>
    <depid>90139aa1-9705-4b07-9963-d60691d3b0ad</depid>
    <vm_group>utr-asa-1</vm_group>
    <vm_source>
      <vmid>50261fbc-88a0-8601-71a9-069460720d4f</vmid>
      <hostid>host-10</hostid>
      <hostname>172.16.103.14</hostname>
      <interfaces>
        <interface>
          <nicid>1</nicid>
          <type>virtual</type>
          <port_id/>
          <network>VNFNetwork</network>
          <subnet/>
          <ip_address>192.168.0.254</ip_address>
          <mac_address>00:50:56:a6:d8:1d</mac_address>
        </interface>
      </interfaces>
    </vm_source>
    <vm_target>
  </vm_target>
    <event>
      <type>VM_UPDATED</type>
    </event>
  </escEvent>
</notification>
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-08-17T12:03:12.553+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
```

```
      <status>SUCCESS</status>
      <status_code>200</status_code>
      <status_message>Service group update completed successfully</status_message>
      <depname>u1-asa</depname>
      <tenant>admin</tenant>
      <tenant_id>SystemAdminTenantId</tenant_id>
      <depid>90139aa1-9705-4b07-9963-d60691d3b0ad</depid>
      <vm_source>
   </vm_source>
      <vm_target>
   </vm_target>
      <event>
         <type>SERVICE_UPDATED</type>
      </event>
   </escEvent>
</notification>
```

Adding a Static IP Pool

You can add a new static IP pool to the existing deployment.

NETCONF request to add a static IP pool:

```
<scaling>
<min_active>2</min_active>
<max_active>5</max_active>
<elastic>true</elastic>
<static_ip_address_pool>
<network>IP-pool-network-A</network>
<ip_address_range>
<start>172.16.5.13</start>
<end>172.16.5.13</end>
</ip_address_range>
</static_ip_address_pool>
<static_ip_address_pool>
<network>IP-pool-network-B</network>
<ip_address_range>
<start>172.16.7.13</start>
<end>172.16.7.13</end>
</ip_address_range>
</static_ip_address_pool>
</scaling>
```

Deleting a Static IP Pool

You can delete the existing IP pools in a running deployment.

NETCONF request to delete a static IP pool:

```
<scaling>
<min_active>2</min_active>
<max_active>5</max_active>
<elastic>true</elastic>
<static_ip_address_pool>
<network>IP-pool-network-A</network>
<ip_address_range>
<start>172.16.5.13</start>
<end>172.16.5.13</end>
</ip_address_range>
</static_ip_address_pool>
<static_ip_address_pool nc:operation="delete">
<network>IP-pool-network-B</network>
<ip_address_range>
<start>172.16.7.13</start>
```

```
<end>172.16.7.13</end>
</ip_address_range>
</static_ip_address_pool>
</scaling>
```

Note
- You cannot update an already existing static IP pool in an existing deployment. You can only add a new static IP pool, or delete if the static IP pool is not in use.

- You cannot update the IP address of an interface. That is, you cannot deploy with one IP address, and then add a new IP in the same nic id.

The following scenarios are supported or rejected because of the dependencies within the static IP pools, interfaces, and networks.

Request	Supported or Rejected
Add or delete new static IP pools in single or different requests.	Supported
Add interfaces with static IP.	Supported
Add an interface and the corresponding IP pool in the same request.	Supported
Delete an interface, retaining the corresponding IP pool.	Supported
Delete an interface and its corresponding IP pool in the same request.	Supported
Delete an IP pool, when one of its IPs are being used in an interface in a VM.	Rejected
Add a network, and a static IP pool having different network in a single request.	Supported
To an existing network, add a corresponding interface and an IP pool in the same update.	Supported
Add a new network in an update, and a new corresponding IP pool in the next update.	Supported
Add an IP pool without corresponding network.	Rejected
Delete a network and the referencing IP pool in the same request, when none of the IPs are being used in any interfaces.	Supported
Delete a network which is being used in an IP pool and interface.	Rejected
To an existing network, add an interface and an IP pool in the same update.	Supported

Request	Supported or Rejected
Delete an IP pool that does not have any IPs used in interface, though the network with subnet is present.	Supported
Add an IP pool which already exists.	Request is accepted by NETCONF but no action taken
Update the IP addresses of an existing IP pool.	Rejected

Updating the Day 0 Configuration in a VM Group

To update (add, delete or change) the day-0 configuration of a VM group in an existing deployment, edit-config the deployment and update the configuration under config_data. The new day-0 config file is only applied on future deployment, which is triggered by either VM recovery (that is undeploy/deploy) or scale-out.

Note To change the existing day-0 config file, the URL or path must be specified. This enables ESC to detect the change that has occurred in the configuration.

In the example below, if a VM ALIVE event is not received, you can change the action from triggering auto recovery to simply logging the event.

Existing configuration:

```
<config_data>
  <configuration>
    <dst>WSA_config.txt</dst>

<file>https://172.16.73.167:4343/day0/cfg/vWSA/node/001-wsa/provider/Symphony_VNF_P-1B/file>

  </configuration>
  <configuration>
    <dst>license.txt</dst>

<file>https://172.16.73.167:4343/day0/cfg/vWSA/node/001-wsa/provider/Symphony_VNF_P-1B/wsa-license.txt</file>

  </configuration>
</config_data>
```

New configuration:

```
<config_data>
  <configuration>
    <dst>WSA_config.txt</dst>

<file>https://172.16.73.167:4343/day0/cfg/vWSA/node/001-wsa/provider/Symphony_VNF_P-1B/file>

