Elastic Services Controller Overview

Cisco Elastic Services Controller (ESC) is a Virtual Network Functions Manager (VNFM) managing the lifecycle of Virtual Network Functions (VNFs). ESC provides agentless and multi vendor VNF management by provisioning the virtual services. ESC monitors the health of VNFs, promotes agility, flexibility, and programmability in Network Function Virtualization (NFV) environments. It provides the flexibility to define rules for monitoring and associate actions that are triggered based on the outcome of these rules. Based on the monitoring results, ESC performs scale in or scale out operations on the VNFs. In the event of a VM failure ESC also supports automatic VM recovery.

ESC fully integrates with Cisco and other third party applications. As a standalone product, the ESC can be deployed as a VNF Manager. ESC integrates with Cisco Network Services Orchestrator (NSO) to provide VNF management along with orchestration. ESC as a VNF Manager targets the virtual managed services and all service provider NFV deployments such as virtual packet core, virtual load balancers, virtual security services and so on. Complex services include multiple VMs that are orchestrated as a single service with dependencies between them.

- Change History, page 1
- Key Features of Elastic Services Controller, page 2
- ESC Architecture, page 2
- Understanding the ESC Lifecycle, page 3

Change History

Table 1: Change History for ESC 3.1

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>November 3, 2017</td>
<td>Updated with comments.</td>
<td>VIM Connector Configurations for OpenStack</td>
</tr>
</tbody>
</table>
Key Features of Elastic Services Controller

- Provides open and modular architecture, which allows multi-vendor OSS, NFVO, VNF and VIM support.
- Provides end-to-end dynamic provisioning and monitoring of virtualized services using a single point of configuration.
- Provides customization across different phases of lifecycle management; while monitoring the VM, service advertisement, and custom actions.
- Provides agentless monitoring with an integrated Monitoring Actions (MONA) engine. The monitoring engine provides simple and complex rules, to decide scale in and scale out of VMs.
- Provides scale in and scale out options based on the load of the network.
- Deploys, reboots or redeploys VMs based on the monitoring errors and threshold conditions detected as part of healing (also known as recovery).
- Supports service agility by providing faster VNF deployment and life cycle management.
- Supports multi-tenant environments.
- Supports deploying VMs on multiple VIMs (OpenStack only).
- Supports non admin roles for ESC users on OpenStack.
- Supports REST and NETCONF / YANG interfaces to provide better hierarchical configuration and data modularity.

ESC Architecture

Cisco Elastic Services Controller (ESC) is built as an open and modular architecture, that allows multi-vendor support. It performs lifecycle management of the VNFs, that is, onboarding the VNFs, deploying, monitoring, and making VNF level lifecycle decisions such as healing and scaling based on the KPI requirements. ESC and its managed VNFs are deployed as VMs running within a Virtual Infrastructure Manager (VIM). Currently, OpenStack and VMware vCenter are the supported VIMs. The ESC core engine manages transactions, validations, policies, workflows, and VM state machines. The monitoring and actions service engine in ESC performs monitoring based on several monitoring methods. Events are triggered based on the monitoring actions. The monitoring engine also supports custom monitoring plugins.

ESC can be configured for High Availability. For details, see the Cisco Elastic Services Controller Install and Upgrade Guide.

ESC interacts with the top orchestration layer using the REST and NETCONF/YANG NB APIs. The orchestration layer can be a Cisco NSO or any third party OSS or NFV Orchestrator. ESC integrates with NSO using NETCONF/YANG northbound interface support. A configuration template, Virtual Network Function Descriptor (VNFD) file is used to describe the deployment parameters and operational behaviors of
the VNFs. The VNFD file is used in the process of onboarding a VNF and managing the lifecycle of a VNF instance. The figure below represents the Cisco Elastic Services Controller architecture.

**Figure 1: Cisco Elastic Services Controller Architecture**

---

**Understanding the ESC Lifecycle**

Cisco Elastic Services Controller (ESC) provides a single point of control to manage all aspects of VNF lifecycle for generic virtual network functions (VNFs) in a dynamic environment. It provides advanced VNF lifecycle management capabilities through an open, standards-based platform that conforms to the ETSI VNF management and orchestration (MANO) reference architecture.

You can orchestrate VNFs within a virtual infrastructure domain—either on OpenStack or VMware vCenter. A VNF deployment is initiated as a service request. The service request comprises of templates that consist of XML payloads and configuration parameters.

---

**Note**

You can deploy VNFs either on OpenStack or VMware vCenter. Hybrid VNF deployment is not supported.
ESC manages the complete lifecycle of a VNF. A VNF deployment is initiated as a service request through northbound interface or the ESC portal.

