Deploying Virtual Network Functions

You can orchestrate VNFs within a virtual infrastructure domain—either on OpenStack or VMware vCenter. A VNF deployment is initiated as a service request through northbound interface or the ESC portal. The service request comprises of templates that consist of XML payloads and deployment parameters. This chapter describes the procedures to deploy VNFs (OpenStack or VMware vCenter), and the operations that you can perform during a deployment. For more information on deployment parameters, see Configuring Deployment Parameters.

You can assign a static IP address to connect the network to the VNF. The deployment datamodel introduces a new `ip_address` attribute from ESC Release 2.0 and later to specify the static IP address. See the Cisco Elastic Services Controller Deployment Attributes for more details.

### Important

- Deploying VNFs on OpenStack, page 1
- Deploying VNFs on VMware vCenter, page 8
- Unified Deployment, page 14
- Undeploying Virtual Network Functions, page 15

## Deploying VNFs on OpenStack

This section describes several deployment scenarios for Elastic Services Controller (ESC) and the procedure to deploy VNFs. The following table lists the different deployment scenarios:
To deploy VNFs on multiple OpenStack VIMs, see Deploying VNFs on Multiple OpenStack VIMs, on page 5.

**Deploying VNFs on a Single OpenStack 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 images, and flavors through ESC
- Deploying the VNFs using out-of-band images, flavors, volumes, and ports

Before you deploy the VNFs, you must ensure that the images, flavors, volumes, and ports are available on OpenStack, or you must create these resources. For more details on creating images, flavors, and volumes see Managing Resources Overview.

To deploy VMs on multiple VIMs, see Deploying VNFs on Multiple OpenStack VIMs, on page 5.
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. If ESC is unable to find the deployment details for a particular service, it uses the existing flavors and images under the `vm_group` to continue the deployment. If ESC is unable to find the image and flavor details, the deployment fails.

**Important**
You can also specify the subnet that is used for a network. The deployment data model introduces a new `subnet` attribute to specify the subnet. See the Cisco Elastic Services Controller Deployment Attributes for more details.

**Note**
When a SERVICE_UPDATE configuration fails, the minimum and maximum number of VMs change causing a scale in or scale out. ESC cannot rollback the minimum or maximum number of VMs in the configuration because of errors caused on OpenStack. The CDB (an ESC DB) would be out of synchronization. In this case, another SERVICE_UPDATE configuration must be performed to do a manual rollback.

For deployments on OpenStack, the UUID or name can be used to refer to the image and flavor. The name has to be unique on the VIM. If there are multiple images with the same name, the deployment cannot identify the right image and the deployment fails.

ESC also allows you to attach an out-of-band port to a VNF. To do this, pass the UUID or the name of the port in the deployment request file while initiating a service request.

When you undeploy or restore a VNF, the ports attached to that VNF will only be detached and not deleted.

ESC release 2.2 and later, all deployment and ESC event notifications show tenant UUID.

For example,

```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2016-01-22T15:14:52.484+00:00</eventTime>
  <escEvent xmlns="http://www.cisco.com/esc/esc">
    <status>SUCCESS</status>
    <status_code>200</status_code>
    <status_message>VIM Driver: VM successfully created, VM Name: [SystemAdminxyz_ab_NwDepMod1_0_5e6b7957-20e7-4df9-9113-e5fc8c047e91]</status_message>
    <tenant>admin</tenant>
    <tenant_id>62cd11f560b44bf5815eaad41fc94c80</tenant_id>
  </escEvent>
</notification>
```

A reboot time parameter is introduced in the deployment request. This provides more granular control to the reboot wait time of recovery in a deployment. In a deployment, when the VM reboots, the monitor is set with the reboot time. If the reboot time expires before receiving the VM ALIVE event, the next action such as VM_RECOVERY_COMPLETE, or undeploy is performed.