  </configuration>
  <configuration>
    <dst>license.txt</dst>

<file>https://172.16.73.167:4343/day0/cfg/vWSA/node/002-wsa/provider/Symphony_VNF_P-1B/wsa-license.txt</file>

  </configuration>
</config_data>
```

SERVICE_UPDATED notification is received after updating the configuration.

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-05-05T00:35:15.359+00:00</eventTime>
```

```
   <escEvent xmlns="http://www.cisco.com/esc/esc">
     <status>SUCCESS</status>
     <status_code>200</status_code>
     <status_message>Service group update completed successfully</status_message>
     <depname>900cd7554d31-5454000964474c1cbc07256792e63240-cloudvpn</depname>
     <tenant>Symphony_VNF_P-1B</tenant>
     <tenant_id>3098b55808e84484a4f8bab2160a41a7</tenant_id>
     <depid>b7d566ce-1ee6-4147-8c23-c8bcb5d05fd4</depid>
     <vm_source/>
     <vm_target/>
     <event>
       <type>SERVICE_UPDATED</type>
     </event>
   </escEvent>
</notification>
```

For more information on day-0 configuration, see Day Zero Configuration.

Updating the KPIs and Rules

ESC allows updating KPIs and rules for a VM in the existing deployment. Edit the datamodel to update the KPIs and rules section.

For example, to change the *Polling Frequency* in an existing deployment, update the *<poll_frequency>* element in the KPI section of the datamodel.

Change *<poll_frequency>3</poll_frequency>* to *<poll_frequency>20</poll_frequency>* in the sample below.

```
   <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>
```

Similarly, the existing rules can be updated for a VM. For example, to switch off the auto- recovery on a boot failure and to log the action, update *<action>FALSE recover autohealing</action>* to *<action>FALSE log</action>* in the sample below.

```
   <rules>
              <admin_rules>
                <rule>
                  <event_name>VM_ALIVE</event_name>
                  <action>ALWAYS log</action>
                  <action>FALSE recover autohealing</action>
                  <action>TRUE servicebooted.sh</action>
                </rule>
               ...
    ...
            </rules>
```

✎

Note
- During the KPIs or rules update, auto-recovery does not happen as the monitors are unset. Auto-recovery happens when the monitors are reset in the deployment.

- The *event_name* cannot be modified during an update. It can only be added or deleted.

For more information on KPIs and Rules, see the KPIs and Rules Section.

Updating the Number of VMs in a Deployment (Updating Manual Scale In/ Scale Out)

You can add and remove VMs from an existing deployment by changing the min_active and max_active values in the scaling section of the datamodel. This alters the size of the initial deployment.

In the example below, the deployment has an initial count of 2 VMs, which can scale out to 5 VMs.

```
 <esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:ns3="http://www.cisco.com/esc/esc_notifications"
xmlns:ns0="http://www.cisco.com/esc/esc" xmlns="http://www.cisco.com/esc/esc">
  <version>1.0.0</version>
     . . .
    <vm_group>
       </interfaces>
         <interface>
           <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
           <nicid>1</nicid>
           <ip_address>10.0.0.0</ip_address>
         </interface>
       </interfaces>
     <scaling>
        <min_active>2</min_active>
        <max_active>5</max_active>
        <elastic>true</elastic>


         . . .
```

The example below creates an additional 8 VMs bringing the number of active VMs up to a minimum of 10. See the table below for more scenarios.

```
<esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:ns3="http://www.cisco.com/esc/esc_notifications"
xmlns:ns0="http://www.cisco.com/esc/esc" xmlns="http://www.cisco.com/esc/esc">
  <version>1.0.0</version>
     . . .
    <vm_group>
       </interfaces>
         <interface>
           <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
           <nicid>1</nicid>
           <ip_address>10.0.0.0</ip_address>
         </interface>
       </interfaces>
     <scaling>
        <min_active>10</min_active>
        <max_active>15</max_active>
        <elastic>true</elastic>
        <static_ip_address_pool>
           <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
           <gateway>192.168.0.1</gateway> <!-- not used -->
```

```
            <netmask>255.255.255.0</netmask> <!-- not used -->
            <ip_address>10.0.0.0</ip_address>
        </static_ip_address_pool>
    </scaling>
```

The table below shows some more scenarios on updating the minimum and maximum values in the scaling section.

Table 2: Updating the Number of VMs in a Deployment

Scenario	Old Value	New Value	Active Value
If the initial number of VMs are a minimum of 2 and maximum of 5 in the scaling section, updating the minimum number of VMs to 3 would create one additional VM. This assumes that the active number of VMs remains at 2.	The old minimum number of VMs is 2.	The new minimum number of VMs is 3.	The active number of VMs is 2.
If the initial number of VMs is a minimum value of 2 and maximum value of 5, then updating the minimum value to 3 would update the database but will not impact the deployment. This scenario will occur if the original deployment has scaled creating one additional VM.	The old minimum value is 2.	The new minimum value is 3.	The active count is 3.
If the initial number of VMs is a minimum of 2 and maximum of 5, then updating the minimum value to 1 will update the database but will not impact the deployment. Having an active number of VMs greater than the minimum value is a valid deployment as the number of active VMs falls within the minimum or maximum range.	The old minimum value is 2.	The new minimum value is 1.	The active number of VMs is 2.