**Figure 2: ESC Lifecycle**

The figure explains the lifecycle management of ESC:

- **Onboarding**—In ESC, you can onboard any new VNF type as long as it meets the prerequisites for supporting it on OpenStack and VMware vCenter. For example on Openstack, Cisco ESC supports raw image format apart from qcow2 and vmdk disk formats and config drive support for the VNF bootstrap mechanism. You can define the XML template for the new VNF type to onboard the VNF with ESC.

- **Deploying**—When a VNF is deployed, ESC applies Day Zero configuration for a new service. A typical configuration includes credentials, licensing, connectivity information (IP address, gateway), and other static parameters to make the new virtual resource available to the system. It also activates licenses for the new VNFs.

- **Monitoring**—ESC monitors the health of virtual machines using various methodologies including ICMP Ping, SNMP and so on. It tracks performance metrics such as CPU use, memory consumption, and other core parameters. The requester can specify all of the characteristics (for example, vCPU, memory, disk, monitoring KPIs, and more) typically associated with spinning up and managing a virtual machine in an XML template. It also provides an elaborate framework to monitor service performance-related metrics and other key parameters that you define.

- **Healing**—ESC heals the VNFs when there is a failure. The failure scenarios are configured in the KPI section of the data model. ESC uses KPI to monitor the VM and the events are triggered based on the KPI conditions. The actions to be taken for every event that is triggered is configured in the rules section during the deployment.

- **Updating**—ESC allows deployment updates after a successful deployment. You can either perform all the updates (that is, 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) in a single deployment or individually.
• **Undeploy**— ESC allows you to undeploy an already deployed VNF. This operation is either done using the northbound APIs or through the ESC portal.

The following section explains how to deploy VNFs on OpenStack and VMware vCenter:

### Deploying VNFs on OpenStack

In ESC, VNF deployment is initiated as a service request either originating from the ESC portal or the northbound interfaces. The service request comprises of templates that consist of XML payloads. These resources must either be available on OpenStack or can be created in ESC using the ESC portal or the northbound interfaces. For more information on managing resources in ESC, see Managing Resources Overview. The deployment data model refers to the resources to deploy VNFs on OpenStack.

Based on how the resources are setup, you can deploy VNFs in one of the following ways:

<table>
<thead>
<tr>
<th>Scenarios</th>
<th>Description</th>
<th>Resources</th>
<th>Advantages</th>
</tr>
</thead>
</table>
| Deploying VNFs on a single VIM by creating images and flavors through ESC | The deployment data model refers to the images and flavors created and then deploys VNFs. | Images and Flavors are created through ESC using NETCONF/REST APIs. | • The images and flavors can be used in multiple VNF deployments.  
• You can delete resources (images, flavors, and volumes) created by ESC. |
| Deploying VNFs on a single VIM using out-of-band images, flavors, volumes, and ports | The deployment data model refers to the out-of-band images, flavors, volumes, and ports in OpenStack and then deploys VNFs. | Images, Flavors, Volumes, and Ports are not created through ESC. | • The images, flavors, volumes, ports can be used in multiple VNF deployments.  
• You cannot delete resources that are not created by ESC. |
| Deploying VNFs on multiple VIMs using out-of-band resources | The deployment data model refers to out-of-band images, flavors, networks and VIM projects and then deploys VNFs. | Images, Flavors, VIM projects (specified in the locators) and Networks are not created through ESC. They must exist out-of-band in the VIM. | You can specify the VIM (to deploy VMs) that needs to be configured in ESC within a deployment. |

**Note**

For more information on Deploying VNFs on OpenStack, see Deploying VNFs on OpenStack.
Deploying VNFs on VMware vCenter

In ESC, VNF deployment is initiated as a service request either originating from the ESC portal or the Northbound interface. The service request comprises of templates that consist of XML payloads such as Networks, Images, and so on. These resources must be available on VMware vCenter. For more information on managing VM resources in ESC, see Managing Resources Overview. The deployment data model refers to the resources to deploy VNFs on VMware vCenter.

In Cisco ESC Release 2.0 and later, when you deploy VNFs on VMware vCenter, you can either use the out-of-band images that are already available on VMware vCenter or create an image in the ESC portal or using REST APIs. For more information on creating images in the ESC portal, see Managing Images. The deployment data model refers to these images to deploy VNFs.

Advantages

<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>Images are also referred to as Templates on VMware vCenter.</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>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>
</tbody>
</table>

For more information on Deploying VNFs on VMware vCenter, see Deploying VNFs on VMware vCenter.