**Note**
The bootup time is used, if the reboot time is not provided.
The data model change is as follows:

```
<esc_datamodel xmlns="http://www.cisco.com/esc/esc">
  <tenants>
    <tenant>
      <name>tenant</name>
      <deployments>
        <deployment>
          <name>depz</name>
          <vm_group>
            <name>g1</name>
            <image>Automation-Cirros-Image</image>
            <flavor>Automation-Cirros-Flavor</flavor>
            <bootup_time>100</bootup_time>
            <reboot_time>30</reboot_time>
            <recovery_wait_time>10</recovery_wait_time>
            <interfaces>
              <interface>
                <nicid>0</nicid>
                <port>057a1c22-722e-44da-845b-a193e02807f7</port>
                <network>esc-net</network>
              </interface>
            </interfaces>
            <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>
                  <nicid>0</nicid>
                  <type>ICMPPing</type>
                  <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/>
          </config_data>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

Sample notification is as follows:

```
20:43:48,133 11-Oct-2016 WARN ===== SEND NOTIFICATION STARTS =====
20:43:48,133 11-Oct-2016 WARN Type: VM_RECOVERY_INIT
```
You can deploy VNFs on multiple VIMs of the same type using ESC.

ESC Release 3.0 and later supports deploying VNFs on multiple OpenStack VIMs only.

To deploy VMs on a single instance of OpenStack, see Deploying VNFs on OpenStack, on page 1. To deploy VNFs on multiple VIMs, you must:

- Configure the VIM connector and its credentials
- Create a tenant within ESC

A VIM connector registers the VIM to ESC. To deploy VNFs on multiple VIMs, you must configure the VIM connector and its credentials for each instance of the VIM. You can configure a VIM connector either at the time of installation using the bootvm.py parameters, or using the VIM connector APIs. A default VIM connector is used for a single VIM deployment. For multi VIM deployment, the locator attribute is used to specify the VIM connector.

Typically an ESC, which supports multi VIM deployment has,
• a default VIM on which ESC creates and manages resources,
• and a non-default VIM on which only deployments are supported.

For more details, see Managing VIM Connectors.

A root tenant in the data model hierarchy, which is a tenant within ESC (with the vim_mapping attribute set to false), and an out-of-band VIM tenant placed within the locator attribute must be available for deploying VNFs on multiple VIMs. If the root tenant does not exist, ESC can create a tenant during the multiple VIM deployment itself. You can create more than one ESC tenant. A user can use more than one tenant for multiple VIMs. For more information, see Managing Tenants.

In a multiple VIM deployment, you can specify the target VIM for each VM group. You can deploy each VM group on a different VIM, but the VMs within the VM group are deployed on the same VIM.

You must add a locator attribute to the VM group in the data model to enable multiple VIM deployment. The locator node consists of the following attributes:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vim_id</td>
<td>is the vim id of the target VIM. ESC defines the vim_id and maps it to the vim_connector id. The vim connector must exist before deploying to the VIM specified by the vim_id.</td>
</tr>
<tr>
<td>vim_project</td>
<td>is the tenant name created in target VIM. This is an out-of-band tenant or project existing in OpenStack.</td>
</tr>
</tbody>
</table>

In the example below, the esc-tenant is a tenant within ESC. There is no mapping to the VIM tenant, and the VMs are not deployed on this esc-tenant. The vim_project, project-test-tenant (within the locator attribute), which is created out-of-band is the tenant on which the VMs are deployed.

```xml
<tenants>
  <tenant>
    <name>esc-tenant</name>
    <deployments>
      <deployment>
        <name>dep-1</name>
        <vm_group>
          <name>group-1</name>
          <locator>
            <vim_id>vim-1</vim_id>
            <vim_project>project-test-tenant</vim_project>
          </locator>
        </vm_group>
      </deployment>
    </deployments>
  </tenant>
</tenants>
```

You can deploy VNFs on a single VIM as well with the locator attribute. That is, the data model with the locator attribute can also be used for deploying VMs on a single OpenStack VIM. To deploy without the
locator attribute (ESC Release 2.x data model), see Deploying VNFs on a Single OpenStack VIM, on page 2

The deployment data model is as follows:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<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"
    <tenants>
        <tenant>
            <name>test-esc-tenant1</name>
            <deployments>
                <deployment>
                    <name>dep-1</name>
                    <vm_group>
                        <name>g1</name>
                        <locator>
                            <vim_id>vim1</vim_id>
                            <vim_project>project-test</vim_project>
                        </locator>
                        <bootup_time>150</bootup_time>
                        <recovery_wait_time>30</recovery_wait_time>
                        <flavor>Automation-Cirros-Flavor</flavor>
                        <image>Automation-Cirros-Image</image>
                        <interfaces>
                            <interface>
                                <nicid>0</nicid>
                                <network>esc-net</network>
                            </interface>
                        </interfaces>
                        <scaling>
                            <min_active>1</min_active>
                            <max_active>1</max_active>
                            <elastic>true</elastic>
                        </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>
                    </vm_group>
                </deployment>
            </deployments>
        </tenant>
    </tenants>
</esc_datamodel>
```
A sample multiple VIM deployment data model using out-of-band resources, and creating a root tenant as part of the deployment:

```xml
<esc_datamodel>
  <tenants>
    <tenant>
      <!-- This root level tenant is an ESC tenant either previously created or created here marked by vim_mapping attribute. -->
      <name>esc-tenant-A</name>
      <vim_mapping>false</vim_mapping>
      <deployments>
        <deployment>
          <name>dep-1</name>
          <vm_group>
            <name>Grp-1</name>
            <locator>
              <vim_id>SiteA</vim_id>
              <!-- vim_project: OOB project/tenant that should already exist in the target VIM -->
              <vim_project>Project-X</vim_project>
            </locator>
            <!-- All other details in vm group remain the same. -->
            <flavor>Flavor-1</flavor>
            <image>Image-1</image>
          </vm_group>
        </deployment>
      </deployments>
    </tenant>
  </tenants>
</esc_datamodel>
```