Scenario	Old Value	New Value	Active Value
If the initial number of VMs is a minimum of 2 and maximum of 5, then updating the maximum to 6 will update the database but will not impact the deployment. Having an active number of VMs lesser than the maximum value is a valid deployment as the number of active VMs falls within the minimum or maximum range.	The old maximum value is 5.	The new maximum value is 6.	The active number of VMs is 2.
If the initial number of VMs is a minimum of 2 and maximum of 5, then updating the maximum value to 4 will update the database but will not have any impact on the deployment. Having an active VM count lesser than the maximum value is a valid deployment as the number of active VMs falls within the minimum or maximum range.	The old maximum value is 5.	The new maximum value is 4.	The active number of VMs is 2.
If the initial number of VMs is a minimum of 2 and maximum of 5, then updating the maximum number of VMs to 4 will update the database and remove one VM from the deployment. The last VM created will be removed bringing the active and maximum count down to 4.	The old maximum value is 5.	The new maximum value is 4.	The active number of VMs is 4.

If static IPs are used, adding more VMs to a deployment needs update to the scaling pool section.

The deployment datamodel is as follows:

```
<esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:ns3="http://www.cisco.com/esc/esc_notifications"
xmlns:ns0="http://www.cisco.com/esc/esc" xmlns="http://www.cisco.com/esc/esc">
  <version>1.0.0</version>
```

```
      . . .
  <vm_group>
     </interfaces>
       <interface>
         <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
         <nicid>1</nicid>
         <ip_address>23.23.23.23</ip_address>
       </interface>
     </interfaces>
     <scaling>
       <min_active>1</min_active>
       <max_active>1</max_active>
       <elastic>true</elastic>
       <static_ip_address_pool>
         <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
         <gateway>192.168.0.1</gateway> <!- not used ->
         <netmask>255.255.255.0</netmask> <!- not used ->
         <ip_address>23.23.23.23</ip_address>
       </static_ip_address_pool>
     </scaling>
```

Pools are linked to interfaces through network id. The updated datamodel is as follows:

```
Update payload
<esc_datamodel xmlns:ns2="urn:ietf:params:xml:ns:netconf:notification:1.0"
xmlns:ns1="urn:ietf:params:xml:ns:netconf:base:1.0"
xmlns:ns3="http://www.cisco.com/esc/esc_notifications"
xmlns:ns0="http://www.cisco.com/esc/esc" xmlns="http://www.cisco.com/esc/esc">
  <version>1.0.0</version>
      . . .
    <vm_group>
       <interfaces>
         <interface>
           <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
           <nicid>1</nicid>
           <ip_address>23.23.23.23</ip_address>
         </interface>
       </interfaces>
       <scaling>
         <min_active>2</min_active>
         <max_active>2</max_active>
         <elastic>true</elastic>
         <static_ip_address_pool>
             <network>1fbf9fc2-3074-4ae6-bb0a-09d526fbada6</network>
             <gateway>192.168.0.1</gateway>
             <netmask>255.255.255.0</netmask>
             <ip_address>10.0.0.0</ip_address>
             <ip_address>10.0.0.24</ip_address>
         </static_ip_address_pool>
       </scaling>
```

The first IP is also included in the update datamodel. If a value is not present in the update list it will be removed from the pool. This results in creating a single VM using the IP address 10.0.0.24.

✎

Note   You cannot remove a specific VM from the deployment.

Updating the Recovery Wait Time

You can now update the recovery wait time in an existing deployment. In the example below, the <recovery_wait_time> parameter is set to 60 seconds during the initial deployment.

```
<vm_group>
 <name>CSR</name>
 <recovery_wait_time>60</recovery_wait_time>
```

The recovery wait time is updated to 100 seconds in the existing deployment.

```
<vm_group>
 <name>CSR</name>
 <recovery_wait_time>100</recovery_wait_time>
```

Updating the recovery wait time impacts the VMs created in the existing deployment.

After receiving a VM_DOWN event, recovery wait time allows ESC to wait for a certain amount of time before proceeding with the VM recovery workflow. The time allocated for recovery wait time allows the VM to restore network connectivity or heal itself. If a VM_ALIVE is triggered within this time, VM recovery is canceled.

Updating the Recovery Policy

You can add the recovery policy, or update the existing recovery policy parameters while updating a deployment.

Auto recovery is triggered automatically without notification. For manual recovery, the VM_MANUAL_RECOVERY_NEEDED notification is sent, and the recovery starts only if the user sends command.

When the recovery type is set to auto, the recovery starts automatically without notification. When the recovery type is set to manual, the VM_MANUAL_RECOVERY_NEEDED notification is sent, and the recovery starts only if the user sends command.

In the example below, the recovery action is set to REBOOT_THEN_REDEPLOY during initial deployment. It is updated to REBOOT_ONLY during the deployment update. If the recovery is not successful, the maximum number of retries is 1 in the initial deployment. You can update the maximum retries as well in an existing deployment. In the example below, the maximum number of retries is updated to 3.

Initial Deployment

```
<recovery_policy>
        <action_on_recovery>REBOOT_THEN_REDEPLOY</action_on_recovery>
            <max_retries>1</max_retries>
        </recovery_policy>
```

Deployment Update

```
<recovery_policy>
        <action_on_recovery>REBOOT_ONLY</action_on_recovery>
            <max_retries>3</max_retries>
        </recovery_policy>
```

The recovery policy notification is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2017-06-21T12:35:12.354+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>Service group update completed successfully</status_message>
    <depname>jenkins-update-recovery-success-dep-201102</depname>
    <tenant>jenkins-update-recovery-success-tenant-201102</tenant>
    <tenant_id>11ade63bac8a4010a969df0d0b91b9bf</tenant_id>
    <depid>574b2e11-61a9-4d9b-83b1-e95a3aa56fdd</depid>
    <event>
```

```
        <type>SERVICE_UPDATED</type>
    </event>
  </escEvent>
</notification>
```

During the deployment update, a recovery policy cannot be overwritten with LCS. For example, a recovery policy with REBOOT_ONLY cannot be overwritten with lifecycle stage (LCS).

Updating an Image

You can update the image reference of VMs in an existing deployment.

The datamodel update is as follows:

Existing datamodel:

```
<recovery_wait_time>30</recovery_wait_time>
<flavor>Automation-Cirros-Flavor</flavor>
<image>Automation-Cirros-Image</image>
```

New datamodel:

```
<recovery_wait_time>30</recovery_wait_time>
<flavor>Automation-Cirros-Flavor</flavor>
<image>Automation-CSR-Image-3_14</image>
```

You receive a service update notification after the image is updated.

```
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-05-10T17:34:00.605+00:00</eventTime>
<escEvent xmlns="http://www.cisco.com/esc/esc">
<status>SUCCESS</status>
<status_code>200</status_code>
<status_message>Service group update completed successfully</status_message>
<depname>ud-A</depname>
<tenant>ut-AM</tenant>
<tenant_id>24e21e581ad441ebbb3bd22e69c36322</tenant_id>
<depid>e009b1cc-0aa9-4abd-8aac-265be7f9a80d</depid>
<event>
<type>SERVICE_UPDATED</type>
</event>
</escEvent>
</notification>
```

The new image reference appears in the opdata:

```
<vm_group>
<name>ug-1</name>
<flavor>m1.large</flavor>
<image>cirror</image>
<vm_instance>
<vm_id>9a63afed-c70f-4827-91e2-72bdd86c5e39</vm_id>
```

If an incorrect image name is provided, then the following error appears:

```
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
<eventTime>2018-05-08T19:28:12.321+00:00</eventTime>
```

```
<escEvent xmlns="http://www.cisco.com/esc/esc">
<status>FAILURE</status>
<status_code>500</status_code>
<status_message>Error during service update: Failed to [Update] deployment: The image
Automation-1-Cirros-Image cannot be found on the virtual infrastructure
manager.</status_message>
<depname>ud-A</depname>
<tenant>ut-AL</tenant>
<tenant_id>4fb19d82c5b34b33aa6162c0b33f07d7</tenant_id>
<depid>6eed6eba-4f3f-401d-83be-91d703ee4946</depid>
<event>
<type>SERVICE_UPDATED</type>
</event>
</escEvent>
</notification>
```

Rollback scenarios for Image Update

You must update the image reference even when the service is in error state so that the image reference gets updated in the subsequent update. The table below lists the image update rollback conditions, the expected behavior and notifications.

Rollback condition	Expected behavior	Notification

The service is in the ERROR state, and the request has image update only	The image is updated but the service remains in the ERROR state	`?xml version="1.0" encoding="UTF-8"?`
		`<notification`
		`xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">`
		`<eventTime>2018-06-06T13:59:04.331+00:00</eventTime>`
		`<escEvent xmlns="http://www.cisco.com/esc/esc">`
		`<status>SUCCESS</status>`
		`<status_code>200</status_code>`
		`<status_message>Deployment update successful. But one or more VMs are still in ERROR state.</status_message>`
		`<depname>ud-A</depname>`
		`<tenant>ut-JJ</tenant>`
		`<tenant_id>0dbb67d6457642f68520565ce785976a</tenant_id>`
		`<depid>0feea6bc-310c-49c8-8416-94f89a324bfb</depid>`
		`<event>`
		`<type>SERVICE_UPDATED</type>`
		`</event>`
		`</escEvent>`
		`</notification>`

Service is in ERROR state and the request is sent to remove the VM group (in error)	The VM group is removed and the service is in ACTIVE state	

Rollback condition	Expected behavior	Notification
The service is in ERROR state. A request to remove the VM group (in error) is sent along with an image update request in the same VM group	The VM group should be removed. There is no impact due to the image update. The service is back to ACTIVE state	

The service is in ERROR state. A request to remove the VM groups (in active) is sent along with the image update in a different VM group (in error)	The VM group (in active) is removed. The image updated in the vm group (in error). the service remains in the ERROR state.	`?xml version="1.0" encoding="UTF-8"?`
		`<notification`
		`xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">`
		`<eventTime>2018-06-06T13:59:04.331+00:00</eventTime>`
		`<escEvent xmlns="http://www.cisco.com/esc/esc">`
		`<status>SUCCESS</status>`
		`<status_code>200</status_code>`
		`<status_message>Deployment update successful. But one or more VMs are still in ERROR state.</status_message>`
		`<depname>ud-A</depname>`
		`<tenant>ut-JJ</tenant>`
		`<tenant_id>0dbb67d6457642f68520565ce785976a</tenant_id>`
		`<depid>0feea6bc-310c-49c8-8416-94f89a324bfb</depid>`
		`<event>`
		`<type>SERVICE_UPDATED</type>`
		`</event>`
		`</escEvent>`
		`</notification>`

Rollback condition	Expected behavior	Notification
The service is in the ERROR state. A single VM group is present (in error). The image update request is sent.	The image is updated but the service remains in the ERROR state. The VM group (in error) cannot be removed, as it is the only one in the service. User must undeploy and redeploy.	

Adding a VM Group (vCloud Director)

ESC supports only addition and deletion of VM group(s) in vCD. One or multiple VM group(s) can be added or deleted in a service update.