All the VIMs specified in a multi VIM deployment must be configured and in CONNECTION_SUCCESSFUL status for the request to be accepted by ESC. If a VIM specified in the deployment is unreachable or in any other status, the request is rejected.

You can apply the affinity and anti-affinity rules for VMs in a multiple VIM deployment. For more information, see Affinity and Anti-Affinity Rules on OpenStack.

Multi VIM deployment supports recovery using the Lifecycle Stages (LCS). For more information on supported LCS, see Recovery Policy (Using the Policy Framework). You can update an existing multi VIM deployment. However, the locator attribute within the VM group cannot be updated. For more information on updating an existing deployment, see Updating an Existing Deployment.

## Deploying VNFs on VMware vCenter

This section describes the deployment scenario for Elastic Services Controller (ESC) and the procedure to deploy VNFs on VMware. In Cisco ESC Release 2.1 and later, you can deploy VNFs using out-of-band image definitions. The following table lists the deployment scenarios:
### 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.

#### Table of Scenarios

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Description</th>
<th>data model templates</th>
<th>Images</th>
<th>Advantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deploying VNFs on a single VIM by creating Images through ESC</td>
<td>The process of VNF deployment is as follows: 1 VNF Deployment- The deployment data model refers to the images created and then deploys VNFs.</td>
<td>• deployment data model  • image data model</td>
<td>Images are created through ESC using REST APIs.</td>
<td>• The images can be used in multiple VNF deployments.  • You can add or delete image definitions through ESC.</td>
</tr>
<tr>
<td>Important</td>
<td>Images are also referred to as Templates on VMware vCenter.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Deploying VNFs on a single VIM using out-of-band images</td>
<td>1 VNF Deployment- The deployment data model refers to the out-of-band images on VMware vCenter and then deploys VNFs.</td>
<td>• deployment data model  • Image on VMware vCenter</td>
<td>Images cannot be created or deleted through ESC.</td>
<td>• 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.</td>
</tr>
</tbody>
</table>
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.

In ESC 2.2, 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.

Assigning the MAC Address

ESC deployment on VMware vCenter supports assigning MAC address using the MAC address range, or MAC address list from the MAC address pool to deploy VMs to the network.

You can assign MAC address in the following ways:

Using the Interface

```
<interfaces>
  <interface>
    <nicid>1</nicid>
    <network>MANAGEMENT_NETWORK</network>
    <ip_address>10.88.0.11</ip_address>
    <mac_address>fa:16:3e:73:19:a0</mac_address>
  </interface>
</interfaces>
```

During scaling, you can assign the MAC address list or MAC address range from the MAC address pool.

```
<scaling>
  <min_active>2</min_active>
  <max_active>2</max_active>
  <elastic>true</elastic>
  <static_ip_address_pool>
    <network>MANAGEMENT_NETWORK</network>
    <ip_address>10.88.0.11</ip_address>
    <ip_address>10.88.0.12</ip_address>
    <ip_address>10.88.0.13</ip_address>
  </static_ip_address_pool>
</scaling>
```
<static_mac_address_pool>
  <network>MANAGEMENT_NETWORK</network>
  <mac_address>fa:16:3e:73:19:a0</mac_address>
  <mac_address>fa:16:3e:73:19:a1</mac_address>
  <mac_address>fa:16:3e:73:19:a2</mac_address>
</static_mac_address_pool>

Assign MAC address using MAC address range.