```xml
<?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>vnf-dep</name>
      <vim_mapping>false</vim_mapping>
      <deployments>
        <deployment>
          <!-- vApp instance name -->
          <name>dep</name>
          <policies>
            <placement_group>
              <name>placement-affinity-1</name>
              <type>affinity</type>
              <enforcement>strict</enforcement>
              <vm_group>g1</vm_group>
              <vm_group>g2</vm_group>
              <vm_group>g3</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_id>
                    <!-- vCD orgnization -->
                    <vim_project>esc</vim_project>
                    <!-- vDC name -->
                    <vim_vdc>VDC-1</vim_vdc>
                </locator>
                <!-- VM name in vAppTemplate -->
                <image>vm-001</image>
                <bootup_time>120</bootup_time>
                <recovery_wait_time>5</recovery_wait_time>
                <interfaces>
                    <interface>
                      <nicid>0</nicid>
                      <network>MgtNetwork</network>
                      <ip_address>10.0.0.155</ip_address>
                      <mac_address>00:1C:B3:09:85:15</mac_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>10.0.0.155</ip_address>
                    </static_ip_address_pool>
                    <static_mac_address_pool>
                      <network>MgtNetwork</network>
                      <mac_address>00:1C:B3:09:85:15</mac_address>
                    </static_mac_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>30</poll_frequency>
                        <polling_unit>seconds</polling_unit>
                        <continuous_alarm>false</continuous_alarm>
                      </metric_collector>
                    </kpi>
                </kpi_data>
                <rules>
                    <admin_rules>
```

```xml
                    <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>
                <recovery_policy>
                  <action_on_recovery>REBOOT_ONLY</action_on_recovery>
                </recovery_policy>
          </vm_group>
          <vm_group>
            <name>g2</name>
            <locator>
              <!-- vCD vim connector id -->
              <vim_id>vcd</vim_id>
              <!-- vCD orgnization -->
              <vim_project>esc</vim_project>
              <!-- vDC name -->
              <vim_vdc>VDC-1</vim_vdc>
            </locator>
            <!-- VM name in vAppTemplate -->
            <image>vm-002</image>
            <bootup_time>120</bootup_time>
            <recovery_wait_time>5</recovery_wait_time>
            <interfaces>
              <interface>
                <nicid>0</nicid>
                <network>MgtNetwork</network>
                <ip_address>10.0.0.156</ip_address>
                <mac_address>00:1C:B3:09:85:16</mac_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>10.0.0.156</ip_address>
              </static_ip_address_pool>
              <static_mac_address_pool>
                <network>MgtNetwork</network>
                <mac_address>00:1C:B3:09:85:16</mac_address>
              </static_mac_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>30</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>
              <recovery_policy>
                <action_on_recovery>REBOOT_ONLY</action_on_recovery>
              </recovery_policy>
            </vm_group>
            <vm_group>
              <name>g3</name>
              <locator>
                <!-- vCD vim connector id -->
                <vim_id>vcd</vim_id>
                <!-- vCD orgnization -->
                <vim_project>esc</vim_project>
                <!-- vDC name -->
                <vim_vdc>VDC-1</vim_vdc>
              </locator>
              <!-- VM name in vAppTemplate -->
              <image>vm-002</image>
              <bootup_time>120</bootup_time>
              <recovery_wait_time>5</recovery_wait_time>
              <interfaces>
                <interface>
                  <nicid>0</nicid>
                  <network>MgtNetwork</network>
                  <ip_address>20.0.0.157</ip_address>
                  <mac_address>00:1C:B3:09:85:17</mac_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>10.0.0.157</ip_address>
                </static_ip_address_pool>
                <static_mac_address_pool>
                  <network>MgtNetwork</network>
                  <mac_address>00:1C:B3:09:85:17</mac_address>
                </static_mac_address_pool>
              </scaling>
              <kpi_data>
                <kpi>
                  <event_name>VM_ALIVE</event_name>
                  <metric_value>1</metric_value>
                  <metric_cond>GT</metric_cond>
```

```xml
                            <metric_type>UINT32</metric_type>
                            <metric_collector>
                              <type>ICMPPing</type>
                              <nicid>0</nicid>
                              <poll_frequency>30</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>
                        <recovery_policy>
                          <action_on_recovery>REBOOT_ONLY</action_on_recovery>
                        </recovery_policy>
                      </vm_group>
                    </deployment>
                  </deployments>
              </tenant>
          </tenants>
</esc_datamodel>
```

Deleting a VM Group (vCloud Director)

ESC allows deleting a VM group in vCloud Director:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc" xmlns:nc="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>vnf-dep</name>
      <vim_mapping>false</vim_mapping>
      <deployments>
        <deployment>
          <!-- vApp instance name -->
          <name>dep</name>
          <policies>
            <placement_group>
              <name>placement-affinity-1</name>
              <type>affinity</type>
              <enforcement>strict</enforcement>
              <vm_group>g1</vm_group>
              <vm_group>g2</vm_group>
              <vm_group nc:operation="delete">g3</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_id>
    <!-- vCD orgnization -->
    <vim_project>esc</vim_project>
    <!-- vDC name -->
    <vim_vdc>VDC-1</vim_vdc>
  </locator>
  <!-- VM name in vAppTemplate -->
  <image>vm-001</image>
  <bootup_time>120</bootup_time>
  <recovery_wait_time>5</recovery_wait_time>
  <interfaces>
    <interface>
      <nicid>0</nicid>
      <network>MgtNetwork</network>
      <ip_address>10.0.0.155</ip_address>
      <mac_address>00:1C:B3:09:85:15</mac_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>10.0.0.155</ip_address>
    </static_ip_address_pool>
    <static_mac_address_pool>
      <network>MgtNetwork</network>
      <mac_address>00:1C:B3:09:85:15</mac_address>
    </static_mac_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>30</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>
                    <recovery_policy>
                      <action_on_recovery>REBOOT_ONLY</action_on_recovery>
                    </recovery_policy>
                  </vm_group>
                  <vm_group>
                    <name>g2</name>
                    <locator>
                      <!-- vCD vim connector id -->
                      <vim_id>vcd</vim_id>
                      <!-- vCD orgnization -->
                      <vim_project>esc</vim_project>
                      <!-- vDC name -->
                      <vim_vdc>VDC-1</vim_vdc>
                    </locator>
                    <!-- VM name in vAppTemplate -->
                    <image>vm-002</image>
                    <bootup_time>120</bootup_time>
                    <recovery_wait_time>5</recovery_wait_time>
                    <interfaces>
                      <interface>
                        <nicid>0</nicid>
                        <network>MgtNetwork</network>
                        <ip_address>10.0.0.156</ip_address>
                        <mac_address>00:1C:B3:09:85:16</mac_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>10.0.0.156</ip_address>
                      </static_ip_address_pool>
                      <static_mac_address_pool>
                        <network>MgtNetwork</network>
                        <mac_address>00:1C:B3:09:85:16</mac_address>
                      </static_mac_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>30</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>
          <recovery_policy>
            <action_on_recovery>REBOOT_ONLY</action_on_recovery>
          </recovery_policy>
        </vm_group>
        <vm_group nc:operation="delete">
          <name>g3</name>
          <locator>
            <!-- vCD vim connector id -->
            <vim_id>vcd</vim_id>
            <!-- vCD orgnization -->
            <vim_project>esc</vim_project>
            <!-- vDC name -->
            <vim_vdc>VDC-1</vim_vdc>
          </locator>
          <!-- VM name in vAppTemplate -->
          <image>vm-002</image>
          <bootup_time>120</bootup_time>
          <recovery_wait_time>5</recovery_wait_time>
          <interfaces>
            <interface>
              <nicid>0</nicid>
              <network>MgtNetwork</network>
              <ip_address>10.0.0.157</ip_address>
              <mac_address>00:1C:B3:09:85:17</mac_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>10.0.0.157</ip_address>
            </static_ip_address_pool>
            <static_mac_address_pool>
              <network>MgtNetwork</network>
              <mac_address>00:1C:B3:09:85:17</mac_address>
            </static_mac_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>30</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>
            <recovery_policy>
              <action_on_recovery>REBOOT_ONLY</action_on_recovery>
            </recovery_policy>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

Upgrading the Virtual Network Function Software Using Lifecycle Stages

ESC supports upgrading the VNF software application while updating a deployment. Using the policy datamodel, new Lifecycle Stages (conditions) are introduced to support the VNF upgrade. The VNF upgrade policies can be different for different VM groups. These policies are applicable for a group of VMs, and can be specified under <vm_group> rather than the entire deployment.

When a service is initially deployed, the data model has the policies configured for future software upgrade. When a deployment update request is received, VM upgrade is initiated as part of deployment update. LCS::DEPLOY_UPDATE::VM_PRE_VOLUME_DETACH is triggered before ESC detaches a volume. A script is supported at this lifecycle stage to unmount the volume before it is detached. ESC detaches and deletes the old volume which contains the old version of the software. After the volume is detached successfully, LCS::DEPLOY_UPDATE::VM_POST_VOLUME_DETACHED is triggered. A script is run at this LCS for further clean ups. When the new volume with a newer software version is attached, LCS::DEPLOY_UPDATE::VM_VOLUME_ATTACHED is triggered. ESC creates and attaches the new volume which contains the new version of the software. A script is run to mount the volume and trigger software installation. Once the volume is attached,

LCS::DEPLOY_UPDATE::VM_SOFTWARE_VERSION_UPDATED is triggered after ESC has updated the software version of the VM. A script is run at this stage to complete the configuration for the software upgrade.

Data model for VNF Software Upgrade:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
   <tenants>
      <tenant>
         <name>test</name>
         <deployments>
            <deployment>
               <name>dep</name>
               <vm_group>
                  <name>Group1</name>
                  <volumes>
                     <volume nc:operation="delete">
                        <name>v1.0</name>
                        <volid>0</volid>
                     </volume>
                     <volume>
                        <name>v2.0</name>
                        <volid>1</volid>
                        <sizeunit>GiB</sizeunit>
                        <size>2</size>
                        <bus>virtio</bus>
                        <type>lvm</type>
                        <image>Image-v2</image>
                     </volume>
                  </volumes>
                  <software_version>2.0</software_version>
                  <policies>
                     <policy>
                        <name>SVU1</name>
                        <conditions>
                           <condition>
                              <name>LCS::DEPLOY_UPDATE::PRE_VM_VOLUME_DETACH</name>
                           </condition>
                        </conditions>
                        <actions>
                           <action>
                              <name>LOG</name>
                              <type>pre_defined</type>
                           </action>
                        </actions>
                     </policy>
                     <policy>
                        <name>SVU2</name>
                        <conditions>
                           <condition>
                              <name>LCS::DEPLOY_UPDATE::POST_VM_VOLUME_ATTACHED</name>
                           </condition>
                        </conditions>
                        <actions>
                           <action>
                              <name>LOG</name>
                              <type>pre_defined</type>
                           </action>
                        </actions>
                     </policy>
                     <policy>
                        <name>SVU3</name>
```

```
                        <conditions>
                            <condition>

<name>LCS::DEPLOY_UPDATE::POST_VM_SOFTWARE_VERSION_UPDATED</name>
                            </condition>
                        </conditions>
                        <actions>
                            <action>
                                <name>LOG</name>
                                <type>pre_defined</type>
                            </action>
                        </actions>
                    </policy>
                </policies>
            </vm_group>
        </deployment>
    </deployments>
  </tenant>
 </tenants>
</esc_datamodel>
```

In this data model, the existing volume v1.0 with volid of 0 is deleted. A new volume v2.0 with volid of 1 is added. The software version, <software_version> value is changed from 1.0 to 2.0. Three policies are added for the VNF software upgrade.