<static_ip_address_pool>
  <network>MANAGEMENT_NETWORK</network>
  <ip_address_range>
    <start>10.88.0.25</start>
    <end>10.88.0.27</end>
  </ip_address_range>
</static_ip_address_pool>

You cannot change the MAC or IP pool in an existing deployment, or during scaling (when min and max value are greater than 1) of VM instances in a service update.

In VMware vCenter, while assigning the MAC address, the server might override the specified value for "Generated" or "Assigned" if it does not fall in the right ranges or is determined to be a duplicate. Because of this, if ESC is unable to assign the MAC address the deployment fails.

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

- Cisco ESC Release 2.2 and earlier, supports only through REST interface.
- Cisco ESC Release 2.2 and earlier, supports only deployment and not resource management. 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).

Note

If the datacenter locator name is not specified during deployment, ESC deploys the VM in a default data center that is provided in the configuration parameters.

Earlier than Cisco ESC Release 2.2, REST deployments must use datacenter name tag without locators.

- 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.

```xml
<network>
<locators>
<datacenter>OTT-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 Deploying VNFs on VMware vCenter using ESC Portal.

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=127.0.0.1 --user=admin --privKeyFile=/var/confd/homes/admin/.ssh/confd_id_dsa --privKeyType=dsa --get -x "esc_datamodel/opdata"
<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>
<stats>
<hostname>test-ESC-2_2_6_11</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>10.85.103.22</vcenter_ip>
<vcenter_port>80</vcenter_port>
<vcenter_username>root</vcenter_username>
</vmware_config>
</system_config>
<default_locators>
<datacenter>OTT-ESC-4</datacenter>
</default_locators>
</tenants>
</esc_datamodel>
```
Unified Deployment

ESC creates OpenStack resources such as tenants, networks, and subnetworks before deploying a VNF. During unified deployment, you send a single combined request to create or delete the OpenStack resources, and deploy a VNF. You can create multiple networks and subnetworks, but can create only a single VNF and a single tenant using unified deployment.

A unified deployment request is defined as a new deployment request, and any number of networks and subnetworks located directly inside the deployment definition. Networks and subnets located directly inside the tenant are not considered part of a unified deployment request, and will not be removed during a subsequent undeploy request.

Update the deployment data model and the files with the necessary information such as the service and deployment ID, tenant, network and subnetwork ids and so on. You can either use NETCONF or REST APIs. For example, send POST REST and DELETE REST calls.

---

**Note**

A single NETCONF request can be used to perform multiple actions, such as creating networks and subnetworks; creating images, flavors and deploying VNFs.

See the Elastic Services Controller Deployment Attributes for a list of deployment attributes.

- To create a deployment datamodel with a single deployment request, send POST REST call to:
  
  http://[ESC_IP]:8080/v0/deployments/[internal_dep_id]

- To delete a single deployment request, send DELETE REST call to:
  
  http://[ESC_IP]:8080/v0/deployments/[internal_dep_id]

The VNF will be undeployed, and the network and subnet will be deleted in the specified order.

---

**Note**

If tenant creation fails as part of a unified deployment request, a manual rollback is needed to clean up ESC. As part of manual rollback, first an undeploy request is required to clean up the deployment, followed by a delete tenant request to clean up the failed tenant creation.

During an undeploy request, any network and subnetwork created as part of the unified deployment request will be deleted along with the VNF. However, the tenant created through unified deployment request will not be deleted.
Undeploying Virtual Network Functions

You can undeploy an already deployed VNF. Use the REST or NETCONF / YANG APIs to undeploy the VNF.

---

Important

You can also undeploy VNFs using the ESC portal. For more information, see ESC Portal Dashboard.

---

Sample undeploy request is as follows:

```plaintext
DELETE /v0/deployments/567 HTTP/1.1
Host: client.host.com
Content-Type: application/xml
Accept: application/xml
Client-Transaction-Id: 123456
Callback:/undeployservicecallback
```

For more details, see Cisco Elastic Services Controller API Guides.

A reboot time parameter is introduced in the deployment request. This provides more flexibility to the operation time of the deployment. In a deployment, when the VM reboots, the monitor is set with the reboot time. If the reboot time expires before the VM alive event, the next action such as vm_recovery_complete, or undeploy is performed.