✎

Note
- Instead of deleting and creating a new volume, you can update the volume properties. You can retain the name, vol_id, and image properties. If any of the above three properties change, then the volume will be deleted and created again.

- The volume size can be extended, and the bootable property can be changed. Other properties such as volume type, and image properties that are changed will trigger the volume to be created again.

- To update the volume id, you must remove the volume and add the volume again with a different volume id.

- The volume created by ESC cannot be updated by an out of band volume with same volume id, and vice versa.

Supported Lifecycle Stages (LCS) for VNF Software Upgrade

Each lifecycle stage has a condition and an action. Based on the condition, the action is executed. For information on policy driven data model, see Policy-Driven Data model. The following three conditions are configured for the VNF software upgrade:

Condition Name	Scope	Description
LCS::DEPLOY_UPDATE::VM_PRE_VOLUME_DETACH	Deployment	Triggered just before the ESC detaches a volume
LCS::DEPLOY_UPDATE::POST_VM_VOLUME_DETACHED	Deployment	Triggered immediately after ESC has detached a volume

Condition Name	Scope	Description
LCS::DEPLOY_UPDATE::POST_VM_VOLUME_ATTACHED	Deployment	Triggered immediately after ESC has attached a new volume
LCS:DEPLOY_UPDATE::POST_VM_SOFTWARE_VERSION_UPDATED	Deployment	Triggered immediately after ESC has updated the software version of the VM

LCS::DEPLOY_UPDATE::PRE_VM_VOLUME_DETACH

This LCS condition is triggered before ESC detaches the volume. A script is run to unmount the volume before it is detached.

```
<policy>
    <name>SVU1</name>
    <conditions>
        <condition>
            <name>LCS::DEPLOY_UPDATE::PRE_VM_VOLUME_DETACH</name>
        </condition>
    </conditions>
    <actions>
        <action>
            <name>LOG</name>
            <type>pre_defined</type>
        </action>
    </actions>
</policy>
```

LCS::DEPLOY_UPDATE::POST_VM_VOLUME_ATTACHED

This LCS is triggered after the ESC has attached a new volume. A script is run to mount the volume and install new applications on the new volume.

```
<policy>
    <name>SVU2</name>
    <conditions>
        <condition>
            <name>LCS::DEPLOY_UPDATE::POST_VM_VOLUME_ATTACHED</name>
        </condition>
    </conditions>
    <actions>
        <action>
            <name>LOG</name>
            <type>pre_defined</type>
        </action>
    </actions>
</policy>
```

LCS::DEPLOY_UPDATE::POST_VM_SOFTWARE_VERSION_UPDATED

This LCS is triggered after the ESC has updated the software version of the VM. A Script is run to perform final configurations to complete the software upgrade.

```
<policy>
   <name>SVU3</name>
   <conditions>
      <condition>
          <name>LCS::DEPLOY_UPDATE::POST_VM_SOFTWARE_VERSION_UPDATED</name>
      </condition>
   </conditions>
```

```
      <actions>
         <action>
            <name>LOG</name>
            <type>pre_defined</type>
         </action>
      </actions>
</policy>
```

Note   All three policies above show LOG action as the predefined action in the data model sample. If a script execution is needed, then a SCRIPT action can be added. See the Script action section below for a sample script.

Script Action

In the above examples, all the actions are pre-defined logs. You can have custom scripts instead.

```
<action>
   <name>unmount_volume</name>
   <type>SCRIPT</type>
   <properties>
      <property>
         <name>script_filename</name>
         <value>/opt/cisco/esc/esc-scripts/unmount.sh</value>
      </property>
      <property>
         <name>user_param</name>
         <value>value</value>
      </property>
   </properties>
</action>
```

All the property name and value pairs are passed to the script as space separated parameters. In the above example, the unmount.sh value will be be called by the scripts as follows:

```
/opt/cisco/esc/esc-scripts/unmount.sh user_param value
```

Prebuilt property names can be set to pass the ESC internal ids to the specified script. The prebulit property names are as follows:

```
<property>
<name>internal_deployment_id</name>
</property>
<property>
<name>external_deployment_id</name>
</property>
<property>
<name>deployment_name</name>
</property>
<property>
<name>internal_tenant_id</name>
</property>
<property>
<name>external_tenant_id</name>
</property>
```

Here is an example of a script with the prebuilt property names and values, which ESC generates.

```
script_name.sh deployment_name my-deployment-name external_deployment_id
18fbcfd5-8b63-44e0-97ec-68de25902917
external_tenant_id my-tenant-id internal_deployment_id my-tenant-idmy-deployment-name
internal_tenant_id my-tenant-id
```

By default, ESC allows 15 minutes for the script execution to complete. Some scripts may take longer time to complete. An optional property can be specified to extend the timeout value in seconds. In the example below, the timeout of the script is set to 3600 seconds.

```
<property>
    <name>wait_max_timeout</name>
    <value>3600</value>
</property>
```

Notifications for Virtual Network Function Software Upgrade

Notifications are triggered at each stage of the VNF Software upgrade.

Volume Detached

```
status SUCCESS
        status_code 200
        status_message Detached 1 volume: [Volume=test-esc-1,volid=1]
        depname dep
        tenant test
        tenant_id 9132cc90b8324a1c95a6c00975af6206
        depid eb4fe3b5-138d-41a3-b6ff-d6fa9035ca6c
        vm_group Group1
        vm_source {
            vmid cd4eeb61-61db-45a6-9da1-793be08c4de6
            hostid 8e96b8830d7bfbb337ce665586210fcca9644cbe238240e207350735
            hostname my-server-5
            software_version 1.0
            interfaces {
                interface {
                    nicid 0
                    type virtual
                    port_id 26412180-45cf-4f0b-ab45-d05bb7ca7091
                    network 943fda9e-79f8-400c-b442-3506f102721a
                    subnet e313b95c-ca1f-4c81-8d60-c9e721a85d0b
                    ip_address 192.168.0.56
                    mac_address fa:16:3e:18:90:1e
                    netmask 255.255.255.0
                    gateway 192.168.0.1
                }
            }
            volumes {
                volume {
                    display_name test-esc-1__v0_0_0_1
                    external_id 5d008a12-6fb1-492a-b648-4cf7fc8c68b1
                    bus virtio
                    type lvm
                    size 2
                }
            }
        }
        vm_target {
        }
        event {
            type VM_UPDATED
        }
```

```
        }
    }

Volume Removed

notification {
    eventTime 2016-11-24T00:27:25.457+00:00
    escEvent {
        status SUCCESS
        status_code 200
        status_message Removed 1 volume: [Volume=test-esc-3,volid=1]
        depname dep
        tenant test
        tenant_id 9132cc90b8324a1c95a6c00975af6206
        depid f938ca24-d0c2-42b3-a757-66b0543fe0a6
        vm_group Group1
        vm_source {
            vmid 91379ad1-1cfc-4a10-abaf-068d01ae92b9
            hostid 101f55110748903af4844a2517e854f64843b9ac8d880ad68be8af59
            hostname my-server-4
            software_version 1.0
            interfaces {
                interface {
                    nicid 0
                    type virtual
                    port_id a8201c3e-2c6e-4313-94d0-1b4eee14f08a
                    network 943fda9e-79f8-400c-b442-3506f102721a
                    subnet e313b95c-ca1f-4c81-8d60-c9e721a85d0b
                    ip_address 192.168.0.220
                    mac_address fa:16:3e:eb:bd:77
                    netmask 255.255.255.0
                    gateway 192.168.0.1
                }
            }
        }
        vm_target {
        }
        event {
            type VM_UPDATED
        }
    }
}

Volume Attached

notification {
    eventTime 2016-11-23T19:54:48.105+00:00
        status_message Attached 1 volume: [Volume=test-esc-2,volid=0]
        depname dep
        tenant test
        tenant_id 9132cc90b8324a1c95a6c00975af6206
        depid eb4fe3b5-138d-41a3-b6ff-d6fa9035ca6c
        vm_group Group1
        vm_source {
            vmid cd4eeb61-61db-45a6-9da1-793be08c4de6
            hostid 8e96b8830d7bfbb337ce665586210fcca9644cbe238240e207350735
            hostname my-server-5
            software_version 1.1
            interfaces {
                interface {
                    nicid 0
                    type virtual
                    port_id 26412180-45cf-4f0b-ab45-d05bb7ca7091
```

```
                    network 943fda9e-79f8-400c-b442-3506f102721a
                    subnet e313b95c-ca1f-4c81-8d60-c9e721a85d0b
                    ip_address 192.168.0.56
                    mac_address fa:16:3e:18:90:1e
                    netmask 255.255.255.0
                    gateway 192.168.0.1
                }
            }
            volumes {
                volume {
                    display_name test-esc-2__v0_0_0_1
                    external_id bf5c9a01-e9fb-42fa-89ee-73699d6c519c
                    bus virtio
                    type lvm
                    size 2
                }
            }
        }
        vm_target {
        }
        event {
            type VM_UPDATED
        }
    }
}
```

Software Version Updated

```
notification {
    eventTime 2016-11-23T20:06:56.75+00:00
    escEvent {
        status SUCCESS
        status_code 200
        status_message VM Software Updated. VM name:
[dep_Group1_0_c9edef63-4d9d-43ea-af1b-16527ed2edae], previous version: [1.0], current
version: [1.1]
        depname dep
        tenant test
        tenant_id 9132cc90b8324a1c95a6c00975af6206
        depid eb4fe3b5-138d-41a3-b6ff-d6fa9035ca6c
        vm_group Group1
        vm_source {
            vmid cd4eeb61-61db-45a6-9da1-793be08c4de6
            hostid 8e96b8830d7bfbb337ce665586210fcca9644cbe238240e207350735
            hostname my-server-5
            software_version 1.1
            interfaces {
                interface {
                    nicid 0
                    type virtual
                    port_id 26412180-45cf-4f0b-ab45-d05bb7ca7091
                    network 943fda9e-79f8-400c-b442-3506f102721a
                    subnet e313b95c-ca1f-4c81-8d60-c9e721a85d0b
                    ip_address 192.168.0.56
                    mac_address fa:16:3e:18:90:1e
                    netmask 255.255.255.0
                    gateway 192.168.0.1
                }
            }
            volumes {
                volume {
                    display_name test-esc-2__v0_0_0_1
                    external_id bf5c9a01-e9fb-42fa-89ee-73699d6c519c
                    bus virtio
```

```
                        type lvm
                        size 2
                    }
                }
            }
            vm_target {
            }
            event {
                type VM_SOFTWARE_VERSION_UPDATED
            }
        }
    }
}
```

Service Updated

```
notification {
    eventTime 2016-11-23T20:06:56.768+00:00
    escEvent {
        status SUCCESS
        status_code 200
        status_message Service group update completed successfully
        depname dep
        tenant test
        tenant_id 9132cc90b8324a1c95a6c00975af6206
        depid eb4fe3b5-138d-41a3-b6ff-d6fa9035ca6c
        vm_source {
        }
        vm_target {
        }
        event {
            type SERVICE_UPDATED
        }
    }
}
```